Microsoft Exchange Server 2013 CU6 Documentation Help

Официальная документация компании Microsoft.
Дата выхода: 12/09/2014г.

Подготовил Pavel Nagaev.
Последнюю версию документации в PDF вы найдете на сайте
http://www.ExchangeFAQ.ru

Создано:19.11.2014, 9:07

Table of Contents

Exchange Server 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-20*

Welcome to Microsoft Exchange Server 2013! We know you're eager to get started, but there are a few things you should be aware of before you start working with Exchange 2013 and using this content.

- If you want a quick overview of what's new in Exchange 2013, check out What's new in Exchange 2013.
- If you want to learn more about Exchange 2013, check out the Exchange Server 2013 TechCenter.
- If you need more help or want to share ideas, the Exchange Server forums are a great place to start.
- To get started with Exchange 2013, head for Planning and deployment. It lays out the recommended sequence for preparing for and then installing Exchange 2013 and includes the following important topics:
  - Exchange 2013 system requirements
  - Exchange 2013 prerequisites
  - Prepare Active Directory and domains
  - Install Exchange 2013 using the Setup wizard
  - Install Exchange 2013 using unattended mode
  - Exchange 2013 post-Installation tasks

Tip:

Have you heard about the Exchange Server Deployment Assistant? It's a free online tool that helps you quickly deploy Exchange 2013 in your organization by asking you a few questions and creating a customized deployment checklist just for you. If you want to learn more about it, go to Exchange Server Deployment Assistant.

Important:

Make sure you read Release notes for Exchange 2013 before you begin your deployment. The release notes contain important information about issues you might run into during and after your deployment.

- For information on how to download Exchange 2013, see Updates for Exchange 2013.

Exchange 2013 Help

The Help content for Exchange 2013 consists of the following top-level categories:

- What's new in Exchange 2013
- Planning and deployment
- Permissions
- Messaging policy and compliance
- Anti-spam and anti-malware protection
- Mail flow
- Recipients
- Collaboration
- Email addresses and address books
- Sharing
- Clients and mobile
- Unified Messaging
- Mailbox and Client Access servers
- Edge Transport servers
- High availability and site resilience
- Exchange Management Shell
- Exchange admin center in Exchange 2013
- Server health and performance
- About Exchange documentation

Note:
Check out our other Exchange content: **Exchange Online** **Exchange Server 2013 Hybrid Deployments** **Exchange Online Protection**

Download the Exchange 2013 Help file

Looking for an offline version of Exchange 2013 Help content? Download the Help file from the Microsoft Download Center. (The online Help content may be more up-to-date than the offline Help file.)

Tell us what you think

If you have comments or questions about our topics or about the overall Help experience, we'd love to hear from you. Just send your feedback to Exchange 2013 Help Feedback. Your comments will help us provide the most accurate and concise content.

What's new in Exchange 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-19*

Microsoft Exchange Server 2013 brings a new rich set of technologies, features, and services to the Exchange Server product line. Its goal is to support people and organizations as their work habits evolve from a communication focus to a collaboration focus. At the same time, Exchange Server 2013 helps lower the total cost of ownership whether you deploy Exchange 2013 on-premises or provision your mailboxes in the cloud. New features and functionality in Exchange 2013 are designed to do the following:

- **Support a multigenerational workforce** Social integration and making it easier to find people is important to users. *Smart Search* learns from users' communication and collaboration behavior to enhance and prioritize search results in Exchange. Also, with Exchange 2013, users can merge contacts from multiple sources to provide a single view of a person, by linking contact information pulled from multiple locations.
- **Provide an engaging experience** Microsoft Outlook 2013 and Microsoft Outlook Web App have a fresh new look. Outlook Web App emphasizes a streamlined user interface that also supports the use of touch, enhancing the mobile device experience with Exchange.
- **Integrate with SharePoint and Lync** Exchange 2013 offers greater integration with Microsoft SharePoint 2013 and Microsoft Lync 2013 through site mailboxes and In-Place eDiscovery. Together, these products offer a suite of features that make scenarios such as enterprise eDiscovery and collaboration using site mailboxes possible.
- **Help meet evolving compliance needs** Compliance and eDiscovery are challenging for many organizations. Exchange 2013 helps you to find and search data not only in Exchange, but across your organization. With improved search and indexing, you can search across Exchange 2013, Lync 2013, SharePoint 2013, and Windows file servers. In addition, data loss prevention (DLP) can help keep your organization safe from users mistakenly sending sensitive information to unauthorized people. DLP helps you identify, monitor, and protect sensitive data through deep content analysis.
- **Provide a resilient solution** Exchange 2013 builds upon the Exchange Server 2010 architecture and has been redesigned for simplicity of scale, hardware utilization, and failure isolation.

For information about the changes made to Exchange Server 2013 since release to manufacturing (RTM), see Updates for Exchange 2013.

See the following sections for more information about what's new in Exchange 2013:

Exchange admin center

Exchange 2013 architecture

Setup

Messaging policy and compliance

Anti-malware protection

Mail flow

Recipients

Sharing and collaboration

Integration with SharePoint and Lync

Clients and mobile

Unified Messaging

Batch mailbox moves

High availability and site resilience

Exchange workload management

Note:

For information about features in earlier versions of Exchange that have been removed, discontinued, or replaced in Exchange Server 2013, see What's discontinued in Exchange 2013. Also, you may be interested in Release notes for Exchange 2013.

Exchange admin center

Exchange 2013 provides a single unified management console that allows for ease of use and is optimized for management of on-premises, online, or hybrid deployments. The *Exchange admin center* (EAC) in Exchange 2013 replaces the Exchange 2010 Exchange Management Console (EMC) and the Exchange Control Panel (ECP). (However, "ECP" is still the name of the virtual directory used by the EAC.) Some EAC features include:

- **List view** The list view in EAC has been designed to remove key limitations that existed in ECP. ECP was limited to displaying up to 500 objects and, if you wanted to view objects that weren't listed in the details pane, you needed to use searching and filtering to find those specific objects. In Exchange 2013, the viewable limit from within the EAC list view is approximately 20,000 objects. After the EAC returns the results, the EAC client performs the searching and sorting, which greatly increases the performance compared to the ECP in Exchange 2010. In addition, paging has been added so that you can page to the results. You can also configure page size and export to a .csv file.
- **Add/Remove columns to the Recipient list view** You can choose which columns to view, and with local cookies, you can save your custom list views per machine that you use to access the EAC.
- **Secure the ECP virtual directory** You can partition access from the Internet and intranets from within the ECP IIS virtual directory to allow or disallow management features. With this feature, you can permit or deny access to users trying to access the EAC from the Internet outside of your organizational environment, while still allowing access to an end-user's Outlook Web App Options.
- **Public Folder management** In Exchange 2010 and Exchange 2007, public folders were managed through the Public Folder administration console. Public folders are now in the EAC, and you don't need a separate tool to manage them.

- **Notifications** In Exchange 2013, the EAC now has a Notification viewer so that you can view the status of long-running processes and, if you choose, receive notification via an email message when the process completes.
- **Role Based Access Control (RBAC) User Editor** In Exchange 2010, you could use the RBAC User Editor in the Exchange Toolbox to add users to management role groups. In Exchange 2013, the RBAC User Editor functionality is now in the EAC and you don't need a separate tool to manage RBAC.
- **Unified Messaging Tools** In Exchange 2010, you could use the Call Statistics and User Call Logs tools to help provide UM statistics and information about specific calls for a UM-enabled user. In Exchange 2013, the Call Statistics and User Call Logs tools are now in the EAC and you don't need a separate tool to manage them.
- **Groups enhancements** The Exchange Admin Center (EAC) can now display up to 10,000 recipients in the **Groups Select Members** window. By default, up to 500 recipients are returned when you open the **Select Members** window, however, you can choose to list up to 10,000 recipients by clicking **Get All Results** beneath the recipient list. We now support browsing more than 500 recipients by using the scroll bar and we've also added enhanced search features to enable you to filter recipients that are displayed in the recipient list. You can filter by:
  - city
  - company
  - country/region
  - department
  - office
  - title

For more information, see Exchange admin center in Exchange 2013.

Exchange 2013 architecture

Previous versions of Exchange were optimized and architected with certain technological constraints that existed at that time. For example, during development for Exchange 2007, one of the key constraints was CPU performance. To alleviate that constraint, Exchange 2007 was split into different server roles that allowed scale out through server separation. However, server roles in Exchange 2007 and Exchange 2010 were tightly coupled. The tight coupling of the roles had several downsides including version dependency, geo-affinity (requiring all roles in a specific site), session affinity (requiring expensive layer 7 hardware load balancing), and namespace complexity.

Today, CPU horsepower is significantly less expensive and is no longer a constraining factor. With that constraint lifted, the primary design goal for Exchange 2013 is for simplicity of scale, hardware utilization, and failure isolation. With Exchange 2013, we reduced the number of server roles to two: the Client Access server role and the Mailbox server role.

Currently, there isn't an Exchange 2013 version of the Edge Transport server. If you need an Edge Transport server, you can install an Exchange 2007 or Exchange 2010 Edge Transport server in your perimeter network. For more information, see Use an Exchange 2010 or 2007 Edge Transport server

in Exchange 2013.

The Mailbox server includes all the traditional server components found in Exchange 2010: the Client Access protocols, Transport service, Mailbox databases, and Unified Messaging. The Mailbox server handles all activity for the active mailboxes on that server. The Client Access server provides authentication, limited redirection, and proxy services. The Client Access server itself doesn't do any data rendering. The Client Access server is a thin and stateless server. There is never anything queued or stored on the Client Access server. The Client Access server offers all the usual client access protocols: HTTP, POP and IMAP, and SMTP.

With this new architecture, the Client Access server and the Mailbox server have become “loosely coupled”. All processing and activity for a specific mailbox occurs on the Mailbox server that houses the active database copy where the mailbox resides. All data rendering and data transformation is performed local to the active database copy, eliminating concerns of version compatibility between the Client Access server and the Mailbox server.

The Exchange 2013 architecture provides the following benefits:

- **Version upgrade flexibility** No more rigid upgrade requirements. A Client Access server can be upgraded independently and in any order in relation to the Mailbox server.
- **Session indifference** With Exchange 2010, session affinity to the Client Access server role was required for several protocols. In Exchange 2013, the client access and mailbox components reside on the same Mailbox server. Because the Client Access server simply proxies all connections for a user to a specific Mailbox server, no session affinity is required at the Client Access servers. This allows inbound connections to Client Access servers to be balanced using techniques provided by load-balancing technology like least connection or round-robin.
- **Deployment simplicity** With an Exchange 2010 site-resilient design, you needed up to eight different namespaces: two Internet Protocol namespaces, two for Outlook Web App fallback, one for Autodiscover, two for RPC Client Access, and one for SMTP. A legacy namespace was also required if you were upgrading from Exchange 2003 or Exchange 2007. With Exchange 2013, the minimum number of namespaces drops to two. If you’re coexisting with Exchange 2007, you still need to create a legacy hostname, but if you’re coexisting with Exchange 2010 or you’re installing a new Exchange 2013 organization, the minimum number of namespaces you need is two: one for client protocols and one for Autodiscover. You may also need an SMTP namespace.

As a result of these architectural changes, there have been some changes to client connectivity. First, RPC is no longer a supported direct access protocol. This means that all Outlook connectivity must take place using RPC over HTTP (also known as Outlook Anywhere). At first glance, this may seem like a limitation, but it actually has some added benefits. The most obvious benefit is that there is no need to have the RPC client access service on the Client Access server. This results in the reduction of two namespaces that would normally be required for a site-resilient solution. In addition, there is no longer any requirement to provide affinity for the RPC client access service.

Second, Outlook clients no longer connect to a server FQDN as they have done in all previous versions of Exchange. Outlook uses Autodiscover to create a new connection point comprised of mailbox GUID, @ symbol, and the domain portion of the user’s primary SMTP address. This simple

change results in a near elimination of the unwelcome message of “Your administrator has made a change to your mailbox. Please restart.” Only Outlook 2007 and higher versions are supported with Exchange 2013.

The high availability model of the mailbox component has not changed significantly since Exchange 2010. The unit of high availability is still the database availability group (DAG). The DAG still uses Windows Server failover clustering. Continuous replication still supports both file mode and block mode replication. However, there have been some improvements. Failover times have been reduced as a result of transaction log code improvements and deeper checkpoint on the passive databases. The Exchange Store service has been re-written in managed code (see the "Managed Store" section later in this topic). Now, each database runs under its own process, allowing for isolation of store issues to a single database.

Managed Store

In Exchange 2013, the *Managed Store* is the name of the newly rewritten Information Store processes, Microsoft.Exchange.Store.Service.exe and Microsoft.Exchange.Store.Worker.exe. The new Managed Store is written in C# and tightly integrated with the Microsoft Exchange Replication service (MSExchangeRepl.exe) to provide higher availability through improved resiliency. In addition, the Managed Store has been architected to enable more granular management of resource consumption and faster root cause analysis through improved diagnostics.

The Managed Store works with the Microsoft Exchange Replication service to manage mailbox databases, which continues to use Extensible Storage Engine (ESE) as the database engine. Exchange 2013 includes significant changes to the mailbox database schema that provide many optimizations over previous versions of Exchange. In addition to these changes, the Microsoft Exchange Replication service is responsible for all service availability related to Mailbox servers. The architectural changes enable faster database failover and better physical disk failure handling.

The Managed Store is also integrated with the Search Foundation search engine (the same search engine used by SharePoint 2013) to provide more robust indexing and searching when compared to Microsoft Search in previous versions of Exchange.

For more information, see High availability and site resilience.

Certificate management

Managing digital certificates is one of the most important security-related tasks for your Exchange organization. Ensuring that certificates are appropriately configured is key to delivering a secure messaging infrastructure for the enterprise. In Exchange 2010, the Exchange Management Console was the primary method of managing certificates. In Exchange 2013, certificate management functionality is provided in the Exchange admin center, the new Exchange 2013 administrator user interface.

The work in Exchange 2013 related to certificates focused around minimizing the number of

certificates that an Administrator must manage, minimizing the interaction the Administrator must have with certificates, and allowing management of certificates from a central location. Benefits resulting from the changes in certificate management are:

- Certificate management can be performed on the Client Access server or the Mailbox server. The Mailbox server has a self-signed certificate installed by default. The Client Access server automatically trusts the self-signed certificate on the Exchange 2013 Mailbox server, so clients will not receive warnings about a self-signed certificate not being trusted provided that the Exchange 2013 Client Access server has a non-self-signed certificate from either a Windows certificate authority (CA) or a trusted third party.
- In previous versions of Exchange, it was difficult to see when a digital certificate was nearing expiration. In Exchange 2013, the Notifications center will display warnings when a certificate stored on any Exchange 2013 server is about to expire. Administrators can also choose to receive these notifications via email.

For more information, see Digital certificates and SSL.

Setup

Setup has been completely rewritten so that installing Exchange 2013 and making sure you've got the latest product rollups and security fixes is easier than ever. Here are some of the improvements we've made:

- **Always up-to-date Setup** When you run the Setup wizard, you'll be given the option to download and use the latest product rollups, security fixes, and language packs. This option doesn't just update the files that'll be used to run Exchange; Setup itself can be updated. This design enables us to continue to improve Setup post-release and include and update readiness checks as requirements are updated or changed.

If you're using unattended Setup mode, we can't automatically download updates. However, you can still take advantage of running the latest version of Setup by downloading the latest updates beforehand, and use the `/UpdatesDir: <path>` parameter to allow Setup to update itself before the installation process begins.

- **Improved readiness checks** Readiness checks make sure that your computer and your organization are ready for Exchange 2013. After you've provided the necessary information about your installation to Setup, the readiness checks are run before installation begins. The new readiness check engine now runs through all checks before reporting back to you on what actions need to be performed before Setup can continue, and it does so faster than ever. As with previous versions of Exchange, you can tell Setup to install the Windows features required by Setup so you don't have to install them manually.
- **Simplified and modern wizard** We've removed all the steps in the Setup wizard that aren't absolutely required for you to install Exchange. What's left is an easy-to-follow wizard that takes you through the installation process one step at a time.

For more information, see Planning and deployment.

Messaging policy and compliance

There are two new message policy and compliance features in Exchange 2013: Data loss prevention and the Microsoft Rights Management connector.

Data loss prevention (DLP) capabilities help you protect your sensitive data and inform users of internal compliance policies. DLP can also help keep your organization safe from users who might mistakenly send sensitive information to unauthorized people. DLP helps you identify, monitor, and protect sensitive data through deep content analysis. Exchange 2013 offers built-in DLP policies based on regulatory standards such as personally identifiable information (PII) and payment card industry data security standards (PCI), and is extensible to support other policies important to your business. Additionally, the new PolicyTips in Outlook 2013 inform users about policy violations before sensitive data is sent.

The Microsoft Rights Management connector (RMS connector) is an optional application that helps you enhance data protection for your Exchange 2013 server by connecting to cloud-based Microsoft Rights Management services. Once you install the RMS connector, it provides continuous data protection throughout the lifespan of the information and because these services are customizable, you can define the level of protection you need. For example, you can limit email message access to specific users or set view-only rights for certain messages.

To learn more about these features see:

Data loss prevention

Rights Management connector

In-place Archiving, retention, and eDiscovery

Exchange 2013 includes the following improvements to In-Place Archiving, retention, and eDiscovery to help your organization meet its compliance needs:

- **In-Place Hold** In-Place Hold is a new unified hold model that allows you to meet legal hold requirements in the following scenarios:
  - Preserve the results of the query (query-based hold), which allows for scoped immutability across mailboxes.
  - Place a time-based hold to meet retention requirements (for example, retain all items in a mailbox for seven years, a scenario that required the use of Single Item Recovery/Deleted Item Retention in Exchange 2010).
  - Place a mailbox on indefinite hold (similar to litigation hold in Exchange 2010).
  - Place a user on multiple holds to meet different case requirements.
- **In-Place eDiscovery** In-Place eDiscovery allows authorized users to search mailbox data across all mailboxes and In-Place Archives in an Exchange 2013 organization and copy messages to a discovery mailbox for review. In Exchange 2013, In-Place eDiscovery has been enhanced to allow discovery managers to perform more efficient searches and hold. These enhancements include:
  - **Federated search** allows you to search and preserve data across multiple data repositories.

With Exchange 2013, you can perform in-place eDiscovery searches across Exchange, SharePoint 2013, and Lync 2013. You can use the eDiscovery Center in SharePoint 2013 to perform In-Place eDiscovery search and hold.

- **Query-based In-Place Hold** allows you to save the results of the query, which allows for scoped immutability across mailboxes.
- **Export search results** Discovery Managers can export mailbox content to a .pst file from the SharePoint 2013 eDiscovery Console. Mailbox export request cmdlets are no longer required to export a mailbox to a .pst file.
- **Keyword statistics** Search statistics are offered on a per search term basis. This enables a Discovery Manager to quickly make intelligent decisions about how to further refine the search query to provide better results. eDiscovery search results are sorted by relevance.
- **KQL syntax** Discovery Managers can use Keyword Query Language (KQL) syntax in search queries. KQL is similar to the Advanced Query Syntax (AQS), which was used for discovery searches in Exchange 2010.
- **In-Place eDiscovery and Hold wizard** Discovery Managers can use the new In-Place eDiscovery and Hold wizard to perform eDiscovery and hold operations.

Note:

If SharePoint 2013 isn't available, a subset of the eDiscovery functionality is available in the Exchange admin center.

- **Search across primary and archive mailboxes in Outlook Web App** Users can search across their primary and archive mailboxes in Outlook Web App. Two separate searches are no longer necessary.
- **Archive Lync content** Exchange 2013 supports archiving of Lync 2013 content in a user's mailbox. You can place Lync content on hold using In-Place Hold and use In-Place eDiscovery to search Lync content archived in Exchange.
- **Retention policies** Retention policies help your organization reduce risks associated with email and other communications and also meet email retention requirements. Retention policies include the following enhancements:
  - **Support for Calendar and Tasks retention tags** You can create retention policy tags for the Calendar and Tasks default folders to expire items in these folders. Items in these folders are also moved to the user's archive based on the archive policy settings applied to the mailbox.
  - **Improved ability to retain items for a specified period** You can use retention policy and a time-based In-Place Hold to enforce retention of items for a set period.

For more information, see Messaging policy and compliance.

Transport rules

Transport rules in Exchange Server 2013 are a continuation of the features that are available in Exchange Server 2010. However, several improvements have been made to transport rules in Exchange 2013. The most important change is the support for data loss prevention (DLP). There are also new predicates and actions, enhanced monitoring, and a few architectural changes.

For detailed information, see What's new for transport rules.

Information Rights Management

Information Rights Management (IRM) is compatible with Cryptographic Mode 2, an Active Directory Rights Management Services (AD RMS) cryptography mode that supports stronger encryption by allowing you to use 2048-bit keys for RSA and 256-bit keys for SHA-1. Additionally, Mode 2 enables you to use the SHA-2 hashing algorithm. For more information about cryptographic modes in AD RMS, see AD RMS Cryptographic Modes.

Auditing

Exchange 2013 includes the following improvements to auditing:

- **Auditing reports** The EAC includes auditing functionality so that you can run reports or export entries from the mailbox audit log and the administrator audit log. The mailbox audit log records whenever a mailbox is accessed by someone other than the person who owns the mailbox. This can help you determine who has accessed a mailbox and what they have done. The administrator audit log records any action, based on an Exchange Management Shell cmdlet, performed by an administrator. This can help you troubleshoot configuration issues or identify the cause of problems related to security or compliance. For more information, see Exchange auditing reports.
- **Viewing the administrator audit log** Instead of exporting the administrator audit log, which can take up to 24 hours to receive in an email message, you can view administrator audit log entries in the EAC. To do this, go to **Compliance Management** > **Auditing** and click **View the administrator audit log**. Up to 1000 entries will be displayed on multiple pages. To narrow the search, you can specify a date range. For more information, see View the administrator audit log.

Anti-malware protection

The built-in malware filtering capabilities of Exchange 2013 helps protect your network from malicious software transferred through email messages. All messages sent or received by your Exchange server are scanned for malware (viruses and spyware). If malware is detected, the message is deleted. Notifications may also be sent to senders or administrators when an infected message is deleted and not delivered. You can also choose to replace infected attachments with either default or custom messages that notify the recipients of the malware detection.

For more information about anti-malware protection, see Anti-malware protection.

Mail flow

How messages flow through an organization and what happens to them has changed significantly in Exchange 2013. Following is a brief overview of the changes:

- **Transport pipeline** The transport pipeline in Exchange 2013 is now made up of several different

services: the Front End Transport service on Client Access servers, the Transport service on Mailbox servers, and the Mailbox Transport service on Mailbox servers. For more information, see Mail flow.

- **Routing** Mail routing in Exchange 2013 recognizes DAG boundaries as well as Active Directory site boundaries. Also, mail routing has been improved to queue messages more directly for internal recipients. For more information, see Mail routing.
- **Connectors** The default maximum message size for a Send connector or a Receive connector, as specified by the *MaxMessageSize* parameter, has been increased from 10MB to 25MB. For more information about how to set parameters on a connector, see Set-SendConnector and Set-ReceiveConnector.

You can set a Send connector in the Transport service of a Mailbox server to route outbound mail through a Front End transport server in the local Active Directory site, by means of the *FrontEndProxyEnabled* parameter of the **Set-SendConnector** cmdlet, thus consolidating how email is routed from the Transport service.

- **Edge Transport** Currently, there isn't an Exchange 2013 version of the Edge Transport server. If you need an Edge Transport server, you can install an Exchange 2007 or Exchange 2010 Edge Transport server in your perimeter network. For more information, see Use an Exchange 2010 or 2007 Edge Transport server in Exchange 2013.

Recipients

This section describes the enhancements for managing recipients in Exchange 2013:

- **Group naming policy** Administrators can now use the EAC to create a *group naming policy*, which lets you standardize and manage the names of distribution groups created by users in your organization. You can require a specific prefix and suffix be added to the name for a distribution group when it's created, and you can block specific words from being used. This capability helps you minimize the use of inappropriate words in group names.

For more information, see Create a distribution group naming policy.

- **Message tracking** Administrators can also use the EAC to track delivery information for email messages sent to or received by any user in your organization. You just select a mailbox, and then search for messages sent to or received by a different user. You can narrow the search by searching for specific words in the subject line. The resulting delivery report tracks a message through the delivery process and specifies if the message was successfully delivered, pending delivery, or if it wasn't delivered.

For more information, see Track messages with delivery reports.

Sharing and collaboration

This section describes the sharing and collaboration enhancements in Exchange 2013.

- **Public folders** Public folders now take advantage of the existing high availability and storage technologies of the mailbox store. The public folder architecture uses specially designed mailboxes to store both the hierarchy and the public folder content. This new design also means

that there is no longer a public folder database. Public folder replication now uses the continuous replication model. High availability for the hierarchy and content mailboxes is provided by the database availability group (DAG). With this design, we're moving away from a multi-master replication model to a single-master replication model.

For more information, see Public folders.

- **Site mailboxes** Email and documents are traditionally kept in two unique and separate data repositories. Most teams would typically collaborate using both mediums. The challenge is that email and documents are accessed using different clients, which usually results in a reduction in user productivity and a degraded user experience.

The *site mailbox* is a new concept in Exchange 2013 that attempts to solve these problems. Site mailboxes improve collaboration and user productivity by allowing access to both documents in a SharePoint site and email messages in Outlook 2013, using the same client interface. A site mailbox is functionally comprised of SharePoint site membership (owners and members), shared storage through an Exchange mailbox for email messages and a SharePoint site for documents, and a management interface that addresses provisioning and lifecycle needs.

For more information, see Site mailboxes.

- **Shared mailboxes** In previous versions of Exchange, creating a shared mailbox was a multi-step process in which you had to use the Exchange Management Shell to set the delegate permissions. In Exchange 2013, you can now create a shared mailbox in one step via the Exchange admin center. In the EAC, go to **Recipients** > **Shared Mailboxes** to create a shared mailbox. Shared mailbox is now a recipient type, so you can easily search for your shared mailboxes in either the user interface or by using the Shell.

For more information, see Shared mailboxes.

Integration with SharePoint and Lync

Exchange 2013 offers greater integration with SharePoint 2013 and Lync 2013. Benefits of this enhanced integration include:

- Exchange 2013 integrates with SharePoint 2013 to allow users to collaborate more effectively by using site mailboxes.
- Lync Server 2013 can archive content in Exchange 2013 and use Exchange 2013 as a contact store.
- Discovery Managers can perform In-Place eDiscovery and Hold searches across SharePoint 2013, Exchange 2013, and Lync 2013 data.
- Oauth authentication allows partner applications to authenticate as a service or impersonate users where required.

For more information, see Integration with SharePoint and Lync.

Clients and mobile

The Outlook Web App user interface is new and optimized for tablets and smartphones as well as desktop and laptop computers. New features include apps for Outlook, which allow users and

administrators to extend the capabilities of Outlook Web App; Contact linking, the ability for users to add contacts from their LinkedIn accounts; and updates to the look and features of the calendar.

For more information, see What's new for Outlook Web App in Exchange 2013.

Unified Messaging

Unified Messaging in Exchange 2013 contains essentially the same voice mail features included in Exchange 2010. However, some new and enhanced features and functionality have been added to those existing features. More importantly, architectural changes in Exchange 2013 Unified Messaging resulted in components, services, and functionality that were included with the Unified Messaging server role in Exchange 2010 to be divided between the Exchange 2013 Client Access and Mailbox server roles.

For more details, see What's new for Unified Messaging in Exchange 2013.

Batch mailbox moves

Exchange 2013 introduces the concept of batch moves. The new move architecture is built on top of MRS (Mailbox Replication service) moves with enhanced management capability. The new batch move architecture features the following enhancements:

- Ability to move multiple mailboxes in large batches.
- Email notification during move with reporting.
- Automatic retry and automatic prioritization of moves.
- Primary and personal archive mailboxes can be moved together or separately.
- Option for manual move request finalization, which allows you to review a move before you complete it.
- Periodic incremental syncs to migrate the changes.

For more information, see Manage on-premises moves.

High availability and site resilience

Exchange 2013 uses DAGs and mailbox database copies, along with other features such as single item recovery, retention policies, and lagged database copies, to provide high availability, site resilience, and Exchange native data protection. The high availability platform, the Exchange Information Store and the Extensible Storage Engine (ESE), have all been enhanced to provide greater availability, easier management, and to reduce costs. These enhancements include:

- **Managed availability** With managed availability, internal monitoring and recovery-oriented features are tightly integrated to help prevent failures, proactively restore services, and initiate server failovers automatically or alert administrators to take action. The focus is on monitoring and managing the end user experience rather than just server and component uptime to help keep the service continuously available.

- **Managed Store** The Managed Store is the name of the newly rewritten Information Store processes in Exchange 2013. The new Managed Store is written in C# and tightly integrated with the Microsoft Exchange Replication service (MSExchangeRepl.exe) to provide higher availability through improved resiliency.
- **Support for multiple databases per disk** Exchange 2013 includes enhancements that enable you to support multiple databases (mixtures of active and passive copies) on the same disk, thereby leveraging larger disks in terms of capacity and IOPS as efficiently as possible.
- **Automatic reseed** Enables you to quickly restore database redundancy after disk failure. If a disk fails, the database copy stored on that disk is copied from the active database copy to a spare disk on the same server. If multiple database copies were stored on the failed disk, they can all be automatically re-seeded on a spare disk. This enables faster reseeds, as the active databases are likely to be on multiple servers and the data is copied in parallel.
- **Automatic recovery from storage failures** This feature continues the innovation introduced in Exchange 2010 to allow the system to recover from failures that affect resiliency or redundancy. In addition to the Exchange 2010 bugcheck behaviors, Exchange 2013 includes additional recovery behaviors for long I/O times, excessive memory consumption by MSExchangeRepl.exe, and severe cases where the system is in such a bad state that threads can't be scheduled.
- **Lagged copy enhancements** Lagged copies can now care for themselves to a certain extent using automatic log play down. Lagged copies will automatically play down log files in a variety of situations, such as single page restore and low disk space scenarios. If the system detects that page patching is required for a lagged copy, the logs will be automatically replayed into the lagged copy to perform page patching. Lagged copies will also invoke this auto replay feature when a low disk space threshold has been reached, and when the lagged copy has been detected as the only available copy for a specific period of time. In addition, lagged copies can leverage Safety Net, making recovery or activation much easier. *Safety Net* is improved functionality in Exchange 2013 based on the transport dumpster of Exchange 2010.
- **Single copy alert enhancements** The single copy alert introduced in Exchange 2010 is no longer a separate scheduled script. It's now integrated into the managed availability components within the system and is a native function within Exchange.
- **DAG network auto-configuration** DAGs networks can be automatically configured by the system based on configuration settings. In addition to manual configuration options, DAGs can also distinguish between MAPI and Replication networks and configure DAG networks automatically.

For more information about both of these features, see High availability and site resilience and Changes to high availability and site resilience over previous versions.

Exchange workload management

An Exchange workload is an Exchange server feature, protocol, or service that has been explicitly defined for the purposes of Exchange system resource management. Each Exchange workload consumes system resources such as CPU, mailbox database operations, or Active Directory requests to execute user requests or run background work. Examples of Exchange workloads include Outlook

Web App, Exchange ActiveSync, mailbox migration, and mailbox assistants.

There are two ways to manage Exchange workloads in Exchange 2013:

- **Monitor the health of system resources** Managing workloads based on the health of system resources is new in Exchange 2013.
- **Control how resources are consumed by individual users** Controlling how resources are consumed by individual users was possible in Exchange 2010 (where it's called user throttling), and this capability has been expanded for Exchange 2013.

For more information about these features, see Exchange workload management.

What's discontinued in Exchange 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-25*

This topic discusses the components, features, or functionality that have been removed, discontinued, or replaced in Microsoft Exchange Server 2013.

Note:

The following topics may also interest you:

- What's new in Exchange 2013 Information about new features and functionality in Exchange Server 2013.
- Developer roadmap for Exchange 2013 See the "Development technologies removed from Exchange" section for information about the API and Development features discontinued in Exchange 2013.

Discontinued features from Exchange 2010 to Exchange 2013

This section lists the Exchange Server 2010 features that are no longer available in Exchange 2013.

Architecture

Feature	Comments and mitigation
Hub Transport server role	The Hub Transport server role has been replaced by Transport services which run on

	the Mailbox and Client Access server roles. The Mailbox server role includes the Microsoft Exchange Transport, Microsoft Exchange Mailbox Transport Delivery, and the Microsoft Exchange Mailbox Transport Submission service. The Client Access server role includes the Microsoft Exchange Frontend Transport service. For more information, see Mail flow.
Unified Messaging server role	The Unified Messaging server role has been replaced by Unified Messaging services which run on the Mailbox and Client Access server roles. The Mailbox server role includes the Microsoft Exchange Unified Messaging service and the Client Access server role includes the Microsoft Exchange Unified Messaging Call Router service. For more information, see Voice architecture changes.

Management interfaces

Feature	Comments and mitigation
Exchange Management Console and Exchange Control Panel	The Exchange Management Console and the Exchange Control Panel have been replaced by the Exchange Admin Center (EAC). EAC uses the same virtual directory (/ecp) as the Exchange Control Panel. For more information, see Exchange admin center in Exchange 2013.

Client access

Feature	Comments and mitigation
Outlook 2003 is not supported	To connect Microsoft Outlook to Exchange 2013, the use of the Autodiscover service is required. However, Microsoft Outlook 2003

	doesn't support the use of the Autodiscover service.
RPC/TCP access for Outlook clients	In Exchange 2013, Microsoft Outlook clients can connect using Outlook Anywhere (RPC/HTTP) or MAPI over HTTP in Exchange 2013 Service Pack 1 and Outlook 2013 Service Pack 1 and later. If you have Outlook clients in your organization, using Outlook Anywhere and/or MAPI over HTTP is required. For more information, see Outlook Anywhere and MAPI over HTTP.

Outlook Web App and Outlook

Feature	Comments and mitigation
Spell check	Outlook Web App no longer has built-in spell check services. Instead, it uses the spell check features in your Web browsers.
Customizable filters	Outlook Web App no longer has customizable filtered views and no longer supports saving filtered views to Favorites. Customizable filters have been replaced by fixed filters that can be used to view all messages, unread messages, messages sent to the user, or flagged messages.
Message flags	The ability to set a custom date on a message flag isn't available in Outlook Web App. You can use Outlook to set custom dates.
Chat contact list	The chat contact list that appeared in the folder list in Outlook Web App for Exchange 2010 is no longer available.

Search folders	The ability for users to use Search folders isn't currently available in Outlook Web App.

Mail flow

Feature	Comments and mitigation
Linked connectors	The ability to link a Send connector to a Receive connector has been removed. Specifically, the *LinkedReceiveConnector* parameter has been removed from New-SendConnector and Set-SendConnector.

Anti-spam and anti-malware

Feature	Comments and mitigation
Anti-spam agent management in the EMC	In Exchange 2010, when you enabled the anti-spam agents on a Hub Transport server, you could manage the anti-spam agents in the Exchange Management Console (EMC). In Exchange 2013, when you enable the anti-spam agents on a Mailbox server, you can't manage the agents using the EAC. You can only use the Shell. For information about how to enable the anti-spam agents on a Mailbox server, see Enable anti-spam functionality on Mailbox servers.
Connection Filtering agent on Hub Transport servers	In Exchange 2010, when you enabled the anti-spam agents on a Hub Transport server, the Attachment Filter agent was the only anti-spam agent that wasn't available. In Exchange 2013, when you enable the anti-spam agents on a Mailbox server, the Attachment Filter agent and the Connection Filtering agent aren't available. The Connection Filtering agent provides IP

	Allow List and IP Block List capabilities. For information about how to enable the anti-spam agents on a Mailbox server, see Enable anti-spam functionality on Mailbox servers. **Note:** You can't enable the anti-spam agents on an Exchange 2013 Client Access server. Therefore, the only way to get the Connection Filtering agent is to install an Edge Transport server in the perimeter network. For more information, see Edge Transport servers.

Messaging policy and compliance

Feature	Comments and mitigation
Managed Folders	In Exchange 2010, you use managed folders for messaging retention management (MRM). In Exchange 2013, managed folders aren't supported. You must use retention policies for MRM. **Note:** Cmdlets related to managed folders are still available. You can create managed folders, managed content settings and managed folder mailbox policies, and apply a managed folder mailbox policy to a user, but the MRM assistant skips processing of mailboxes that have a managed folder mailbox policy applied.
Port Managed Folder wizard	In Exchange 2010, you use the Port Managed Folder wizard to create retention tags based on managed folder and managed content settings. In Exchange 2013, the Exchange admin center doesn't include this functionality. You can use the **New-RetentionPolicyTag** cmdlet with the *ManagedFolderToUpgrade* parameter to create a retention tag based on a managed folder.

Unified Messaging and voice mail

Feature	Comments and mitigation
Directory lookups using Automatic Speech Recognition (ASR)	In Exchange 2010, Outlook Voice Access users can use speech inputs using Automatic Speech Recognition (ASR) to search for users listed in the directory. Speech inputs could be also used in Outlook Voice Access to navigate menus, messages, and other options. However, even if an Outlook Voice Access user is able to use speech inputs, they have to use the telephone key pad to enter their PIN, and navigate personal options. In Exchange 2013, authenticated and non-authenticated Outlook Voice Access users can't search for users in the directory using speech inputs or ASR in any language. However, callers that call into an auto attendant can use speech inputs in multiple languages to navigate auto attendant menus and search for users in the directory.

Tools

Feature	Comments and mitigation
Exchange Best Practice Analyzer	In Exchange 2010, the Exchange Best Practice Analyzer examined your Exchange deployment and determined whether the configuration was in line with Microsoft best practices. In Exchange 2013, the Exchange Best Practice Analyzer has been replaced by the Office 365 Best Practices Analyzer for Exchange Server 2013.

Mail flow troubleshooter	In Exchange 2010, the mail flow troubleshooter assisted you in troubleshooting common mail flow problems. In Exchange 2013, the mail flow troubleshooter has been retired. You can now use the messaging tracking feature in EAC in Exchange 2013. For more information, see Track messages with delivery reports.
Performance monitor	In Exchange 2010, the Performance Monitor was included in the Exchange toolbox to allow you to collect information about the performance of your messaging system. Performance Monitor is commonly used to view key parameters while troubleshooting performance problems. In Exchange 2013, the Performance Monitor has been retired from the toolbox. You can still find the Performance Monitor under **Administrative Tools** in Windows Server 2008.
Performance troubleshooter	In Exchange 2013, the Performance Troubleshooter has been retired.
Routing Log Viewer	In Exchange 2013, the routing log viewer has been retired.

Discontinued features from Exchange 2007 to Exchange 2013

This section lists the Exchange Server 2007 features that are no longer available in Exchange 2013.

APIs and development

Feature	Comments and mitigation
Exchange WebDAV	Use Exchange Web Services or EWS Managed

	API. Alternatively, you can maintain an Exchange 2007 server for mailboxes that are managed by applications that use WebDAV. For more information, see Migrating from WebDAV.

Architecture

Feature	Comments and mitigation
Storage groups	Exchange 2013 no longer uses the storage group construct, and instead you simply manage mailbox databases. For more information, see Manage mailbox databases in Exchange 2013.
Extensible Storage Engine (ESE) streaming backup APIs	Exchange 2013 supports only Exchange-aware Volume Shadow Copy Service (VSS)-based backups. Exchange 2013 does include a plug-in for Windows Server Backup that enables you to backup and restore data. For information, see Backup, restore, and disaster recovery.
User Datagram Protocol (UDP) notifications	Support for User Datagram Protocol (UDP) notifications is removed from Exchange 2013. This affects the user experience when Outlook 2003 clients connect to their mailboxes on an Exchange 2013 server. For more information, see Microsoft Knowledge Base article 2009942, Folders take a long time to update when an Exchange Server 2010 user uses Outlook 2003 in online mode.

High availability

Feature	Comments and mitigation

Cluster continuous replication (CCR)	Exchange 2013 uses database availability groups (DAGs) and mailbox database copies. For information, see High availability and site resilience.
Local continuous replication (LCR)	Exchange 2013 uses DAGs and mailbox database copies. For information, see High availability and site resilience.
Standby continuous replication (SCR)	Exchange 2013 uses DAGs and mailbox database copies. For information, see High availability and site resilience.
Single copy cluster (SCC)	Exchange 2013 uses DAGs and mailbox database copies. For information, see High availability and site resilience.
Setup /recoverCMS	Exchange 2013 uses Setup /m:recoverServer. For information, see Recover an Exchange Server.
Clustered mailbox servers	Exchange 2013 uses DAGs and mailbox database copies. For information, see High availability and site resilience.

Client access

Feature	Comments and mitigation
Client authentication using Integrated Windows authentication (NTLM) for POP3 and IMAP4 users	NTLM isn't supported for POP3 or IMAP4 client connectivity in Exchange 2013. Connections from POP3 or IMAP4 client programs using NTLM will fail. If you're running the RTM version of Exchange 2013, the recommended alternative to NTLM is to use Plain Text Authentication with SSL. If you're using Exchange 2013, to use NTLM,

	you must retain an Exchange 2007 server in your Exchange 2013 organization.

Outlook Web App and Outlook

Feature	Comments and mitigation
Document access	Outlook Web App can't be used to access Microsoft SharePoint document libraries and Windows file shares.
Message flags	The ability to set a custom date on a message flag isn't available in Outlook Web App 2013. You can use Outlook to set custom dates.
Spell check	Outlook Web App uses the spell check features in your Web browser.
Search Folders	The ability for users to use Search folders isn't currently available in Outlook Web App.
Maximum Cached Views	Exchange 2007 supported modifying the maximum number of cached views per database (msExchMaxCachedViews) parameter for Outlook clients. Exchange 2013 no longer uses this parameter.

Recipient-related features

Feature	Comments and mitigation
Export-Mailbox and **Import-Mailbox** cmdlets	In Exchange 2013, use export requests or import requests. For more information, see Mailbox import and export requests.
Move-Mailbox cmdlet set	In Exchange 2013, use move requests to move mailboxes. For information, see Mailbox moves in Exchange 2013.

ISInteg	In Exchange 2013, use New-MailboxRepairRequest.

Messaging policy and compliance

Feature	Comments and mitigation
Managed Folders	In Exchange 2007, you use managed folders for messaging retention management (MRM). In Exchange 2013, managed folders aren't supported. You must use retention policies for MRM. **Note:** Cmdlets related to managed folders are still available. You can create managed folders, managed content settings and managed folder mailbox policies, and apply a managed folder mailbox policy to a user, but the MRM assistant skips processing of mailboxes that have a managed folder mailbox policy applied.

Unified Messaging and voice mail

Feature	Comments and mitigation
Directory lookups using Automatic Speech Recognition (ASR) for Outlook Voice Access	In Exchange 2007, Outlook Voice Access users can use speech inputs using Automatic Speech Recognition (ASR) in English (US) – (en-US) to search for users listed in the directory. Speech inputs could be also used in Outlook Voice Access to navigate menus, messages, and other options. However, even if an Outlook Voice Access user is able to use speech inputs, they have to use the telephone key pad to enter their PIN, and navigate personal options. In Exchange 2013, authenticated and non-authenticated Outlook Voice Access users can't search for users in the directory using speech

	inputs or ASR in any language. However, callers that call into an auto attendant can use speech inputs in multiple languages to navigate auto attendant menus and search for users in the directory.

What's new for Outlook Web App in Exchange 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-09-04*

For Microsoft Exchange Server 2013, we've added several new features to Microsoft Outlook Web App and updated its design.

Note:

For more details about using Outlook Web App in your Exchange Server 2013 organization, see Outlook Web App.

Apps in Outlook Web App

We've added several apps for Outlook: Bing Maps, Suggested Appointments, and Action Items. These apps are integrated with Outlook and Outlook Web App and extend the information and functionality of messages and calendar items.

Apps in Outlook attempt to anticipate your needs and automatically propose actions you might want to take by using the contents of the email message. For example, if an email message contains a street address, the Bing Maps app offers you a Bing tab with a quick link to a map and directions. Or, if a phrase in the email message suggests a possible action item, the Action Items app creates a suggested Task for your review. An offer to meet is suggested as an Appointment to be added to your calendar, thanks to the Suggested Appointments app.

Apps for Outlook aren't dependent on the version of Exchange Server that you're using. You won't have to worry about breaking or losing any apps for Outlook that you have added when you upgrade Exchange servers or move to a new Exchange version.

Administrators can use the Exchange admin center (EAC) to manage the apps available to users in the organization. Users can then manage their apps. Administrators can also allow users to

download apps from Office.com. For more information about the EAC, see Exchange admin center in Exchange 2013.

In addition, we encourage third-party developers to create additional apps for Outlook and then offer them at Office.com. To learn more, see Build Apps for Office for background information and Mail apps for Outlook for detailed information about building apps for Outlook.

People

- Now, users can link multiple entries for the same person and view the information in a single contact card. For example, if a user has two entries for Holly Holt in his Contacts folder, one entry copied from the organization's address list and one entry that he added manually, he can link the two entries in his Contacts folder and view all the information in one place. Contact linking is done automatically, but the user can also manually link and unlink contacts.
- Connected accounts have been extended to include the ability to connect to a user's LinkedIn account. After the link is established, Outlook Web App automatically adds the user's LinkedIn contacts to the Contacts folder.

Calendar

- Users can now view multiple calendars in a merged view. Entries from each calendar have their own color, making it easy for users to identify which calendar an entry belongs to. In the day view, users can view multiple calendars in a merged view or in separate columns.
- The month view now includes an agenda for the selected day, providing users with helpful information as they review the day's activities.
- In all calendar views, users can click an item to view a pop-up of the item's details. In addition to the details, controls are now available to accept or decline the item if it's a meeting, to edit or delete if it's an appointment, or, if a meeting item, to join the meeting if an online meeting link is included.

Tablets and smartphones

Outlook Web App emphasizes a streamlined user interface that also supports the use of touch, enhancing the mobile device experience with Exchange.

Supported browsers

To experience all Outlook Web App features, use one of the operating system and browser combinations labeled "Best", as noted in the tables below. Outlook Web App is supported by many operating system and web browser combinations, but not all Outlook Web App features are available in all combinations. Some browsers support only the light version of Outlook Web App.

Supported browsers on desktop and laptop computers

In the table below, the following definitions apply:

- Best: All Outlook Web App features are supported.
- Good: Most Outlook Web App features are supported.
- Light: The browser displays the light version of Outlook Web App

Windows operating system and browser combination

Web browser	Windows XP and Windows Server 2003	Windows Vista and Windows Server 2008	Windows 7	Windows 8
Internet Explorer 7	Light	Not available	Not available	Not available
Internet Explorer 8	Light	Light	Light	Not available
Internet Explorer 9	Not available	Best	Best	Not available
Internet Explorer 10	Not available	Not available	Best – plus offline access	Best – plus offline access
Internet Explorer 11	Not available	Not available	Best – plus offline access	Best – plus offline access
Firefox 17 or later	Good	Good	Best	Best
Safari 5 or later	Light	Light	Light	Light
Chrome 24 or later	Good – plus offline access	Good – plus offline access	Best – plus offline access	Best – plus offline access

Note:

In previous versions, Outlook Web App had a built-in spell checker. In Exchange Server 2013, Outlook Web App relies on the web browser for spell checking, which Internet Explorer prior to version 10 doesn't provide.

Note:

Office 365 users will be limited to the light version of Outlook Web App when using Internet Explorer 8. Users whose mailboxes are on a locally managed Exchange server will continue to see the standard version of Outlook Web App when using Internet Explorer 8, but may experience slow or otherwise unsatisfactory performance.

Other Windows operating system and browser combination

Web browser	Mac OS X v10.5	Mac OS X v10.6 and v10.7	Linux
Firefox 17 or later versions	Best	Best	Best
Safari 6 or later versions	Best – plus offline access	Best – plus offline access	Not available
Chrome 24 or later versions	Best – plus offline access	Best – plus offline access	Best – plus offline access

Note:

Operating system and browser combinations not listed display the light version of Outlook Web App.

Supported browsers for tablets and smartphones

You can use the web browser on a tablet or smartphone to sign in to Outlook Web App. The available Outlook Web App features depends on the operating system and browser combination in use, as follows:

- Best: All Outlook Web App features for smartphones and tablets are supported.
- Light: The browser displays the light version of Outlook Web App.

Outlook Web App features available on tablets and smartphones

Device	Application	Support
Windows 8 tablet	Web browser	Best
iOS 6 or later for iPhone 4s or later	Web browser	Best
iOS 6 or later for iPad 2 or later	Web browser	Best
All other smartphones and tablets	Web browser	Light

OWA for Devices app

The OWA for Devices app lets users in an Exchange 2013 on-premises deployment with an Office 365 mailbox or in an Office 365 only organization use their iPhone or iPad access their mailbox. The OWA for iPhone and OWA for iPad apps simplify signing in to their mailbox and allows them access their mailbox even when they don't have an Internet connection. The OWA for iPhone or OWA for iPad apps are recommended instead of using your iPhone's or iPad's browser. For Exchange on-premises deployments, you need to enable push notifications for OWA for Devices to work, see Configuring push notifications proxying for OWA for Devices.

You can download the OWA for iPhone and OWA for iPad apps from the Apple App Store by searching for OWA for iPhone or OWA for iPad or download them from OWA for iPhone in the Apple Store or OWA for iPad in the Apple Store. The table below shows the versions of iPad and iPhone that are supported.

Device	**OS version required**
iPhone 4S, iPhone 5, iPhone 5c or iPhone 5s. This app is optimized for iPhone 5.	iOS 6 or later versions
iPad Wi-Fi (3rd generation), iPad Wi-Fi + Cellular (3rd generation), iPad Wi-Fi (4th generation), iPad Wi-Fi + Cellular (4th generation), iPad mini Wi-Fi, iPad mini Wi-Fi + Cellular, iPad Air, iPad Air Wi-Fi + Cellular, iPad mini with Retina display, iPad mini with Retina display Wi-Fi + Cellular	iOS 6 or later versions

Unavailable features

The following Outlook Web App features are currently unavailable in Exchange 2013. Some of these features may be included in a future release.

- **Distribution list moderation** The ability to moderate distribution lists from Microsoft Outlook Web App isn't currently available in Exchange 2013.
- **Custom date on message flags** The ability to set a custom date on a message flag isn't available in Outlook Web App 2013. You can use Outlook to set custom dates.
- **Reading pane at the bottom of the window** The option to display the reading pane at the bottom of the Outlook Web App window isn't currently available in Exchange 2013.
- **Reply to embedded email messages** The ability for users to reply to email messages sent as attachments isn't currently available in Exchange 2013.

- **Search folders** The ability for users to use Search folders isn't currently available in Exchange 2013.
- **Access to legacy public folders** The ability for users to access public folders located on servers running previous versions of Exchange isn't currently available in Exchange 2013.

What's new for Unified Messaging in Exchange 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-05*

In Microsoft Exchange Server 2013, we're enhancing earlier releases of Exchange by introducing new features and architectural changes. Unified Messaging (UM) in Exchange 2013 includes the same feature set as Exchange 2010 and Exchange 2007, however Unified Messaging is no longer a separate server role. It's now a component of the voice-related features offered in Exchange 2013.

Changes in the Voice architecture

The architecture of Exchange 2013 is different than it was in Exchange 2010 and Exchange 2007. In previous versions of Exchange UM, all the components for Unified Messaging were included on a server that had the UM server role installed. In Exchange 2013, all the Unified Messaging components are split between a Client Access server running the Microsoft Exchange Unified Messaging Call Router service and a Mailbox server running the Microsoft Exchange Unified Messaging service. All the functionality, including the services and worker processes for Unified Messaging, is located on each Mailbox server, with the exception of the Client Access server running the Microsoft Exchange Unified Messaging Call Router service, which proxies incoming calls to the Mailbox server. For details, see Voice architecture changes.

Support for IPv6

Internet Protocol version 6 (IPv6) is the most recent version of the Internet Protocol. IPv6 is intended to correct many of the shortcomings of IPv4, which was the previous version of the IP. As in Exchange 2010, Exchange 2013 Client Access and Mailbox servers fully support IPv6 networks. For details, see IPv6 support in Unified Messaging.

Support for UCMA 4.0 API

Since Service Pack 1 for Exchange 2010, the Unified Messaging role has relied on Unified Communications Managed API v2.0 (UCMA) for signaling and media. Therefore, UCMA 2.0 is a prerequisite for Exchange 2010 UM setup. UCMA 2.0 is downloaded separately and deployed manually by administrators on Unified Messaging servers running Exchange 2010 SP1 or a later version. For Exchange 2013, UCMA 4.0 is required. However, given that the UM server is no longer a separate server role in Exchange 2013, now it's the Client Access and Mailbox servers that require UCMA 4.0.

UCMA 4.0 supports new features in Unified Messaging, such as using the same version of the Speech Engine for both Text-to-Speech (TTS) and Automatic Speech Recognition (ASR). The platform that's used for Exchange 2013, .NET 4.0, includes a single installer file and enables backward compatibility with Exchange 2010 and Exchange 2007 UM servers.

In Exchange 2010 SP2 and SP1, UCMA 2.0 installation is required prior to installing the service pack on a Unified Messaging server. However, UCMA 2.0 had several limitations. UCMA 4.0 corrects many of the shortcomings. In Exchange Server 2013, UM continues to use UCMA. However, the move to the newest version of UCMA provides these benefits:

- The newest build of UCMA incorporates hotfixes and patches.
- UCMA requires .NET 4.0, which is the platform used by Exchange Server 2013. (UCMA 2.0 doesn't support .NET 4.0.)
- UCMA 4.0 supports IPv6.
- Simplified and automated deployment of UCMA 4.0. Exchange 2013 Setup performs a single check for UCMA 4.0.
- UCMA 4.0 setup includes all prerequisites for Exchange 2013.

Note:

UCMA 4.0 is installed when you're installing Exchange 2013. For details about UCMA 4.0 and setup requirements, see Exchange 2013 prerequisites. To upgrade to the most recent version of UCMA, you must first uninstall any previous versions of UCMA that are installed using Add/Remove programs.

Improvements to Voice Mail Preview

Some enhancements to the speech-related services are offered for Exchange Server 2013 UM via the Speech Engine 11.0 and UCMA 4.0. Grammar generation and language improvements are included. In addition, Exchange Server 2013 UM includes several enhancements to the UI and improvements for confidence and accuracy for Voice Mail Preview. For details, see Voice mail preview enhancements.

Enhanced caller ID support

In previous releases of Exchange Unified Messaging, a UM server that took a call used caller ID to try to look up the identity of the calling party. This search extended across Active Directory and the UM user's personal contacts stored in their mailbox.

Exchange users are often annoyed by failures to identify Exchange or personal contacts from their caller ID. Until now, only the default contact folder in Exchange UM was used for this search. But Exchange Server 2013 users are likely to have contacts aggregated from external social networks or contacts in unique folders that the users have created manually. Exchange 2013 supports contact aggregation from external social networks, provides intelligence to link multiple contacts referring to the same person, and uses that data to present person-centric (rather than contact-centric) views. The contacts that are aggregated from external networks are placed in contact folders along with any additional contact folders that users created.The features in Exchange 2013 UM extend the scope of the search to include the user's other Exchange and personal contact folders that were created manually.

Caller ID look-up is integrated with contact aggregation, so that it searches across external contacts. The PersonID property, where present and with a non-null value, improves the user experience for caller ID resolution by suppressing duplicate matches to contacts that are associated with the same person. Because the PersonID property is the same on both results, UM treats this as a match to a single contact.

Enhancements to speech platform and speech recognition

Exchange Server 2013 UM introduces some enhancements to the speech platform and speech recognition, including the following:

- Enhancements and improved accuracy for Voice Mail Preview.
- Support for the Microsoft Speech Platform – Runtime (Version 11.0).
- Speech grammar generation using the system mailbox for an organization.

Exchange Unified Messaging uses static and dynamic speech grammars to recognize commands, names of contacts in the global address list, and names of personal contacts in the user's mailbox. Today, in Exchange Server 2013, every Mailbox server running the Microsoft Exchange Unified Messaging service generates grammars for all UM languages installed on it and stores them in directories. Every Mailbox server stores every possible grammar, which it generates based on the number of dial plans, auto attendants, and the UM languages that are installed.

Grammar files are used as inputs to the speech recognition process and are generated on a periodic basis. The GGG.exe command in Exchange 2007 and Exchange 2010 allowed you to manually update the grammar files without waiting for the scheduled update. In Exchange Server 2013, to address ASR grammar-generation scalability issues for UM, the speech GAL grammar generation no longer happens on the server with the Unified Messaging server role installed. Instead, it happens periodically using the Mailbox Assistant, on the Mailbox server running the Microsoft Exchange Unified Messaging service that hosts the organization's arbitration mailbox. The GAL speech grammar file is stored in the arbitration mailbox for an organization and then later downloaded to all Mailbox servers in that Exchange organization. By default, the Mailbox Assistant runs every 24 hours. You can adjust the frequency by using the **Set-MailboxServer** cmdlet.

Cmdlet updates

For Exchange 2013, many UM cmdlets have been brought over from Exchange 2010, but there have been changes in some of those cmdlets, and new cmdlets have been added for new functionality. For details, see Unified Messaging cmdlet updates.

What's new for transport rules

Exchange Server 2013 > What's new in Exchange 2013 >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-07-03*

In Microsoft Exchange Server 2013, several improvements have been made to transport rules. This topic provides a brief overview of some of the key changes and enhancements. To learn more about transport rules, see Transport rules.

Support for data loss prevention policies

Data loss prevention (DLP) features in Exchange 2013 can help organizations reduce unintentional disclosure of sensitive data. Transport rules have been updated to support creating rules that accompany and enforce DLP policies. To learn more about DLP support in transport rules, see the following topics:

Integrating sensitive information rules with transport rules

Data loss prevention

New predicates and actions

The functionality of transport rules has been extended via the addition of new predicates and actions. Each predicate listed below can be used as a condition or an exception when you're creating transport rules.

For detailed information about using these new predicates and actions, see Transport rule conditions (predicates) and Transport rule actions.

New predicates

AttachmentExtensionMatchesWords Used to detect messages that contain attachments with specific extensions.

AttachmentHasExecutableContent Used to detect messages that contain attachments with executable content.

HasSenderOverride Used to detect messages where the sender has chosen to override a DLP policy restriction.

MessageContainsDataClassifications Used to detect sensitive information in the message body and any of the attachments. For a list of data classifications available, see Sensitive information types inventory.

MessageSizeOver Used to detect messages whose overall size is greater than or equal to the specified limit.

SenderIPRanges Used to detect messages sent from a specific set of IP address ranges.

New actions

GenerateIncidentReport Generates an incident report that is sent to a specified SMTP address. The action also has a parameter called *IncidentReportOriginalMail* that accepts one of two values: IncludeOriginalMail or DoNotIncludeOriginalMail.

NotifySender Controls how the sender of a message that goes against a DLP policy is notified. You can choose to simply inform the sender and route the message normally, or you can choose to reject the message and notify the sender.

StopRuleProcessing Stops the processing of all subsequent rules on the message.

ReportSeverityLevel Sets the specified severity level in the incident report. Values for the action are: Informational, Low, Medium, High, and Off.

RouteMessageOutboundRequireTLS Requires Transport Layer Security (TLS) encryption when routing this message outside your organization. If TLS encryption isn't supported, the message is rejected and not delivered.

Other changes in Transport rules

- **Support for extended regular expression syntax** Transport rules in Exchange 2013 are based on the Microsoft.NET Framework regular expression (regex) functionality and now support extended regular expression syntax.

- **Transport rules agent invocation** The key architectural change in Exchange 2013 for Transport rules is the Transport Rules Agent is invoked on onResolvedMessage. In previous versions of Exchange, the Rules Agent was invoked on onRoutedMessage. This change allowed us to add new actions, such as requiring TLS, which can change how a message is routed. To learn more about the transport rules architecture in Exchange 2013, see Transport rules.
- **Detailed Transport rule information in message tracking logs** Detailed information about Transport rules is now included in message tracking logs. The information includes which rules were triggered for a specific message and the actions taken as a result of processing those rules.
- **New rule monitoring functionality** Exchange 2013 monitors Transport rules that are configured and measures the cost of running these rules both when you're creating the rule and also during regular operation. Exchange can detect and generate alerts for rules that are causing delays in mail delivery.
- **Rule size limit changes** The following changes have been made to Transport rule limits:
  - Maximum size of each rule is 4 kilobytes (KB).
  - The character limit for all regular expressions used in all Transport rules is 20 KB. This limit applies to total number of characters used by all regular expressions, including keywords.

Release notes for Exchange 2013

Exchange Server 2013 > What's new in Exchange 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-25*

Welcome to Microsoft Exchange Server 2013! This topic contains important information that you need to know to successfully deploy Cumulative Update 6 for Exchange 2013. Please read this topic completely before beginning your deployment.

This topic contains the following sections:

- Setup and deployment
- Mailbox
- Public folders
- Mail flow
- Client connectivity
- Exchange 2010 coexistence

Setup and deployment

- **Setup incorrectly requests .NET Framework 4.0** If you attempt to install Exchange 2013 without .NET Framework installed on the computer, Setup incorrectly requests that you install .NET Framework 4.0 when, in fact, .NET Framework 4.5 is required.

To work around this issue, install .NET Framework 4.5. You don’t need to install .NET Framework 4.0.

For a complete list of prerequisites, see Exchange 2013 prerequisites.

- **Exchange XML application configuration files are overwritten during cumulative update installation** Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update or Service Pack. Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange Cumulative Update or Service Pack.
- **MAPI virtual directory isn't created during server recovery** When you run Setup.exe with the *RecoverServer* switch on a server that has the Client Access server role installed, the MAPI virtual directory isn't created. If the MAPI virtual directory doesn't exist, clients that use the MAPI over HTTP protocol to connect to the Exchange server, such as Outlook, won't be able to connect.

Note:
This is only an issue if the MAPI over HTTP protocol is enabled on your Client Access servers. It's disabled by default. If MAPI over HTTP is disabled, clients use the RPC over HTTP protocol instead.

To resolve this issue, follow the steps in Knowledge Base article KB2931223 (MAPI virtual directory is missing from default Web Site node).

For more information about how to install Exchange 2013, see Planning and deployment.

Mailbox

- **Mailbox size increase when migrating from previous Exchange versions** When you move a mailbox from a previous version of Exchange to Exchange 2013, the mailbox size reported may increase 30 percent to 40 percent. Disk space used by the mailbox database has not increased, only the attribution of space used by each mailbox has increased. The increase in mailbox size is due to the inclusion of all item properties into quota calculations, providing a more accurate computation of space consumed by items within their mailbox. This increase may cause some users to exceed their mailbox size quotas when their mailbox is moved to Exchange 2013.

To prevent users from exceeding their mailbox size quotas, increase the database or mailbox quota values to accommodate the new quota calculation. To configure database or mailbox quota values, use the *IssueWarningQuota*, *ProhibitSendQuota*, and *ProhibitSendReceiveQuota* parameters on the **Set-MailboxDatabase** and **Set-Mailbox** cmdlets, respectively.

- **Outlook 2007 and Outlook 2010 clients may be unable to download the Offline Address Book** If the Offline Address Book (OAB) internal URL isn't accessible from the Internet, Outlook 2007 and Outlook 2010 clients may be unable to download the OAB.

To work around this issue for Outlook 2007 and Outlook 2010 clients, make the OAB internal URL accessible from the Internet. Outlook 2013 isn't affected by this issue.

- **Installing Exchange 2013 in an existing Exchange organization may cause all clients to download the OAB** Installing the first Exchange 2013 server into an existing Exchange 2007 or Exchange 2010 organization may cause all clients in the organization to download a new copy of

the OAB, resulting in network saturation and server performance issues. This issue occurs because Exchange 2013 creates a new default OAB in the organization that supersedes the Exchange 2007 or Exchange 2010 OAB. Mailboxes that don't have a specific OAB assigned, or that are located on a mailbox database that doesn't have a specific OAB assigned, will download the new default OAB.

To prevent clients from downloading a new copy of the OAB when Exchange 2013 is installed, assign an OAB to every mailbox or to the mailbox database the mailboxes are located on. This must be done prior to Exchange 2013 being installed in the organization.

- **Users may be routed to an OAB generation mailbox that's not responsible for the requested OAB** Exchange 2013 CU5 and later CUs have changed how OABs are linked to OAB generation mailboxes. This change makes it possible for a user to be routed to an OAB generation mailbox that isn't responsible for the OAB that the user is requesting. This can happen if all of the following are true:
  - You have more than one OAB generation mailbox in your organization.
  - You upgrade the Mailbox servers that host OAB generation mailboxes before you upgrade your Client Access servers.
  - You're upgrading your Exchange 2013 servers from a release prior to CU5 to a later release (for example, upgrading from Exchange 2013 CU3 to Exchange 2013 CU6).
  - Your Client Access servers are running a release prior to CU5.

To work around this issue, make sure that you upgrade your Client Access servers to Exchange 2013 CU6 before you upgrade your Mailbox servers. This will make sure the Client Access servers know how to proxy the requests to the OAB generation mailbox that is responsible for generating the user's OAB..

To read more about the OAB changes in Exchange 2013 CU5, see OAB Improvements in Exchange 2013 Cumulative Update 5.

Public folders

- **Unauthorized senders can no longer send messages to mail-enabled public folders** Prior to Exchange 2013 CU6, unauthorized senders could send messages to mail-enabled public folders. This allowed the possibility for external senders to send mail to mail-enabled public folders regardless of the permissions set on the public folder.

Starting with Exchange 2013 CU6, if you want external senders to send mail to a mail-enabled public folders, the **Anonymous** user needs to be granted at least the **Create Items** permission. If you've set up mail-enabled public folders and haven't done this, external senders will receive a delivery failure notification and the messages won't be delivered to the mail-enabled public folder.

You can use the Shell or Outlook to set the permissions on the Anonymous user. To read more about how to set permissions on the Anonymous user, see Mail-enable or mail-disable a public folder.

- **Legacy public folders can't be accessed via Exchange Web Services** Public folders that are located on Exchange 2007 or Outlook 2010 servers can't be accessed by clients that connect to Exchange 2013 using Exchange Web Services (EWS). Clients that use EWS include Mac Outlook

and Outlook Web App. If an EWS client attempts to access a legacy public folder, they'll receive an error.

The only workaround at this time is to migrate legacy public folders to Exchange 2013. However, there are some issues that must be considered before you migrate your public folders. For more information, see Public folders.

Mail flow

- **TransportAgent cmdlets on Client Access servers require local Windows PowerShell** An issue exists with the ***-TransportAgent** cmdlets that prevents those cmdlets from installing, uninstalling, and managing transport agents on Client Access servers using the Exchange Management Shell. To install, uninstall, and manage transport agents on Client Access servers, you must manually load the Exchange Windows PowerShell snap-in and then run the ***-TransportAgent** cmdlets. If you attempt to install, uninstall, or manage transport agents using the Exchange Management Shell, your changes will be applied to the Exchange 2013 Mailbox server you're connected to.

To install, uninstall, or manage transport agents on Client Access servers, do the following on the Client Access server you want to manage:

Caution:

Loading the `Microsoft.Exchange.Management.PowerShell.SnapIn` Windows PowerShell snap-in and running cmdlets other than the ***-TransportAgent** cmdlets is not supported and may result in irreparable damage to your Exchange deployment.
You must be a local Administrator on the Client Access server where you want to install, uninstall, or manage transport agents. We do not support the modification of access control lists (ACLs) on Exchange files, directories, or Active Directory objects.

Important:

Perform the following procedure on Client Access servers only. You don't need to load the Exchange Windows PowerShell snap-in if you want to manage transport agents on Mailbox servers.

1. Open a new Windows PowerShell window.
2. Run the following command.

```
Add-PSSnapin
Microsoft.Exchange.Management.PowerShell.SnapIn
```

3. Perform transport agent management tasks as normal.
4. Repeat this procedure on each Client Access server you want to manage.

Client connectivity

- **NTLM authentication fails for non-domain joined clients** Authentication between a client, such as Windows Live Mail, and Exchange 2013 may fail when the following conditions are true:
  - Then authentication method the client uses is NTLM.

- The computer isn't joined to the domain.

To work around this issue, you can do one of the following:

- Join the computer the client is running on to the domain.
- Change the authentication type the client uses from NTLM to Basic Auth over TLS.

- **GSSAPI authentication fails when used with the Send-MailMessage cmdlet** Generic Security Service Application Program Interface (GSSAPI) authentication may fail when the **Send-MailMessage** cmdlet, which is included with default installs of Windows PowerShell, is used to send authenticated mail to Exchange 2013. When this happens, you'll see an entry in the **Application** event log on the Exchange 2013 Client Access server that received the connection with the following information:
  - **Source** MSExchangeFrontEndTransport
  - **Event ID** 1035
  - **Description** Inbound authentication failed with error `IllegalMessage` for Receive connector Client Frontend <*server name*>. The authentication mechanism is Gssapi. The source IP address of the client who tried to authenticate to Exchange is [<*client IP address*>].

To work around this issue, you need to remove the `Integrated` authentication method from the client receive connector on your Exchange 2013 Client Access servers. To remove the `Integrated` authentication method from a client receive connector, run the following command on each Exchange 2013 Client Access server that could receive connections from computers running the **Send-MailMessage** cmdlet:

```
Set-ReceiveConnector "<server name>\Client Frontend <server name>" -AuthMechanism Tls, BasicAuth, BasicAuthRequireTLS
```

- **MAPI over HTTP may experience poor performance when you upgrade to Exchange 2013 SP1** If you upgrade from an Exchange 2013 cumulative update to Exchange 2013 SP1 and enable MAPI over HTTP, clients that connect to an Exchange 2013 SP1 server using the protocol may experience poor performance. This is because required settings aren't configured during an upgrade from a cumulative update to Exchange 2013 SP1. This issue doesn't occur if you upgrade to Exchange 2013 SP1 from Exchange 2013 RTM or if you install a new Exchange 2013 SP1 or later server.

Note:

This is only an issue if the MAPI over HTTP protocol is enabled on your Client Access servers. It's disabled by default. If MAPI over HTTP is disabled, clients use the RPC over HTTP protocol instead.

To work around this issue, do the following:

- On servers running the Client Access server role, run the following commands in a Windows Command Prompt:

```
set AppCmdLocation=%windir%\System32\inetsrv
set ExchangeLocation=%ProgramFiles%\Microsoft\Exchange Server\V15
```

```
%AppCmdLocation%\appcmd.exe SET AppPool
"MSExchangeMapiFrontEndAppPool" /CLRConfigFile:"%
ExchangeLocation%\bin
\MSExchangeMapiFrontEndAppPool_CLRConfig.config"
%AppCmdLocation%\appcmd.exe RECYCLE AppPool
"MSExchangeMapiFrontEndAppPool"
```

- On servers running the Mailbox server role, run the following commands in a Windows Command Prompt:

```
set AppCmdLocation=%windir%\System32\inetsrv
set ExchangeLocation=%ProgramFiles%\Microsoft\Exchange
Server\V15
%AppCmdLocation%\appcmd.exe SET AppPool
"MSExchangeMapiMailboxAppPool" /CLRConfigFile:"%
ExchangeLocation%\bin
\MSExchangeMapiMailboxAppPool_CLRConfig.config"
%AppCmdLocation%\appcmd.exe RECYCLE AppPool
"MSExchangeMapiMailboxAppPool"
%AppCmdLocation%\appcmd.exe SET AppPool
"MSExchangeMapiAddressBookAppPool" /CLRConfigFile:"%
ExchangeLocation%\bin
\MSExchangeMapiAddressBookAppPool_CLRConfig.config"
%AppCmdLocation%\appcmd.exe RECYCLE AppPool
"MSExchangeMapiAddressBookAppPool"
```

Exchange 2010 coexistence

- **Requests to access Exchange 2010 mailboxes may not work when proxied through Exchange 2013 Client Access servers** In some situations, the proxy request between the Exchange 2013 and Exchange 2010 Service Pack 3 (SP3) Client Access servers without any update rollups installed may not work correctly and an error appears. This can happen if all of the following conditions are true:
  - A user with an Exchange 2013 mailbox tries to open an Exchange 2010 mailbox using one of the following methods:
    - The **Open Another Mailbox** option in Outlook Web App **-OR-**
    - The **Another user** option in the Exchange admin center
  - The Client Access server the user connected to is running Exchange 2013.
  - The Exchange 2010 Client Access server was upgraded to Exchange 2010 SP3 from the release to manufacturing (RTM) version of Exchange 2010 or a previous Exchange 2010 service pack.

If all the conditions above are true, the user won't be able to access the other user's Exchange 2010

Outlook Web App options and a blank page may appear.

To work around this issue, install Exchange 2010 SP3 Update Rollup 1 or later on each Exchange 2010 server.

Updates for Exchange 2013

Exchange Server 2013 > What's new in Exchange 2013 >

Applies to: *Exchange Server 2013, Exchange Server, Exchange Online*

Topic Last Modified: *2014-08-27*

With Microsoft Exchange Server 2013, we've changed the way we deliver hotfixes and service packs. Instead of the priority-driven hotfix release and rollup update model used by previous versions of Microsoft Exchange, Exchange 2013 now adheres to a scheduled delivery model. In this model, cumulative updates are released quarterly.

For more information about Exchange 2013 Cumulative Update 6 (CU6), see Released: Exchange Server 2013 Cumulative Update 6 and Public Folder Updates in Exchange 2013 CU6: Improving Scale and More.

To download Exchange 2013, see Microsoft Exchange Server 2013 Cumulative Update 6.

Note:
Exchange 2013 cumulative updates and service packs include all the changes from previous updates and are full builds of the product. You don't have to install the previous cumulative updates or service packs to get the latest features.

For more information about updates as they relate to Exchange 2013, including an extensive FAQ, see Servicing Exchange 2013 and “Servicing Model Update” in Released: Exchange Server 2013 Cumulative Update 2.

The following table links to "What's New" topics and Exchange Team blog posts for each release of Exchange 2013. Click on the link for a release to learn more about the changes included in that release.

Exchange 2013 Release	Link
Exchange 2013 CU5	Released: Exchange Server 2013 Cumulative Update 5
Exchange 2013 SP1	New features and improvements included in Exchange 2013 SP1 Released: Exchange Server 2013 Service Pack 1

Exchange 2013 CU3	Released: Exchange Server 2013 Cumulative Update 3
Exchange 2013 CU2	Released: Exchange Server 2013 Cumulative Update 2
Exchange 2013 CU1	Released: Exchange Server 2013 Cumulative Update 1
Exchange 2013 RTM	What's new in Exchange 2013 Exchange Server 2013 Reaches General Availability

New features and improvements included in Exchange 2013 SP1

Windows Server 2012 R2 support

Windows Server 2012 R2 is now a supported operating system in Exchange 2013 SP1. Exchange 2013 SP1 also supports installation in Active Directory environments running Windows Server 2012 R2. For more information, see Exchange 2013 system requirements.

Edge Transport servers return

Edge Transport servers minimize attack surface by handling all Internet-facing mail flow, which provides SMTP relay and smart host services for your Exchange organization, including connection filtering, attachment filtering and address rewriting. For more information, see Edge Transport servers.

OWA Junk Email Reporting

Outlook Web App customers can report missed spam in the inbox (false negative) and misclassified as spam (false positive) messages to Microsoft for analysis by using its built-in junk email reporting options. Depending on the results of the analysis, we can then adjust the anti-spam filter rules for our Exchange Online Protection (EOP) service. For more information, see **Junk Email Reporting in OWA**.

S/MIME for Message Signing and Encryption

Exchange 2013 SP1 now supports S/MIME-based message security with Outlook Web App. Secure/Multipurpose Internet Mail Extensions (S/MIME) allows people to help protect sensitive information by sending signed and encrypted email within their organization. Administrators can enable S/MIME for mailboxes by synchronizing user certificates and then configuring Outlook Web App to support S/MIME. For more information, see S/MIME for message signing and encryption and the Get-SmimeConfig cmdlet reference.

DLP Policy Tips available in the desktop and mobile version of Outlook Web App

Data loss prevention (DLP) Policy Tips are informative notices that are displayed to senders in Outlook when they try sending sensitive information. In Exchange 2013 SP1, this functionality has been extended to both the desktop version of Outlook Web App and the mobile version (named OWA for Devices). You'll see it in action if you have an existing DLP policy with Policy Tips turned on for Outlook. If your policy already includes Policy Tips for Outlook, you don't need to set up anything else. Go ahead and try it out!

Not currently using Policy Tips? To get started, Create a DLP policy from a template, then add a policy tip by editing the policy and adding a **Notify the sender with a Policy Tip** action.

DLP Classification based on Document Fingerprints

Deep content analysis is a cornerstone of DLP in Exchange. Document Fingerprinting expands this capability to enable you to identify standard forms used in your organization, which may contain sensitive information. For example, you can create a fingerprint based off a blank employee information form, and then detect all employee information forms with sensitive content filled in.

DLP sensitive information types for new regions

Exchange 2013 SP1 provides an expanded set of standard DLP sensitive information types covering an increased set of regions, which makes it easier to start using the DLP features. Exchange 2013 SP1 adds region support for Poland, Finland and Taiwan. To learn more about the new DLP sensitive information types, see Sensitive information types inventory.

Using AD FS claims-based authentication with Outlook Web App and ECP

Deploying and configuring Active Directory Federation Services (AD FS) using claims means multifactor authentication can be used with Exchange 2013 SP1 including supporting smartcard and certificate-based authentication in Outlook Web App. In a nutshell, to implement AD FS to support multifactor authentication:

- Install and configure Windows Server 2012 R2 AD FS (this is the most current version of AD FS and contains additional support for multifactor authentication). To learn more about setting up AD FS, see Active Directory Federation Services (AD FS) Overview
- Create relying party trusts and the required AD FS claims.
- Publish Outlook Web App through Web Application Proxy (WAP) on Windows Server 2012 R2.
- Configure Exchange 2013 to use AD FS authentication.
- Configure the Outlook Web App virtual directory to use only AD FS authentication. All other methods of authentication should be disabled.
- Restart Internet Information Services on each Client Access server to load the configuration.

For details, see Using AD FS claims-based authentication with Outlook Web App and EAC

SSL Offloading support

SSL offloading is supported for all of the protocols and related services on Exchange 2013 Client Access servers. By enabling SSL offloading, you terminate the incoming SSL connections on a hardware load balancer instead of on the Client Access servers. Using SSL offloading moves the SSL workloads that are CPU and memory intensive from the Client Access server to a hardware load balancer.

SSL offloading is supported with following protocols and services:

- Outlook Web App
- Exchange Admin Center (EAC)
- Outlook Anywhere
- Offline Address Book (OAB)
- Exchange ActiveSync (EAS)
- Exchange Web Services (EWS)
- Autodiscover
- Mailbox Replication Proxy Service (MRSProxy)
- MAPI virtual directory for Outlook clients

If you have multiple Client Access servers, each Client Access server in your organization must be configured identically. You need to perform the required steps for each protocol or service on every Client Access server in your on-premises organization. For details, see Configuring SSL offloading in Exchange 2013

Public Attachment Handling in Exchange Online

Although there are both private (internal network) and public (external network) settings to control

attachments using Outlook Web App mailbox policies, admins require more consistent and reliable attachment handling when a user signs in to Outlook Web App from a computer on a public network such as at a coffee shop or library. Go here for details, Public Attachment Handling in Exchange Online.

Browser Support for AppCache

Internet Explorer 10 and Windows Store apps using JavaScript support the Application Cache API (or AppCache), as defined in the HTML5 specification, which allows you to create offline web applications. AppCache enables webpages to cache (or save) resources locally, including images, script libraries, style sheets, and so on. In addition, AppCache allows URLs to be served from cached content using standard Uniform Resource Identifier (URI) notation. The following is a list of the browsers that support AppCache:

- Internet Explorer 10 or later versions
- Google Chrome 24 or later versions
- Firefox 23 or later versions
- Safari 6 or later (only on OS X/iOS) versions

Exchange OAuth authentication protocol

Information workers in Exchange on-premises organizations need to collaborate with information workers in Exchange Online organizations when they are connected via an Exchange hybrid deployment. New in Exchange 2013 SP1, this connection can now be enabled and enhanced by using the new Exchange OAuth authentication protocol. The new Exchange OAuth authentication process will replace the Exchange federation trust configuration process and currently enables the following Exchange features:

- Exchange hybrid deployment features, such as shared free/busy calendar information, MailTips, and Message Tracking.
- Exchange In-place eDiscovery

For more information, see Configure OAuth authentication between Exchange and Exchange Online organizations.

Hybrid deployments with multiple Active Directory forests

New in Exchange 2013 SP1, hybrid deployments are now supported in organizations with multiple Active Directory forests. For hybrid deployment features and considerations, multi-forest organizations are defined as organizations having Exchange servers deployed in multiple Active Directory forests. Organizations that utilize a resource forest for user accounts, but maintain all Exchange servers in a single forest, aren't classified as multi-forest in hybrid deployment scenarios. These types of organizations should consider themselves a single forest organization when planning and configuring a hybrid deployment.

For more information, see **Hybrid deployments with multiple Active Directory forests**.

Database Availability Group without an Administrative Access Point

Windows Server 2012 R2 enables you to create a failover cluster without an administrative access point. Exchange 2013 SP1 introduces the ability to leverage this capability and create a database availability group (DAG) without a cluster administrative access point. Creating a DAG without an administrative access point reduces complexity and simplifies DAG management. In addition, it reduces the attack surface of a DAG by removing the cluster/DAG name from DNS, thereby making it unresolvable over the network.

For more information, see High availability and site resilience.

UM Language Packs

The UM language packs for Exchange 2013 SP1 are available. If you install SP1 on your Mailbox servers, you must install the Exchange 2013 SP1 UM language packs. See Exchange Server 2013 SP1 UM Language Packs to download them. UM language packs are specific to the version of Exchange and the Service Pack (SP) installed.

Planning and deployment

Exchange Server 2013 >

Applies to: *Exchange Server*

Topic Last Modified: *2014-01-28*

The following sections contain links to information about planning for and then deploying Microsoft Exchange Server 2013.

Important:
Make sure you read the Release notes for Exchange 2013 topic before you begin your deployment. The release notes contain important information on issues you might encounter during and after your deployment.

Contents

For more information

Planning for Exchange 2013

Use the following links to access information to help you plan the deployment of Exchange Server 2013 into your organization.

Important:
See Establish a Test Environment later in this topic about installing Exchange 2013 in a test environment.

Mailbox and Client Access servers

Learn about the Mailbox and Client Access server roles that are included with Exchange 2013.

Exchange 2013 system requirements

Understand the system requirements that need to be satisfied in your organization before you can install Exchange 2013.

Exchange 2013 prerequisites

Learn which Windows Server 2008 R2 Service Pack 1 (SP1) or Windows Server 2012 features and the other software that needs to be installed to perform a successful installation of Exchange 2013.

Exchange Server Deployment Assistant

Use this tool to generate a customized checklist for planning, installing, or upgrading to Exchange 2013. Guidance is available for multiple scenarios, including an on-premises, hybrid, or cloud deployment.

Active Directory

Read this topic to learn about how Exchange 2013 uses Active Directory and how your Active Directory deployment affects your Exchange deployment.

Anti-malware protection

Read this topic to understand anti-malware protection options for Exchange 2013.

Exchange Server 2013 Hybrid Deployments

Read this topic to help you with planning a hybrid deployment with Microsoft Office 365 and your on-premises Exchange 2013 organization.

Planning for high availability and site resilience

Read this topic to help you with planning to achieve your high availability and business continuity requirements.

Integration with SharePoint and Lync

Read this topic to learn about integrating Exchange 2013, Microsoft SharePoint 2013, and Microsoft Lync 2013 to enable cross-product archiving, hold, and eDiscovery; site mailboxes; authentication; Lync presence; and more.

Planning for Unified Messaging

Read this topic to learn more about planning for Exchange 2013 Unified Messaging.

Exchange 2013 storage configuration options

Read this topic to learn about the storage architectures, disk types, and storage configurations supported by the Exchange 2013 Mailbox server role.

Exchange 2013 virtualization

Read this topic to learn more about how you can deploy Exchange 2013 in a virtualized environment.

Multi-tenancy in Exchange 2013

Read this topic to learn more about how you can configure Exchange 2013 to host multiple and discrete organizations or business units that ordinarily don't share email, data, users, global address lists (GALs), or other commonly used Exchange objects.

Exchange Development Technologies

This topic contains important information about Application Programming Interfaces (APIs) that are available for applications that use Exchange 2013.

Establish a test environment

Before installing Exchange 2013 for the first time, we recommend that you install it in an isolated test environment. This approach reduces the risk of end-user downtime and negative ramifications to the production environment.

The test environment will act as your "proof of concept" for your new Exchange 2013 design and make it possible to move forward or roll back any implementations before deploying into your production environments. Having an exclusive test environment for validation and testing allows you to do pre-installation checks for your future production environments. By installing in a test environment first, we believe that your organization will have a better likelihood of success in a full production implementation.

For many organizations, the costs of building a test lab may be high because of the need to duplicate the production environment. To reduce the hardware costs associated with a prototype lab, we recommend the use of virtualization by using Windows Server 2008 R2 or Windows Server 2012 Hyper-V technologies. Hyper-V enables server virtualization, allowing multiple virtual operating systems to run on a single physical machine.

For more detailed information about Hyper-V, see Server Virtualization. For information about Microsoft support of Exchange 2013 in production on hardware virtualization software, see "Hardware virtualization" in Exchange 2013 system requirements.

Deploying Exchange 2013

The deployment phase is the period during which you install Exchange 2013 into your organization. Before you begin the deployment phase, you should plan your Exchange organization. For more information, see the Planning section earlier in this topic.

Use the following links to access the information you need to help you with deploying Exchange 2013.

Prepare Active Directory and domains

Learn about the steps you need to take to prepare your Active Directory forest for Exchange 2013 and the changes Exchange makes to your forest.

Install Exchange 2013 using the Setup wizard

Follow the steps in this topic to use the Exchange 2013 Setup wizard to install Exchange 2013 into a new Exchange organization.

Install Exchange 2013 using unattended mode

Follow the steps in this topic to use Exchange 2013 unattended setup to install Exchange 2013 into a new Exchange organization.

Install the Exchange 2013 Edge Transport role using the Setup wizard

Follow the steps in this topic to use the Exchange 2013 Setup wizard to install the Exchange 2013 Edge Transport role into a perimeter network.

Upgrade Exchange 2013 to the latest cumulative update or service pack

Follow the steps in this topic to apply the latest cumulative update or service pack to Exchange 2013 servers in your organization.

Upgrade from Exchange 2010 to Exchange 2013

Follow the steps in this topic to install Exchange 2013 into an existing Exchange 2010 organization.

Upgrade from Exchange 2007 to Exchange 2013

Follow the steps in this topic to install Exchange 2013 into an existing Exchange 2007 organization.

Deploy multiple forest topologies for Exchange 2013

Read this topic for information that will help you deploy Exchange 2013 in an organization that contains more than one Active Directory forest.

Deploying voice mail and UM

Read this topic for information that will help you understand deploying Exchange 2013 Unified Messaging, whether a new deployment or an upgrade.

Hybrid Deployment procedures

Read this topic for information that will help you deploy Exchange 2013 in an existing hybrid deployment.

Exchange 2013 post-Installation tasks

Learn about post-installation tasks to complete your Exchange 2013 installation.

Understanding Exchange 2013 Setup

You can use different types and modes of Exchange 2013 Setup to install and maintain the various editions and versions of Exchange 2013.

Exchange editions and versions

Exchange 2013 is available in two server editions: Standard Edition and Enterprise Edition. These are licensing editions that are defined by a product key. For more information, see Exchange Server

Licensing.

Types of Exchange Setup

You have the following options for Exchange 2013 Setup:

- **Exchange Setup UI** Running Setup.exe without any command-line switches provides an interactive experience where you are guided by the Exchange 2013 Setup wizard.
- **Exchange Unattended Setup** Running Setup.exe with command-line switches enables you to install Exchange from an interactive command line or through a script.

Setup.exe is available from the Exchange 2013 DVD or the downloaded source files.

Modes of Exchange Setup

Setup for Exchange 2013 includes several installation modes:

- **Install** Use this mode when you're installing a new server role or adding a server role to an existing installation (maintenance mode). You can use this mode from both the Exchange Setup wizard and the unattended install.
- **Uninstall** Use this mode when you're removing the Exchange installation from a computer. You can use this mode from both the Exchange Setup wizard and the unattended install.
- **Upgrade** Select this mode used when you have an existing installation of Exchange and you're installing a cumulative update or service pack. You can use this mode from both the Exchange Setup wizard and the unattended install.

Note:

Exchange 2013 doesn't support in-place upgrades from previous versions of Exchange. This mode is used only to install cumulative updates or service packs.

- **RecoverServer** Use this mode when there has been a catastrophic failure of a server, and you need to recover data. You must install a server using the same fully qualified domain name (FQDN) as the failed server, and then run Setup with the **/m:RecoverServer** switch. Don't specify the roles to restore. Setup detects the Exchange Server object in Active Directory and installs the corresponding files and configuration automatically. After you recover the server, you can restore databases and reconfigure any additional settings. To run in **RecoverServer** mode, you can't have Exchange installed on the server. The Exchange server object must exist in Active Directory. You can only use this mode during an unattended installation.

Note:

You must complete one mode of Setup before you can use another mode.

For more information

IPv6 support in Exchange 2013

Exchange admin center in Exchange 2013

Exchange 2013 language support

Exchange 2013 readiness checks

Exchange Server Deployment Assistant

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013, Exchange Server, Exchange Online*

Topic Last Modified: *2013-08-27*

The Exchange Server Deployment Assistant is a Web-based tool that can help you with your Microsoft Exchange Server 2013 deployment. The Deployment Assistant asks you a few questions about your current environment and then generates a custom checklist and procedures that help simplify your deployment. To access the Deployment Assistant, see Exchange Server Deployment Assistant.

You can use the Deployment Assistant for the following deployment scenarios:

- **On-Premises only**
  - New installation of Exchange Server 2013
  - Upgrade from Exchange Server 2010 to Exchange 2013
  - Upgrade from Exchange Server 2007 to Exchange 2013
  - Upgrade from a mixed Exchange 2007 and Exchange 2010 organization to Exchange 2013

For more information about this scenario, see Planning and deployment.

- **Hybrid (On-Premises + Exchange Online)**
  - Exchange 2013 on-premises with Exchange Online
  - Exchange 2010 on-premises with Exchange Online
  - Exchange 2007 on-premises with Exchange Online

For more information about this scenario, see **Exchange Server 2013 Hybrid Deployments**.

Important:
If you have an Exchange 2003 on-premises organization and want to configure a new hybrid deployment with Office 365, you must add one or more servers running Exchange 2010 Server Service Pack 3, not Exchange 2013 servers, to your on-premises organization. To do that, we strongly recommend that you use the Exchange 2010 hybrid deployment option in the Exchange Server Deployment Assistant.

- **Cloud only**

For more information about this scenario, see **Understanding Cloud-Only Deployments**.

In addition to the above Exchange 2013 deployment scenarios, the Deployment Assistant also has deployment scenarios for Exchange 2010.

Active Directory

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-11*

Microsoft Exchange Server 2013 uses Active Directory to store and share directory information with Windows. Active Directory forest design for Exchange 2013 is similar to Exchange Server 2010, except in a few ways, which are discussed below.

Active Directory driver

The Active Directory driver is the core Microsoft Exchange component that allows Exchange services to create, modify, delete, and query for Active Directory Domain Services (AD DS) data. In Exchange 2013, all access to Active Directory is done using the Active Directory driver itself. Previously, in Exchange 2010, DSAccess provided directory lookup services for components such as SMTP, message transfer agent (MTA), and the Exchange store.

The Active Directory driver also uses Microsoft Exchange Active Directory Topology (MSExchangeADTopology), which allows the Active Directory driver to use Directory Service Access (DSAccess) topology data. This data includes the list of available domain controllers and global catalog servers available to handle Exchange requests. For more information about the Active Directory Driver, see Active Directory Domain Services.

Active Directory schema changes

Exchange 2013 adds new attributes to the Active Directory domain service schema and also makes other modifications to existing classes and attributes. For more information about Active Directory changes when you install Exchange 2013, see Exchange 2013 Active Directory schema changes.

For more information

To learn more about how Exchange 2013 stores and retrieves information in Active Directory so that you can plan access to it, see Access to Active Directory.

For more information about Active Directory forest design, see AD DS Design Guide.

To learn more about computers running Windows in an Active Directory domain and deploying Exchange 2013 in a domain that has a disjoint namespace, see Disjoint namespace scenarios.

Access to Active Directory

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-05-17*

Microsoft Exchange Server 2013 stores all configuration and recipient information in the Active Directory directory service database. When a computer running Exchange 2013 requires information about recipients and information about the configuration of the Exchange organization, it must query Active Directory to access the information. Active Directory servers must be available for Exchange 2013 to function correctly.

This topic explains how Exchange 2013 stores and retrieves information in Active Directory so that you can plan access to Active Directory. This topic also discusses issues you should be aware of if you try to recover deleted Exchange 2013 Active Directory objects.

Exchange information stored in Active Directory

The Active Directory database stores information in three types of logical partitions that are described in the following sections:

- Schema partition
- Configuration partition
- Domain partition

Schema partition

The schema partition stores the following two types of information:

- **Schema classes** define all the types of objects that can be created and stored in Active Directory.
- **Schema attributes** define all the properties that can be used to describe the objects that are stored in Active Directory.

When you install the first Exchange 2013 server role in the forest or run the Active Directory preparation process, the Active Directory preparation process adds many classes and attributes to the Active Directory schema. The classes that are added to the schema are used to create Exchange-specific objects, such as agents and connectors. The attributes that are added to the schema are used to configure the Exchange-specific objects and the mail-enabled users and groups. These attributes include properties, such as Microsoft Office Outlook Web Access settings and Microsoft Exchange Unified Messaging (UM) settings. Every domain controller and global catalog server in the forest contains a complete replica of the schema partition.

For more information about schema modifications in Exchange 2013, see Exchange 2013 Active Directory schema changes.

Configuration partition

The configuration partition stores information about the forest-wide configuration. This configuration information includes the configuration of Active Directory sites, Exchange global settings, transport settings, and mailbox policies. Each type of configuration information is stored in a container in the configuration partition. Exchange configuration information is stored in a subfolder under the configuration partition's Services container. The information that is stored in this container includes the following:

- Address lists
- Address book mailbox policies
- Administrative groups
- Client access settings
- Connections
- Mobile Mailbox Settings
- Global settings
- Monitoring Settings
- System policies
- Retention policies container
- Transport settings

Every domain controller and global catalog server in the forest contains a complete replica of the configuration partition.

Domain partition

The domain partition stores information in default containers and in organizational units that are created by the Active Directory administrator. These containers hold the domain-specific objects. This data includes Exchange system objects and information about the computers, users, and groups in that domain. When Exchange 2013 is installed, Exchange updates the objects in this partition to support Exchange functionality. This functionality affects how recipient information is stored and accessed.

Each domain controller contains a complete replica of the domain partition for the domain for which it is authoritative. Every global catalog server in the forest contains a subset of the information in every domain partition in the forest.

How Exchange 2013 accesses information in Active Directory

Exchange 2013 uses an Active Directory API to access information that is stored in Active Directory. The Microsoft Exchange Active Directory Topology service runs on all Exchange 2013 server roles.

This service reads information from all Active Directory partitions. The data that is retrieved is cached and is used by Exchange 2013 servers to discover the Active Directory site location of all Exchange services in the organization.

For more information about topology and service discovery, see Planning to use Active Directory sites for routing mail.

Exchange 2013 is an Active Directory site-aware application that prefers to communicate with the directory servers that are located in the same site as the Exchange server to optimize network traffic. Each Exchange 2013 organizational server role must communicate with Active Directory to retrieve information about recipients and information about the other Exchange 2013 server roles. The data that each server role obtains is described in the following sections.

By default, whenever an Exchange 2013 server starts, it binds to a randomly selected domain controller and global catalog server in its own site. You can view the selected directory servers by using the **Get-ExchangeServer** cmdlet in the Exchange Management Shell. You can also use the **Set-ExchangeServer** cmdlet to configure a static list of domain controllers to which an Exchange 2013 server should bind or a list of domain controllers that should be excluded.

Important:
A Windows Server 2008 domain controller can be configured as a read-only directory server. This configuration is useful when you want to deploy a domain controller or global catalog server in a remote site for authentication and authorization purposes, but you don't want to allow administrators in that site to write changes to Active Directory. However, you can't deploy an Exchange 2013 server in any site that contains only read-only directory servers.

Mailbox server role

The Mailbox server role stores configuration information about mailbox users and stores in Active Directory. Additionally, for Exchange 2013, the Mailbox server includes all the traditional server components found in Exchange 2010: the Client Access protocols, Transport service, Mailbox databases, and Unified Messaging. The Mailbox server handles all activity for the active mailboxes on that server.

Client access server role

In Exchange 2013, the Client Access server provides authentication, limited redirection, and proxy services. The Client Access server itself doesn't do any data rendering. The Client Access server is a thin and stateless server. There is never anything queued or stored on the Client Access server. The Client Access server offers all the usual client access protocols: HTTP, POP and IMAP, and SMTP.

Recovery of deleted Exchange objects

Active Directory Recycle Bin helps minimize directory service downtime by enhancing your ability to preserve and recover accidentally deleted Active Directory objects without restoring Active

Directory data from backups, restarting Active Directory Domain Services (AD DS), or rebooting domain controllers.

The most important thing to understand about recovering deleted Exchange-related Active Directory objects is that Exchange objects don't exist in isolation. For example, when you mail-enable a user, several different policies and links are calculated for the user based on your current Exchange configuration. Two problems that may arise when you restore a deleted Exchange configuration or recipient object are:

- **Collisions** Some Exchange attributes must be unique across a forest. For example, proxy (email) addresses must not be the same for two different users. Active Directory doesn't enforce proxy address uniqueness—Exchange administrative tools check for uniqueness. Exchange email address policies also automatically resolve possible conflicts in proxy address assignment based on deterministic rules. Therefore, it's possible to restore an Exchange user object and, as a result, create a collision with proxy addresses or other attributes that should be unique.
- **Misconfigurations** Exchange has automated rules that assign various policies or settings. If you delete a recipient, and then change the rules or policies, restoring an Exchange user object may result in a user being assigned to the wrong policy (or even to a policy that no longer exists).

The following guidelines will help you minimize problems or issues when you recover deleted Exchange-related objects:

- If you deleted an Exchange configuration object using Exchange management tools, don't restore the object. Instead, create the object again using the Exchange management tools (Exchange admin center or Exchange Management Shell).
- If you deleted an Exchange configuration object without using the Exchange management tools, recover the object as soon as possible. The more administrative and configuration changes that have been made in the system since the deletion, the more likely it is that restoring the objects will result in misconfiguration.
- If you recover deleted Exchange recipients (contacts, users, or distribution groups), monitor closely for collisions and errors relating to the recovered objects. If Exchange policies or other configuration relating to recipients may have been modified since the deletion, re-apply current policies to the restored recipients to ensure that they're configured correctly.

For more information

Active Directory Recycle Bin Step-by-Step Guide

Introduction to Active Directory Administrative Center Enhancements (Level 100)

Advanced AD DS Management Using Active Directory Administrative Center (Level 200)

Exchange 2013 Active Directory schema

changes

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-06*

Microsoft Exchange Server 2013 adds new and modifies existing Active Directory schema classes and attributes. This reference topic provides a summary of the Active Directory schema changes that are made when you install the release to manufacturing (RTM) version of Exchange 2013 or any of its cumulative updates or service packs. Refer to the .ldf files for more information about changes to the Active Directory schema. The .ldf files are located in the \amd64\Setup\Data\ directory in the Exchange installation files.

Note:
The Active Directory schema changes identified in this topic may not apply to all editions of an Exchange Server version. To verify that Active Directory has been successfully prepared, see the "How do you know this worked?" section in Prepare Active Directory and domains.

This document includes the following sections:

- Exchange 2013 CU6 Active Directory schema changes
- Exchange 2013 CU5 Active Directory schema changes
- Exchange 2013 SP1 Active Directory schema changes
- Exchange 2013 CU3 Active Directory schema changes
- Exchange 2013 CU2 Active Directory schema changes
- Exchange 2013 CU1 Active Directory schema changes
- Exchange 2013 RTM Active Directory schema changes

Exchange 2013 CU6 Active Directory schema changes

This section summarizes the changes that are made to the Active Directory schema when you install Exchange 2013 CU6. This section includes the following subsections:

- Classes modified by Exchange 2013 CU6
- Attributes added by Exchange 2013 CU6
- Attributes modified by Exchange 2013 CU6

Classes modified by Exchange 2013 CU6

This section contains the attributes added in Exchange 2013 CU6.

Class	Change	Attribute/Class
Mail-Recipient	add: mayContain	msExchAuxMailboxParentObjec

		tldLink
Mail-Recipient	add: mayContain	msExchAuxMailboxParentObjec tldBL

Attributes added by Exchange 2013 CU6

This section contains the attributes added in Exchange 2013 CU6.

- ms-Exch-Aux-Mailbox-Parent-Object-Id-Link
- ms-Exch-Aux-Mailbox-Parent-Object-Id-BL

Attributes modified by Exchange 2013 CU6

This section contains the attributes modified in Exchange 2013 CU6.

Attribute	Change	Value
ms-Exch-Smtp-TLS-Certificate	replace: rangeUpper	1024

Exchange 2013 CU5 Active Directory schema changes

This section summarizes the changes that are made to the Active Directory schema when you install Exchange 2013 CU5. This section includes the following subsections:

- Classes added by Exchange 2013 CU5
- Attributes added by Exchange 2013 CU5
- Attributes modified by Exchange 2013 CU5

Classes added by Exchange 2013 CU5

This section contains the classes added in Exchange 2013 CU5.

Class	Change
ms-Exch-Unified-Policy	ntdsSchemaAdd
ms-Exch-Unified-Rule	ntdsSchemaAdd

Attributes added by Exchange 2013 CU5

This section contains the attributes added in Exchange 2013 CU5.

- ms-Exch-UG-Member-Link
- ms-Exch-UG-Member-BL

Attributes modified by Exchange 2013 CU5

This section contains the attributes modified in Exchange 2013 CU5.

Attribute	Change	Value
ms-Exch-Smtp-Receive-Tls-Certificate-Name	Replace: rangeUpper	1024
Mail-Recipient	Replace: mayContain	msExchUGMemberLink
Top	Replace: mayContain	msExchUGMemberBL

Exchange 2013 SP1 Active Directory schema changes

This section summarizes the changes that are made to the Active Directory schema when you install Exchange 2013 Service Pack 1 (SP1). This section includes the following subsections:

- Classes added by Exchange 2013 SP1
- Classes modified by Exchange 2013 SP1
- Attributes added by Exchange 2013 SP1
- Global catalog attributes added by Exchange 2013 SP1
- Attributes modified by Exchange 2013 SP1

Classes added by Exchange 2013 SP1

This section contains the classes added in Exchange 2013 SP1.

Class	Change
ms-Exch-Intra-Organization-Connector	ntdsSchemaModify
ms-Exch-Client-Access-Rule	ntdsSchemaModify

Classes modified by Exchange 2013 SP1

This section contains the attributes added in Exchange 2013 SP1.

Class	Change	Attribute/Class
ms-Exch-Mail-Storage	add: mayContain	msExchMailboxContainerGuid
ms-Exch-Mail-Storage	add: mayContain	msExchUnifiedMailbox
ms-Exch-OAB	add: mayContain	msExchOABGeneratingMailbox

		Link
Top	add: mayContain	msExchOABGeneratingMailboxBL

Attributes added by Exchange 2013 SP1

This section contains the attributes added in Exchange 2013 SP1.

- ms-Exch-OAB-Generating-Mailbox-Link
- ms-Exch-OAB-Generating-Mailbox-BL

Global catalog attributes added by Exchange 2013 SP1

The following global catalog attributes are added by Exchange 2013 SP1:

- ms-Exch-Mailbox-Container-Guid
- ms-Exch-Unified-Mailbox

Attributes modified by Exchange 2013 SP1

This section contains the attributes modified in Exchange 2013 SP1.

Attribute	Change	Value
Ms-exch-schema-version-pt	rangeUpper	15292

Exchange 2013 CU3 Active Directory schema changes

This section summarizes the changes that are made to the Active Directory schema when you install Exchange 2013 CU3. This section includes the following subsections:

- Classes added by Exchange 2013 CU3
- Attributes added by Exchange 2013 CU3
- Attributes modified by Exchange 2013 CU3

Classes added by Exchange 2013 CU3

This section contains the classes added in Exchange 2013 CU3.

Class	Change
msExchThrottlingPolicy	ntdsSchemaModify

Attributes added by Exchange 2013 CU3

This section contains the attributes added in Exchange 2013 CU3.

- ms-Exch-Encryption-Throttling-Policy-State-Ex

Attributes modified by Exchange 2013 CU3

This section contains the attributes modified in Exchange 2013 CU3.

Attribute	Change	Value
ms-Exch-Coexistence-Secure-Mail-Certificate-Thumbprintms-Exch-Sync-Cookie	rangeUpper	1024

Exchange 2013 CU2 Active Directory schema changes

This section summarizes the changes that are made to the Active Directory schema when you install Exchange 2013 CU2. This section includes the following subsections:

- Classes added by Exchange 2013 CU2
- Classes modified by Exchange 2013 CU2
- Attributes added by Exchange 2013 CU2
- Attributes modified by Exchange 2013 CU2

Classes added by Exchange 2013 CU2

This section contains the classes added in Exchange 2013 CU2.

Class	Change
ms-Exch-Config-Settings	ntdsSchemaAdd
ms-Exch-Encryption-Virtual-Directory	ntdsSchemaAdd

Classes modified by Exchange 2013 CU2

This section contains the classes modified in Exchange 2013 CU2.

Class	Change	Attribute/Class
ms-Exch-Base-Class	Add:mayContain	msExchTenantCountry
ms-Exch-Base-Class	Add:mayContain	msExchConfigurationXML
ms-Exch-Account-Forest	Add:mayContain	msExchPartnerId

Attributes added by Exchange 2013 CU2

This section contains the attributes added in Exchange 2013 CU2.

- ms-Exch-Tenant-Country

Attributes modified by Exchange 2013 CU2

This section contains the attributes modified in Exchange 2013 CU2.

Attribute	Change	Value
ms-Exch-Sync-Cookie	rangeUpper	262144
ms-Exch-Schema-Version-Pt	rangeUpper	15281

Object IDs added by Exchange 2013 CU2

The following class object IDs are added when you install Exchange 2013 CU1:

- 1.2.840.113556.1.5.7000.62.50204
- 1.2.840.113556.1.5.7000.62.50205

The following attribute object IDs are added when you install Exchange 2013 CU1:

- 1.2.840.113556.1.4.7000.102.52130

Exchange 2013 CU1 Active Directory schema changes

This section summarizes the changes that are made to the Active Directory schema when you install Exchange 2013 CU1. This section includes the following subsections:

- Classes modified by Exchange 2013 CU1
- Classes added by Exchange 2013 CU1
- Attributes modified by Exchange 2013 CU1
- Object IDs added by Exchange 2013 CU1
- Indexed attributes added by Exchange 2013 CU1
- Property sets modified by Exchange 2013 CU1
- Global catalog attributes added by Exchange 2013 CU1

Classes modified by Exchange 2013 CU1

This section contains the classes modified in Exchange 2013 CU1.

Class	Change	Attribute/Class
Exch-Configuration-Unit-Container	add:mayContain	msExchArchiveRelease

Exch-Configuration-Unit-Container	add:mayContain	msExchMailboxRelease
Exch-Exchange-Server	add:mayContain	msExchArchiveRelease
Exch-Exchange-Server	add:mayContain	msExchMailboxRelease
Exch-OAB	add:mayContain	msExchLastUpdateTime
Exch-Accepted-Domain	add:mayContain	msExchOfflineOrgIdHomeRealmRecord
Exch-Base-Class	add:mayContain	msExchCapabilityIdentifiers
Exch-Base-Class	add:mayContain	msExchObjectID
Exch-Base-Class	add:mayContain	msExchProvisioningTags
Exch-Organization-Container	add:mayContain	msExchMaxABP
Exch-Organization-Container	add:mayContain	msExchMaxOAB
Exch-Organization-Container	add:mayContain	pFContacts
Exch-Team-Mailbox-Provisioning-Policy	add:mayContain	msExchConfigurationXML
Exch-MDB-Availability-Group	add:mayContain	msExchEvictedMembersLink
Top	add:mayContain	msExchEvictedMemebersBL
Exch-On-Premises-Organization	add:mayContain	msExchTrustedDomainLink
Exch-OWA-Mailbox-Policy	add:mayContain	msExchConfigurationXML
Exch-OWA-Virtual-Directory	add:mayContain	msExchConfigurationXML

Classes added by Exchange 2013 CU1

This section contains the classes modified in Exchange 2013 CU1.

Class	Change
Exch-Mapi-Virtual-Directory	ntdsSchemaAdd
Exch-Push-Notifications-App	ntdsSchemaAdd

Attributes modified by Exchange 2013 CU1

This section contains the attributes modified in Exchange 2013 CU1.

Class	Change	Attribute/Class
Exch-Mailflow-Policy-Transport-Rules-Template-Xml	rangeUpper	256000
Exch-Configuration-Unit-Container	rangeUpper	15254

Object IDs added by Exchange 2013 CU1

The following attribute object IDs are added when you install Exchange 2013 CU1:

- 1.2.840.113556.1.4.7000.102.52109
- 1.2.840.113556.1.4.7000.102.52110
- 1.2.840.113556.1.4.7000.102.52127
- 1.2.840.113556.1.4.7000.102.52126
- 1.2.840.113556.1.4.7000.102.52128
- 1.2.840.113556.1.4.7000.102.52129

The following class object IDs are added when you install Exchange 2013 CU1:

- 1.2.840.113556.1.5.7000.62.50202
- 1.2.840.113556.1.5.7000.62.50203

Indexed attributes added by Exchange 2013 CU1

The following table lists the attributes that are added to the list of indexed attributes when you install Exchange 2013 CU1.

Attribute	Search flag value
ms-Exch-Provisioning-Tags	1

Property sets modified by Exchange 2013 CU1

The following property sets are modified when you install Exchange 2013 CU1:

- Exchange-Information

Global catalog attributes added by Exchange 2013 CU1

The following global catalog attributes are added by Exchange 2013 CU1:

- ms-Exch-Offline-OrgId-Home-Realm-Record
- ms-Exch-EvictedMembers-Link
- ms-Exch-EvictedMemebers-BL

Exchange 2013 RTM Active Directory schema changes

This section summarizes the changes that are made to the Active Directory schema when you install Exchange 2013 RTM. This section includes the following subsections:

- Classes modified by Exchange 2013 RTM
- Attributes modified by Exchange 2013 RTM
- Classes added by Exchange 2013 RTM
- Attributes added by Exchange 2013 RTM
- MAPI IDs added by Exchange 2013 RTM
- Extended rights added by Exchange 2013 RTM
- Object IDs added by Exchange 2013 RTM
- Indexed attributes added by Exchange 2013 RTM
- Global catalog attributes added by Exchange 2013 RTM

Classes modified by Exchange 2013 RTM

This section contains the classes modified in Exchange 2013 RTM.

Class	Change	Attribute/Class
Mail-Recipient	add:mayContain	msExchLocalizationFlags
Mail-Recipient	add:mayContain	msExchRoleGroupType
ms-Exch-Base-Class	add:mayContain	msExchDirsyncAuthorityMetadata
ms-Exch-Base-Class	add:mayContain	msExchDirsyncStatusAck
ms-Exch-Base-Class	add:mayContain	msExchEdgeSyncConfigFlags
ms-Exch-Base-Class	add:mayContain	msExchHABRootDepartmentLink
ms-Exch-Base-Class	add:mayContain	msExchDefaultPublicFolderMail

		box
ms-Exch-Base-Class	add:mayContain	msExchForestModeFlag
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchDirsyncStatus
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchIsDirsyncStatusPending
ms-Exch-Mail-Storage	add:mayContain	msExchPreviousArchiveDatabase
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchDirSyncServiceInstance
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchOrganizationUpgradePolicyLink
ms-Exch-OWA-Mailbox-Policy	add:mayContain	msExchOWASetPhotoURL
ms-Exch-OWA-Virtual-Directory	add:mayContain	msExchOWASetPhotoURL
ms-Exch-Exchange-Server	add:mayContain	msExchWorkloadManagementPolicyLink
Mail-Recipient	add:mayContain	ms-DS-GeoCoordinates-Altitude
Mail-Recipient	add:mayContain	ms-DS-GeoCoordinates-Latitude
Mail-Recipient	add:mayContain	ms-DS-GeoCoordinates-Longitude
ms-Exch-Resource-Policy	add:mayContain	msExchCustomerExpectationCritical

ms-Exch-Resource-Policy	add:mayContain	msExchDiscretionaryCritical
ms-Exch-Resource-Policy	add:mayContain	msExchInternalMaintenanceCritical
ms-Exch-Resource-Policy	add:mayContain	msExchUrgentCritical
ms-Exch-Availability-Address-Space	add:mayContain	msExchFedTargetAutodiscoverEPR
ms-Exch-Private-MDB	add:mayContain	msExchMailboxDatabaseTransportFlags
Mail-Recipient	add:mayContain	msExchRecipientSoftDeletedStatus
Mail-Recipient	add:mayContain	msExchWhenSoftDeletedTime
ms-Exch-Custom-Attributes	add:mayContain	msExchExtensionCustomAttribute1
ms-Exch-Custom-Attributes	add:mayContain	msExchExtensionCustomAttribute2
ms-Exch-Custom-Attributes	add:mayContain	msExchExtensionCustomAttribute3
ms-Exch-Custom-Attributes	add:mayContain	msExchExtensionCustomAttribute4
ms-Exch-Custom-Attributes	add:mayContain	msExchExtensionCustomAttribute5
ms-Exch-Virtual-Directory	add:mayContain	msExchMRSProxyFlags
ms-Exch-Virtual-Directory	add:mayContain	msExchMRSProxyMaxConnections
ms-Exch-Active-Sync-Device	add:mayContain	msExchDeviceClientType

ms-Exch-Exchange-Server	add:mayContain	msExchMalwareFilteringDeferAttempts
ms-Exch-Exchange-Server	add:mayContain	msExchMalwareFilteringDeferWaitTime
ms-Exch-Exchange-Server	add:mayContain	msExchMalwareFilteringFlags
ms-Exch-Exchange-Server	add:mayContain	msExchMalwareFilteringPrimaryUpdatePath
ms-Exch-Exchange-Server	add:mayContain	msExchMalwareFilteringSecondaryUpdatePath
ms-Exch-Exchange-Server	add:mayContain	msExchMalwareFilteringUpdateFrequency
ms-Exch-Exchange-Server	add:mayContain	msExchMalwareFilteringUpdateTimeout
ms-Exch-Mail-Storage	add:mayContain	msExchTeamMailboxExpiration
ms-Exch-Mail-Storage	add:mayContain	msExchTeamMailboxOwners
ms-Exch-Mail-Storage	add:mayContain	msExchTeamMailboxSharePointLinkedBy
ms-Exch-Mail-Storage	add:mayContain	msExchTeamMailboxSharePointUrl
ms-Exch-Mail-Storage	add:mayContain	msExchTeamMailboxShowInClientList
ms-Exch-Mail-Storage	add:mayContain	msExchCalendarLoggingQuota
ms-Exch-MSO-Sync-Service-Instance	add:mayContain	msExchMSOForwardSyncNonRecipientCookie
ms-Exch-MSO-Sync-Service-Instance	add:mayContain	msExchMSOForwardSyncRecipientCookie

ms-Exch-MSO-Sync-Service-Instance	add:mayContain	msExchMSOForwardSyncReplayList
ms-Exch-MSO-Sync-Service-Instance	add:mayContain	msExchAccountForestLink
Top	add:mayContain	msExchAccountForestBL
Top	add:mayContain	msExchTrustedDomainBL
Top	add:mayContain	msExchAcceptedDomainBL
Top	add:mayContain	msExchHygieneConfigurationMalwareBL
Top	add:mayContain	msExchHygieneConfigurationSpamBL
ms-Exch-Base-Class	add:mayContain	msExchELCMailboxFlags
Mail-Recipient	add:mayContain	msExchHomeMTASL
Mail-Recipient	add:mayContain	msExchMailboxMoveSourceArchiveMDBLinkSL
Mail-Recipient	add:mayContain	msExchMailboxMoveSourceMDBLinkSL
Mail-Recipient	add:mayContain	msExchMailboxMoveTargetArchiveMDBLinkSL
Mail-Recipient	add:mayContain	msExchMailboxMoveTargetMDBLinkSL
ms-Exch-Accepted-Domain	add:mayContain	msExchHygieneConfigurationLink
ms-Exch-Accepted-Domain	add:mayContain	msExchTransportResellerSettingsLinkSL

ms-Exch-Configuration-Unit-Container	add:mayContain	msExchManagementSiteLinkSL
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchOrganizationUpgradePolicyLinkSL
ms-Exch-Fed-OrgId	add:mayContain	msExchFedDelegationTrustSL
ms-Exch-Mail-Gateway	add:mayContain	msExchHomeMDBSL
ms-Exch-Mail-Gateway	add:mayContain	msExchHomeMTASL
ms-Exch-Mail-Storage	add:mayContain	msExchArchiveDatabaseLinkSL
ms-Exch-Mail-Storage	add:mayContain	msExchDisabledArchiveDatabaseLinkSL
ms-Exch-Mail-Storage	add:mayContain	msExchHomeMDBSL
ms-Exch-Mail-Storage	add:mayContain	msExchMailboxMoveTargetMDBLinkSL
ms-Exch-Mail-Storage	add:mayContain	msExchPreviousArchiveDatabaseSL
ms-Exch-Mail-Storage	add:mayContain	msExchPreviousHomeMDBSL
ms-Exch-MRS-Request	add:mayContain	msExchMailboxMoveSourceMDBLinkSL
ms-Exch-MRS-Request	add:mayContain	msExchMailboxMoveStorageMDBLinkSL
ms-Exch-MRS-Request	add:mayContain	msExchMailboxMoveTargetMDBLinkSL
ms-Exch-OAB	add:mayContain	msExchOffLineABServerSL
ms-Exch-Routing-Group-Connector	add:mayContain	msExchHomeMTASL

ms-Exch-Site-Connector	add:mayContain	msExchHomeMTASL
ms-Exch-Tenant-Perimeter-Settings	add:mayContain	msExchTransportResellerSettingsLinkSL
ms-Exch-Domain-Content-Config	add:mayContain	msExchContentByteEncoderTypeFor7BitCharsets
ms-Exch-Domain-Content-Config	add:mayContain	msExchContentPreferredInternetCodePageForShiftJis
ms-Exch-Domain-Content-Config	add:mayContain	msExchContentRequiredCharSetCoverage
ms-Exch-Coexistence-Relationship	add:mayContain	msExchCoexistenceOnPremisesSmartHost
ms-Exch-Coexistence-Relationship	add:mayContain	msExchCoexistenceSecureMailCertificateThumbprint
ms-Exch-Coexistence-Relationship	add:mayContain	msExchCoexistenceTransportServers
Mail-Recipient	add:mayContain	ms-exch-group-external-member-count
Mail-Recipient	add:mayContain	ms-exch-group-member-count
Ms-Exch-Organization-Container	add:mayContain	ms-exch-organization-flags-2
Mail-Recipient	add:mayContain	msExchGroupExternalMemberCount
Mail-Recipient	add:mayContain	msExchGroupMemberCount
Mail-Recipient	add:mayContain	msExchShadowWhenSoftDeletedTime

ms-Exch-Organization-Container	add:mayContain	msExchOrganizationFlags2
ms-Exch-Organization-Container	add:mayContain	msExchUMAvailableLanguages
ms-Exch-Control-Point-Config	add:mayContain	msExchRMSOnlineCertificationLocationUrl
ms-Exch-Control-Point-Config	add:mayContain	msExchRMSOnlineKeySharingLocationUrl
ms-Exch-Control-Point-Config	add:mayContain	msExchRMSOnlineLicensingLocationUrl
ms-Exch-Throttling-Policy	add:mayContain	msExchThrottlingPolicyFlags
ms-Exch-Exchange-Server	add:mayContain	msExchMalwareFilteringScanTimeout
ms-Exch-Hosted-Content-Filter-Config	add:mayContain	msExchSpamCountryBlockList
ms-Exch-Hosted-Content-Filter-Config	add:mayContain	msExchSpamLanguageBlockList
ms-Exch-Hosted-Content-Filter-Config	add:mayContain	msExchSpamNotifyOutboundRecipients
ms-Exch-Malware-Filter-Config	add:mayContain	msExchMalwareFilterConfigExternalSenderAdminAddress
ms-Exch-Malware-Filter-Config	add:mayContain	msExchMalwareFilterConfigInternalSenderAdminAddress
ms-Exch-MSO-Sync-Service-Instance	add:mayContain	msExchActiveInstanceSleepInterval
ms-Exch-MSO-Sync-Service-	add:mayContain	msExchPassiveInstanceSleepInt

Instance		erval
ms-Exch-MSO-Sync-Service-Instance	add:mayContain	msExchSyncDaemonMaxVersion
ms-Exch-MSO-Sync-Service-Instance	add:mayContain	msExchSyncDaemonMinVersion
Mail-Recipient	add:mayContain	msExchPublicFolderMailbox
Mail-Recipient	add:mayContain	msExchPublicFolderSmtpAddress
ms-Exch-Hosted-Content-Filter-Config	add:mayContain	msExchSpamDigestFrequency
ms-Exch-Hosted-Content-Filter-Config	add:mayContain	msExchSpamQuarantineRetention
ms-Exch-Organization-Container	add:mayContain	msExchWACDiscoveryEndpoint
ms-Exch-Organization-Container	add:mayContain	msExchAdfsAuthenticationRawConfiguration
ms-Exch-Organization-Container	add:mayContain	msExchServiceEndPointURL
ms-Exch-Public-Folder	add:mayContain	msExchPublicFolderEntryId
ms-Exch-Transport-Rule	add:mayContain	msExchTransportRuleImmutableId
ms-Exch-Transport-Settings	add:mayContain	msExchTransportMaxRetriesForLocalSiteShadow
ms-Exch-Transport-Settings	add:mayContain	msExchTransportMaxRetriesForRemoteSiteShadow

ms-Exch-Transport-Settings	add:mayContain	msExchConfigurationXML
ms-Exch-Account-Forest	possSuperiors	msExchContainer
ms-Exch-Base-Class	add:mayContain	msExchCanaryData0
ms-Exch-Base-Class	add:mayContain	msExchCanaryData1
ms-Exch-Base-Class	add:mayContain	msExchCanaryData2
ms-Exch-Base-Class	add:mayContain	msExchCorrelationId
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchRelocateTenantCompletionTargetVector
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchRelocateTenantFlags
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchRelocateTenantSafeLockdownSchedule
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchRelocateTenantSourceForest
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchRelocateTenantStartLockdown
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchRelocateTenantStartRetired
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchRelocateTenantStartSync
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchRelocateTenantStatus
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchRelocateTenantTargetForest
ms-Exch-Configuration-Unit-	add:mayContain	msExchRelocateTenantTransitio

Container		nCounter
ms-Exch-Configuration-Unit-Container	add:mayContain	msExchSyncCookie
ms-Exch-Throttling-Policy	add:mayContain	msExchAnonymousThrottlingPolicyStateEx
ms-Exch-Throttling-Policy	add:mayContain	msExchEASThrottlingPolicyStateEx
ms-Exch-Throttling-Policy	add:mayContain	msExchEWSThrottlingPolicyStateEx
ms-Exch-Throttling-Policy	add:mayContain	msExchGeneralThrottlingPolicyStateEx
ms-Exch-Throttling-Policy	add:mayContain	msExchIMAPThrottlingPolicyStateEx
ms-Exch-Throttling-Policy	add:mayContain	msExchOWAThrottlingPolicyStateEx
ms-Exch-Throttling-Policy	add:mayContain	msExchPOPThrottlingPolicyStateEx
ms-Exch-Throttling-Policy	add:mayContain	msExchPowershellThrottlingPolicyStateEx
ms-Exch-Throttling-Policy	add:mayContain	msExchRCAThrottlingPolicyStateEx
ms-Exch-Mailflow-Policy	add:mayContain	msExchImmutableId
ms-Exch-MDB	add:mayContain	msExchCalendarLoggingQuota
ms-Exch-Transport-Rule	add:mayContain	msExchImmutableId
ms-Exch-MSO-Sync-Service-Instance	add:mayContain	msExchSyncServiceInstanceNewTenantMaxVersion

ms-Exch-MSO-Sync-Service-Instance	add:mayContain	msExchSyncServiceInstanceNewTenantMinVersion

Attributes modified by Exchange 2013 RTM

This section contains the attributes modified in Exchange 2013 RTM.

Class	Change	Value
ms-Exch-Schema-Version-Pt	rangeUpper	15137
ms-Exch-HAB-Root-Department-Link	replace: isMemberOfPartialAttributeSet	TRUE
ms-Exch-Archive-GUID	replace: searchFlags	9
ms-Exch-Accepted-Domain-Name	replace: searchFlags	9
ms-Exch-Bypass-Audit	replace: searchFlags	19
ms-Exch-Mailbox-Audit-Enable	replace: searchFlags	19
ms-Exch-Extension-Custom-Attribute-1	isMemberOfPartialAttributeSet:	TRUE
ms-Exch-Extension-Custom-Attribute-2	isMemberOfPartialAttributeSet:	TRUE
ms-Exch-Extension-Custom-Attribute-3	isMemberOfPartialAttributeSet:	TRUE
ms-Exch-Extension-Custom-Attribute-4	isMemberOfPartialAttributeSet:	TRUE
ms-Exch-Extension-Custom-Attribute-5	isMemberOfPartialAttributeSet	TRUE
ms-Exch-Coexistence-On-Premises-Smart-Host	ntdsSchemaAdd	attributeID: 1.2.840.113556.1.4.7000.102.51992

		isMemberOfPartialAttributeSet: FALSE (not in global catalogue) searchFlags: 0 (no index)
ms-Exch-Coexistence-Secure-Mail-Certificate-Thumbprint	ntdsSchemaAdd	attributeID: 1.2.840.113556.1.4.7000.102.51991 isMemberOfPartialAttributeSet: FALSE (not in global catalogue) searchFlags: 0 (no index)
ms-Exch-Coexistence-Transport-Servers	ntdsSchemaAdd	attributeID: 1.2.840.113556.1.4.7000.102.51990 isMemberOfPartialAttributeSet: FALSE (not in global catalogue) searchFlags: 0 (no index)
ms-Exch-ELC-Mailbox-Flags	replace: attributeSecurityGuid	F6SzsVXskUGzJ7cuM +OK8g==
ms-Exch-Malware-Filtering-Update-Frequency	rangeUpper	38880
ms-Exch-MSO-Forward-Sync-Non-Recipient-Cookie	rangeUpper	20480
ms-Exch-MSO-Forward-Sync-Recipient-Cookie	rangeUpper	20480
ms-Exch-Role-Entries	rangeUpper	8192
ms-Exch-Group-External-Member-Count	ntdsSchemaModify	isMemberOfPartialAttributeSet: TRUE MAPIID:36066
ms-Exch-Group-Member-Count	ntdsSchemaModify	replace: isMemberOfPartialAttributeSeti

		sMemberOfPartialAttributeSet: TRUE MAPIID: 36067

Classes added by Exchange 2013 RTM

The following classes are added when you install Exchange 2013 RTM:

- ms-Exch-ActiveSync-Device-Autoblock-Threshold
- ms-Exch-MSO-Forward-Sync-Divergence
- ms-Exch-MSO-Sync-Service-Instance
- ms-Exch-Organization-Upgrade-Policy
- ms-Exch-Workload-Policy
- ms-Exch-Resource-Policy
- ms-Exch-Team-Mailbox-Provisioning-Policy
- ms-Exch-Account-Forest
- ms-Exch-Hosted-Content-Filter-Config
- ms-Exch-Hygiene-Configuration
- ms-Exch-Malware-Filter-Config
- ms-Exch-Protocol-Cfg-SIP-Container
- ms-Exch-Protocol-Cfg-SIP-FE-Server
- ms-Exch-Exchange-Transport-Server
- ms-Exch-Auth-Auth-Config
- ms-Exch-Auth-Auth-Server
- ms-Exch-Auth-Partner-Application
- ms-Exch-Mailflow-Policy-Collection
- ms-Exch-Mailflow-Policy

Attributes added by Exchange 2013 RTM

The following attributes encompass a set of changes required for new features and are added when you install Exchange 2013 RTM:

- ms-Exch-ActiveSync-Device-AutoBlock-Duration
- ms-Exch-ActiveSync-Device-Autoblock-Threshold-Incidence-Duration
- ms-Exch-ActiveSync-Device-Autoblock-Threshold-Incidence-Limit
- ms-Exch-ActiveSync-Device-Autoblock-Threshold-Type
- ms-Exch-Dirsync-Authority-Metadata
- ms-Exch-Dirsync-Status
- ms-Exch-Dirsync-Status-Ack
- ms-Exch-Edge-Sync-Config-Flags
- ms-Exch-Is-Dirsync-Status-Pending,
- ms-Exch-Localization-Flags
- ms-Exch-Previous-Archive-Database

- ms-Exch-RoleGroup-Type
- ms-Exch-External-Directory-Object-Class
- ms-Exch-MSO-Forward-Sync-Divergence-Count
- ms-Exch-MSO-Forward-Sync-Divergence-Timestamp
- ms-Exch-MSO-Forward-Sync-Divergence-Related-Object-Link
- ms-Exch-Default-Public-Folder-Mailbox
- ms-Exch-Forest-Mode-Flag
- ms-Exch-Dir-Sync-Service-Instance
- ms-Exch-Organization-Upgrade-Policy-Date
- ms-Exch-Organization-Upgrade-Policy-Enabled
- ms-Exch-Organization-Upgrade-Policy-MaxMailboxes
- ms-Exch-Organization-Upgrade-Policy-Priority
- ms-Exch-Organization-Upgrade-Policy-Source-Version
- ms-Exch-Organization-Upgrade-Policy-Status
- ms-Exch-Organization-Upgrade-Policy-Target-Version
- ms-Exch-OWA-Set-Photo-URL
- ms-Exch-Organization-Upgrade-Policy-Link
- ms-Exch-Organization-Upgrade-Policy-BL
- ms-Exch-Workload-Classification
- ms-Exch-Workload-Management-Is-Enabled
- ms-Exch-Workload-Type
- ms-Exch-Workload-Management-Policy-Link
- ms-Exch-Workload-Management-Policy-BL
- ms-Exch-Workload-Management-Policy
- ms-Exch-Customer-Expectation-Overloaded
- ms-Exch-Customer-Expectation-Underloaded
- ms-Exch-Discretionary-Overloaded
- ms-Exch-Discretionary-Underloaded
- ms-Exch-Internal-Maintenance-Overloaded
- ms-Exch-Internal-Maintenance-Underloaded
- ms-Exch-Resource-Type
- ms-Exch-Urgent-Overloaded
- ms-Exch-Urgent-Underloaded
- ms-DS-GeoCoordinates-Altitude
- ms-DS-GeoCoordinates-Latitude
- ms-DS-GeoCoordinates-Longitude
- ms-Exch-Customer-Expectation-Critical
- ms-Exch-Discretionary-Critical
- ms-Exch-Internal-Maintenance-Critical
- ms-Exch-Urgent-Critical
- ms-Exch-Mailbox-Database-Transport-Flags
- ms-Exch-Extension-Custom-Attribute-1

- ms-Exch-Extension-Custom-Attribute-2
- ms-Exch-Extension-Custom-Attribute-3
- ms-Exch-Extension-Custom-Attribute-4
- ms-Exch-Extension-Custom-Attribute-5
- ms-Exch-MRS-Proxy-Flags
- ms-Exch-MRS-Proxy-Max-Connections
- ms-Exch-Recipient-SoftDeleted-Status
- ms-Exch-When-Soft-Deleted-Time
- ms-Exch-Device-Client-Type
- ms-Exch-Malware-Filtering-Defer-Attempts
- ms-Exch-Malware-Filtering-Defer-Wait-Time
- ms-Exch-Malware-Filtering-Flags
- ms-Exch-Malware-Filtering-Primary-Update-Path
- ms-Exch-Malware-Filtering-Secondary-Update-Path
- ms-Exch-Malware-Filtering-Update-Frequency
- ms-Exch-Malware-Filtering-Update-Timeout
- ms-Exch-Team-Mailbox-Expiration
- ms-Exch-Team-Mailbox-Expiry-Days
- ms-Exch-Team-Mailbox-Owners
- ms-Exch-Team-Mailbox-SharePoint-Linked-By
- ms-Exch-Team-Mailbox-SharePoint-Url
- ms-Exch-Team-Mailbox-Show-In-Client-List
- ms-Exch-Account-Forest-Link
- ms-Exch-Account-Forest-BL
- ms-Exch-Trusted-Domain-Link
- ms-Exch-Trusted-Domain-BL
- ms-Exch-Archive-Database-Link-SL
- ms-Exch-Disabled-Archive-Database-Link-SL
- ms-Exch-Fed-Delegation-Trust-SL
- ms-Exch-Home-MDB-SL
- ms-Exch-Home-MTA-SL
- ms-Exch-Mailbox-Move-Source-Archive-MDB-Link-SL
- ms-Exch-Mailbox-Move-Source-MDB-Link-SL
- ms-Exch-Mailbox-Move-Storage-MDB-Link-SL
- ms-Exch-Mailbox-Move-Target-Archive-MDB-Link-SL
- ms-Exch-Mailbox-Move-Target-MDB-Link-SL
- ms-Exch-Malware-Filter-Config-Alert-Text
- ms-Exch-Malware-Filter-Config-External-Body
- ms-Exch-Malware-Filter-Config-External-Subject
- ms-Exch-Malware-Filter-Config-Flags
- ms-Exch-Malware-Filter-Config-From-Address
- ms-Exch-Malware-Filter-Config-From-Name

- ms-Exch-Malware-Filter-Config-Internal-Body
- ms-Exch-Malware-Filter-Config-Internal-Subject
- ms-Exch-Management-Site-Link-SL
- ms-Exch-Off-Line-AB-Server-SL
- ms-Exch-Organization-Upgrade-Policy-Link-SL
- ms-Exch-Previous-Archive-Database-SL
- ms-Exch-Previous-Home-MDB-SL
- ms-Exch-RMS-Computer-Accounts-Link-SL
- ms-Exch-Spam-Add-Header
- ms-Exch-Spam-Asf-Settings
- ms-Exch-Spam-Asf-Test-Bcc-Address
- ms-Exch-Spam-False-Positive-Cc
- ms-Exch-Spam-Flags
- ms-Exch-Spam-Modify-Subject
- ms-Exch-Spam-Outbound-Spam-Cc
- ms-Exch-Spam-Redirect-Address
- ms-Exch-Transport-Reseller-Settings-Link-SL
- ms-Exch-Hygiene-Configuration-Link
- ms-Exch-Accepted-Domain-BL
- ms-Exch-Malware-Filter-Config-Link
- ms-Exch-Hygiene-Configuration-Malware-BL
- ms-Exch-Hosted-Content-Filter-Config-Link
- ms-Exch-Hygiene-Configuration-Spam-BL
- ms-Exch-Content-Byte-Encoder-Type-For-7-Bit-Charsets
- ms-Exch-Content-Preferred-Internet-Code-Page-For-Shift-Jis
- ms-Exch-Content-Required-Char-Set-Coverage
- ms-Exch-Group-External-Member-Count
- ms-Exch-Group-Member-Count
- ms-Exch-Organization-Flags-2
- ms-Exch-RMSOnline-Certification-Location-Url
- ms-Exch-RMSOnline-Key-Sharing-Location-Url
- ms-Exch-RMSOnline-Licensing-Location-Url
- ms-Exch-Shadow-When-Soft-Deleted-Time
- ms-Exch-Throttling-Policy-Flags
- ms-Exch-Malware-Filter-Config-External-Sender-Admin-Address
- ms-Exch-Malware-Filter-Config-Internal-Sender-Admin-Address
- ms-Exch-Malware-Filtering-Scan-Timeout
- ms-Exch-Spam-Country-Block-List
- ms-Exch-Spam-Language-Block-List
- ms-Exch-Spam-Notify-Outbound-Recipients
- ms-Exch-Auth-App-Secret
- ms-Exch-Auth-Application-Identifier

- ms-Exch-Auth-Auth-Server-Type
- ms-Exch-Auth-Authorization-Url
- ms-Exch-Auth-Certificate-Data
- ms-Exch-Auth-Certificate-Thumbprint
- ms-Exch-Auth-Flags
- ms-Exch-Auth-Issuer-Name
- ms-Exch-Auth-Issuing-Url
- ms-Exch-Auth-Linked-Account
- ms-Exch-Auth-Metadata-Url
- ms-Exch-Auth-Realm
- ms-Exch-Mailflow-Policy-Countries
- ms-Exch-Mailflow-Policy-Keywords
- ms-Exch-Mailflow-Policy-Publisher-Name
- ms-Exch-Mailflow-Policy-Transport-Rules-Template-Xml
- ms-Exch-Mailflow-Policy-Version
- ms-Exch-Public-Folder-EntryId
- ms-Exch-Public-Folder-Mailbox
- ms-Exch-Public-Folder-Smtp-Address
- ms-Exch-Spam-Digest-Frequency
- ms-Exch-Spam-Quarantine-Retention
- ms-Exch-Transport-MaxRetriesForLocalSiteShadow
- ms-Exch-Transport-MaxRetriesForRemoteSiteShadow
- ms-Exch-Transport-Rule-Immutable-Id
- ms-Exch-WAC-Discovery-Endpoint
- ms-Exch-Anonymous-Throttling-Policy-State-Ex
- ms-Exch-Canary-Data-0
- ms-Exch-Canary-Data-1
- ms-Exch-Canary-Data-2
- ms-Exch-Correlation-Id
- ms-Exch-EAS-Throttling-Policy-State-Ex
- ms-Exch-EWS-Throttling-Policy-State-Ex
- ms-Exch-General-Throttling-Policy-State-Ex
- ms-Exch-IMAP-Throttling-Policy-State-Ex
- ms-Exch-OWA-Throttling-Policy-State-Ex
- ms-Exch-POP-Throttling-Policy-State-Ex
- ms-Exch-Powershell-Throttling-Policy-State-Ex
- ms-Exch-RCA-Throttling-Policy-State-Ex
- ms-Exch-Relocate-Tenant-Completion-Target-Vector
- ms-Exch-Relocate-Tenant-Flags
- ms-Exch-Relocate-Tenant-Safe-Lockdown-Schedule
- ms-Exch-Relocate-Tenant-Source-Forest
- ms-Exch-Relocate-Tenant-Start-Lockdown

- ms-Exch-Relocate-Tenant-Start-Retired
- ms-Exch-Relocate-Tenant-Start-Sync
- ms-Exch-Relocate-Tenant-Status
- ms-Exch-Relocate-Tenant-Target-Forest
- ms-Exch-Relocate-Tenant-Transition-Counter
- ms-Exch-Sync-Cookie
- ms-Exch-Adfs-Authentication-Raw-Configuration
- ms-Exch-Service-End-Point-URL
- ms-Exch-Sync-Service-Instance-New-Tenant-Max-Version
- ms-Exch-Sync-Service-Instance-New-Tenant-Min-Version

MAPI IDs added by Exchange 2013 RTM

The following MAPI IDs are added when you install Exchange 2013 RTM:

- 36066
- 36067

Extended rights added by Exchange 2013 RTM

The following table lists the extended rights that are added when you install Exchange 2013 RTM. Installing Exchange 2013 RTM doesn't modify any existing extended rights.

Identifier	Values
CN=ms-Exch-SMTP-Accept-XProxyFrom,CN=Extended-Rights,<ConfigurationContainerDN>	changetype: ntdsSchemaAdd displayName: Accept XProxyFrom objectClass: controlAccessRight rightsGuid: 5bee2b72-50d7-49c7-ba66-39a25daa1e92 validAccesses: 256

Object IDs added by Exchange 2013 RTM

The following attribute object IDs are added when you install Exchange 2013 RTM:

- 1.2.840.113556.1.4.7000.102.51794
- 1.2.840.113556.1.4.7000.102.51795
- 1.2.840.113556.1.4.7000.102.51792
- 1.2.840.113556.1.4.7000.102.51791
- 1.2.840.113556.1.4.7000.102.51789
- 1.2.840.113556.1.4.7000.102.51787
- 1.2.840.113556.1.4.7000.102.51788

- 1.2.840.113556.1.4.7000.102.51786
- 1.2.840.113556.1.4.7000.102.51790
- 1.2.840.113556.1.4.7000.102.51774
- 1.2.840.113556.1.4.7000.102.51773
- 1.2.840.113556.1.4.7000.102.51775
- 1.2.840.113556.1.4.7000.102.51798
- 1.2.840.113556.1.4.7000.102.51801
- 1.2.840.113556.1.4.7000.102.51800
- 1.2.840.113556.1.4.7000.102.51799
- 1.2.840.113556.1.4.7000.102.51805
- 1.2.840.113556.1.4.7000.102.51796
- 1.2.840.113556.1.4.7000.102.51797
- 1.2.840.113556.1.4.7000.102.51814
- 1.2.840.113556.1.4.7000.102.51813
- 1.2.840.113556.1.4.7000.102.51815
- 1.2.840.113556.1.4.7000.102.51816
- 1.2.840.113556.1.4.7000.102.51818
- 1.2.840.113556.1.4.7000.102.51812
- 1.2.840.113556.1.4.7000.102.51819
- 1.2.840.113556.1.4.7000.102.51806
- 1.2.840.113556.1.4.7000.102.51820
- 1.2.840.113556.1.4.7000.102.51821
- 1.2.840.113556.1.4.7000.102.51811
- 1.2.840.113556.1.4.7000.102.51807
- 1.2.840.113556.1.4.7000.102.51810
- 1.2.840.113556.1.4.7000.102.51808
- 1.2.840.113556.1.4.7000.102.51809
- 1.2.840.113556.1.4.7000.102.51830
- 1.2.840.113556.1.4.7000.102.51829
- 1.2.840.113556.1.4.7000.102.51824
- 1.2.840.113556.1.4.7000.102.51823
- 1.2.840.113556.1.4.7000.102.51827
- 1.2.840.113556.1.4.7000.102.51826
- 1.2.840.113556.1.4.7000.102.51822
- 1.2.840.113556.1.4.7000.102.51833
- 1.2.840.113556.1.4.7000.102.51832
- 1.2.840.113556.1.4.2183
- 1.2.840.113556.1.4.2184
- 1.2.840.113556.1.4.2185
- 1.2.840.113556.1.4.7000.102.51838
- 1.2.840.113556.1.4.7000.102.51836
- 1.2.840.113556.1.4.7000.102.51837

- 1.2.840.113556.1.4.7000.102.51839
- 1.2.840.113556.1.4.7000.102.51840
- 1.2.840.113556.1.4.7000.102.51876
- 1.2.840.113556.1.4.7000.102.51877
- 1.2.840.113556.1.4.7000.102.51878
- 1.2.840.113556.1.4.7000.102.51879
- 1.2.840.113556.1.4.7000.102.51880
- 1.2.840.113556.1.4.7000.102.51851
- 1.2.840.113556.1.4.7000.102.51852
- 1.2.840.113556.1.4.7000.102.51859
- 1.2.840.113556.1.4.7000.102.51860
- 1.2.840.113556.1.4.7000.102.51861
- 1.2.840.113556.1.4.7000.102.51864
- 1.2.840.113556.1.4.7000.102.51863
- 1.2.840.113556.1.4.7000.102.51862
- 1.2.840.113556.1.4.7000.102.51865
- 1.2.840.113556.1.4.7000.102.51866
- 1.2.840.113556.1.4.7000.102.51867
- 1.2.840.113556.1.4.7000.102.51868
- 1.2.840.113556.1.4.7000.102.51871
- 1.2.840.113556.1.4.7000.102.51870
- 1.2.840.113556.1.4.7000.102.51872
- 1.2.840.113556.1.4.7000.102.51875
- 1.2.840.113556.1.4.7000.102.51874
- 1.2.840.113556.1.4.7000.102.51873
- 1.2.840.113556.1.4.7000.102.51869
- 1.2.840.113556.1.4.7000.102.51915
- 1.2.840.113556.1.4.7000.102.51883
- 1.2.840.113556.1.4.7000.102.51914
- 1.2.840.113556.1.4.7000.102.51931
- 1.2.840.113556.1.4.7000.102.51932
- 1.2.840.113556.1.4.7000.102.51939
- 1.2.840.113556.1.4.7000.102.51930
- 1.2.840.113556.1.4.7000.102.51940
- 1.2.840.113556.1.4.7000.102.51933
- 1.2.840.113556.1.4.7000.102.51934
- 1.2.840.113556.1.4.7000.102.51935
- 1.2.840.113556.1.4.7000.102.51936
- 1.2.840.113556.1.4.7000.102.51952
- 1.2.840.113556.1.4.7000.102.51923
- 1.2.840.113556.1.4.7000.102.51927
- 1.2.840.113556.1.4.7000.102.51926

- 1.2.840.113556.1.4.7000.102.51922
- 1.2.840.113556.1.4.7000.102.51929
- 1.2.840.113556.1.4.7000.102.51928
- 1.2.840.113556.1.4.7000.102.51925
- 1.2.840.113556.1.4.7000.102.51924
- 1.2.840.113556.1.4.7000.102.51941
- 1.2.840.113556.1.4.7000.102.51942
- 1.2.840.113556.1.4.7000.102.51937
- 1.2.840.113556.1.4.7000.102.51943
- 1.2.840.113556.1.4.7000.102.51944
- 1.2.840.113556.1.4.7000.102.51938
- 1.2.840.113556.1.4.7000.102.51882
- 1.2.840.113556.1.4.7000.102.51881
- 1.2.840.113556.1.4.7000.102.51921
- 1.2.840.113556.1.4.7000.102.51918
- 1.2.840.113556.1.4.7000.102.51920
- 1.2.840.113556.1.4.7000.102.51916
- 1.2.840.113556.1.4.7000.102.51919
- 1.2.840.113556.1.4.7000.102.51917
- 1.2.840.113556.1.4.7000.102.51945
- 1.2.840.113556.1.4.7000.102.51946
- 1.2.840.113556.1.4.7000.102.51948
- 1.2.840.113556.1.4.7000.102.51949
- 1.2.840.113556.1.4.7000.102.51947
- 1.2.840.113556.1.4.7000.102.51950
- 1.2.840.113556.1.4.7000.102.51951
- 1.2.840.113556.1.4.7000.102.51954
- 1.2.840.113556.1.4.7000.102.51955
- 1.2.840.113556.1.4.7000.102.51953
- 1.2.840.113556.1.4.7000.102.51993
- 1.2.840.113556.1.4.7000.102.51995
- 1.2.840.113556.1.4.7000.102.51994
- 1.2.840.113556.1.4.7000.102.51998
- 1.2.840.113556.1.4.7000.102.51997
- 1.2.840.113556.1.4.7000.102.52004
- 1.2.840.113556.1.4.7000.102.52003
- 1.2.840.113556.1.4.7000.102.52002
- 1.2.840.113556.1.4.7000.102.52001
- 1.2.840.113556.1.4.7000.102.52005
- 1.2.840.113556.1.4.7000.102.52007
- 1.2.840.113556.1.4.7000.102.52006
- 1.2.840.113556.1.4.7000.102.51996

- 1.2.840.113556.1.4.7000.102.52008
- 1.2.840.113556.1.4.7000.102.52017
- 1.2.840.113556.1.4.7000.102.52014
- 1.2.840.113556.1.4.7000.102.52021
- 1.2.840.113556.1.4.7000.102.52020
- 1.2.840.113556.1.4.7000.102.52012
- 1.2.840.113556.1.4.7000.102.52011
- 1.2.840.113556.1.4.7000.102.52013
- 1.2.840.113556.1.4.7000.102.52018
- 1.2.840.113556.1.4.7000.102.52019
- 1.2.840.113556.1.4.7000.102.52022
- 1.2.840.113556.1.4.7000.102.52015
- 1.2.840.113556.1.4.7000.102.52016
- 1.2.840.113556.1.4.7000.102.52029
- 1.2.840.113556.1.4.7000.102.52030
- 1.2.840.113556.1.4.7000.102.52039
- 1.2.840.113556.1.4.7000.102.52041
- 1.2.840.113556.1.4.7000.102.52037
- 1.2.840.113556.1.4.7000.102.52035
- 1.2.840.113556.1.4.7000.102.52034
- 1.2.840.113556.1.4.7000.102.52036
- 1.2.840.113556.1.4.7000.102.52032
- 1.2.840.113556.1.4.7000.102.52033
- 1.2.840.113556.1.4.7000.102.52031
- 1.2.840.113556.1.4.7000.102.52024
- 1.2.840.113556.1.4.7000.102.52040
- 1.2.840.113556.1.4.7000.102.52023
- 1.2.840.113556.1.4.7000.102.52042
- 1.2.840.113556.1.4.7000.102.52051
- 1.2.840.113556.1.4.7000.102.52052
- 1.2.840.113556.1.4.7000.102.52053
- 1.2.840.113556.1.4.7000.102.52065
- 1.2.840.113556.1.4.7000.102.52043
- 1.2.840.113556.1.4.7000.102.52044
- 1.2.840.113556.1.4.7000.102.52045
- 1.2.840.113556.1.4.7000.102.52046
- 1.2.840.113556.1.4.7000.102.52047
- 1.2.840.113556.1.4.7000.102.52048
- 1.2.840.113556.1.4.7000.102.52049
- 1.2.840.113556.1.4.7000.102.52050
- 1.2.840.113556.1.4.7000.102.52063
- 1.2.840.113556.1.4.7000.102.52061

- 1.2.840.113556.1.4.7000.102.52059
- 1.2.840.113556.1.4.7000.102.52058
- 1.2.840.113556.1.4.7000.102.52055
- 1.2.840.113556.1.4.7000.102.52056
- 1.2.840.113556.1.4.7000.102.52054
- 1.2.840.113556.1.4.7000.102.52060
- 1.2.840.113556.1.4.7000.102.52057
- 1.2.840.113556.1.4.7000.102.52062
- 1.2.840.113556.1.4.7000.102.52064

The following class object IDs are added when you install Exchange 2013 RTM:

- 1.2.840.113556.1.5.7000.62.50161
- 1.2.840.113556.1.5.7000.62.50162
- 1.2.840.113556.1.5.7000.62.50163
- 1.2.840.113556.1.5.7000.62.50166
- 1.2.840.113556.1.5.7000.62.50164
- 1.2.840.113556.1.5.7000.62.50165
- 1.2.840.113556.1.5.7000.62.50167
- 1.2.840.113556.1.5.7000.62.50170
- 1.2.840.113556.1.5.7000.62.50172
- 1.2.840.113556.1.5.7000.62.50171
- 1.2.840.113556.1.5.7000.62.50173
- 1.2.840.113556.1.5.7000.62.50178
- 1.2.840.113556.1.5.7000.62.50174
- 1.2.840.113556.1.5.7000.62.50176
- 1.2.840.113556.1.5.7000.62.50177
- 1.2.840.113556.1.5.7000.62.50187
- 1.2.840.113556.1.5.7000.62.50188
- 1.2.840.113556.1.5.7000.62.50190
- 1.2.840.113556.1.5.7000.62.50189
- 1.2.840.113556.1.5.7000.62.50191
- 1.2.840.113556.1.5.7000.62.50192

Indexed attributes added by Exchange 2013 RTM

The following table lists the attributes that are added to the list of indexed attributes when you install Exchange 2013 RTM.

Attribute	Search flag value
ms-Exch-Is-Dirsync-Status-Pending	1
ms-Exch-Archive-GUID	9

ms-Exch-Accepted-Domain-Name	9
ms-Exch-Bypass-Audit	9
ms-Exch-Mailbox-Audit-Enable	19
ms-Exch-Default-Public-Folder-Mailbox	19
ms-Exch-OWA-Set-Photo-URL	16
ms-Exch-Organization-Upgrade-Policy-Link	1
ms-DS-GeoCoordinates-Altitude	1
ms-DS-GeoCoordinates-Latitude	1
ms-DS-GeoCoordinates-Longitude	1
ms-Exch-Mailbox-Database-Transport-Flags	16
ms-Exch-Extension-Custom-Attribute-1	1
ms-Exch-Extension-Custom-Attribute-2	1
ms-Exch-Extension-Custom-Attribute-3	1
ms-Exch-Extension-Custom-Attribute-4	1
ms-Exch-Extension-Custom-Attribute-5	1
ms-Exch-Recipient-SoftDeleted-Status	27
ms-Exch-When-Soft-Deleted-Time	17
ms-Exch-Device-Client-Type	1
ms-Exch-Team-Mailbox-Expiration	16
ms-Exch-Team-Mailbox-Expiry-Days	16
ms-Exch-Team-Mailbox-Owners	16
ms-Exch-Team-Mailbox-SharePoint-Linked-By	16

ms-Exch-Team-Mailbox-SharePoint-Url	16
ms-Exch-Team-Mailbox-Show-In-Client-List	16
ms-Exch-Home-MDB-SL	1
ms-Exch-Home-MTA-SL	1
ms-Exch-Mailbox-Move-Source-Archive-MDB-Link-SL	1
ms-Exch-Mailbox-Move-Source-MDB-Link-SL	1
ms-Exch-Mailbox-Move-Target-Archive-MDB-Link-SL	1
ms-Exch-Organization-Upgrade-Policy-Link-SL	1
ms-Exch-Previous-Archive-Database-SL	8
ms-Exch-Previous-Home-MDB-SL	8
ms-Exch-Auth-Issuer-Name	1
ms-Exch-Auth-Application-Identifier	1
ms-Exch-Transport-Rule-Immutable-Id	1
ms-Exch-Public-Folder-EntryId	24
ms-Exch-Public-Folder-Mailbox	24
ms-Exch-Public-Folder-Smtp-Address	24
ms-Exch-Relocate-Tenant-Completion-Target-Vector	8
ms-Exch-Relocate-Tenant-Flags	8
ms-Exch-Relocate-Tenant-Safe-Lockdown-Schedule	8

ms-Exch-Relocate-Tenant-Start-Lockdown	8
ms-Exch-Relocate-Tenant-Start-Retired	8
ms-Exch-Relocate-Tenant-Start-Sync	8
ms-Exch-Relocate-Tenant-Transition-Counter	8
ms-Exch-Sync-Cookie	8
ms-Exch-Relocate-Tenant-Source-Forest	9
ms-Exch-Relocate-Tenant-Status,	9
ms-Exch-Relocate-Tenant-Target-Forest	9

Global catalog attributes added by Exchange 2013 RTM

The following global catalog attributes are added by Exchange 2013 RTM:

- ms-Exch-Dirsync-Authority-Metadata
- ms-Exch-Dirsync-Status
- ms-Exch-Dirsync-Status-Ack
- ms-Exch-Edge-Sync-Config-Flags
- ms-Exch-Is-Dirsync-Status-Pending
- ms-Exch-Localization-Flags
- ms-Exch-Previous-Archive-Database
- ms-Exch-RoleGroup-Type
- ms-Exch-HAB-Root-Department-Link
- ms-Exch-Default-Public-Folder-Mailbox
- ms-Exch-Team-Mailbox-Expiration
- ms-Exch-Team-Mailbox-Expiry-Days
- ms-Exch-Team-Mailbox-Owners
- ms-Exch-Team-Mailbox-SharePoint-Linked-By
- ms-Exch-Team-Mailbox-SharePoint-Url
- ms-Exch-Team-Mailbox-Show-In-Client-List
- ms-Exch-Recipient-SoftDeleted-Status
- ms-Exch-When-Soft-Deleted-Time
- ms-Exch-Device-Client-Type
- ms-Exch-Extension-Custom-Attribute-1
- ms-Exch-Extension-Custom-Attribute-2
- ms-Exch-Extension-Custom-Attribute-3
- ms-Exch-Extension-Custom-Attribute-4

- ms-Exch-Extension-Custom-Attribute-5
- ms-Exch-Archive-Database-Link-SL
- ms-Exch-Disabled-Archive-Database-Link-SL
- ms-Exch-Home-MDB-SL
- ms-Exch-Home-MTA-SL
- ms-Exch-Mailbox-Move-Source-Archive-MDB-Link-SL
- ms-Exch-Mailbox-Move-Source-MDB-Link-SL
- ms-Exch-Mailbox-Move-Storage-MDB-Link-SL
- ms-Exch-Mailbox-Move-Target-Archive-MDB-Link-SL
- ms-Exch-Mailbox-Move-Target-MDB-Link-SL
- ms-Exch-Previous-Archive-Database-SL
- ms-Exch-Previous-Home-MDB-SL
- ms-Exch-RMS-Computer-Accounts-Link-SL
- ms-Exch-Group-Member-Count
- ms-Exch-Group-External-Member-Count
- ms-Exch-Correlation-Id
- ms-Exch-Relocate-Tenant-Completion-Target-Vector,
- ms-Exch-Relocate-Tenant-Flags
- ms-Exch-Relocate-Tenant-Safe-Lockdown-Schedule
- ms-Exch-Relocate-Tenant-Source-Forest
- ms-Exch-Relocate-Tenant-Start-Lockdown
- ms-Exch-Relocate-Tenant-Start-Retired
- ms-Exch-Relocate-Tenant-Start-Sync
- ms-Exch-Relocate-Tenant-Status
- ms-Exch-Relocate-Tenant-Target-Forest
- ms-Exch-Relocate-Tenant-Transition-Counter
- ms-Exch-Sync-Cookie

Disjoint namespace scenarios

Exchange Server 2013 > Planning and deployment > Active Directory >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-10*

This topic provides information about the concept of disjoint namespaces and the supported scenarios for deploying Microsoft Exchange 2013 in a domain that has a disjoint namespace.

Contents

Exchange 2013 and disjoint namespaces

Allow Exchange 2013 servers to access domain controllers that are disjoint

View DNS and NetBIOS name-related information of a computer running Windows Server 2008

DNS and NetBIOS domain names

First, some background. Every computer that is on the Internet has a Domain Name System (DNS) name. This is also known as the *machine name* or *host name*. Every computer running the Windows operating system with networking capabilities also has a NetBIOS name.

A computer running Windows in an Active Directory domain has both a DNS domain name and a NetBIOS domain name, as follows:

- **DNS domain name** The DNS domain name consists of one or more subdomains separated by a dot (.) and is terminated by a top-level domain name. For example, in the DNS domain name corp.contoso.com, the subdomains are corp and contoso and the top-level domain name is com.
- **NetBIOS domain name** Typically, the NetBIOS domain name is the subdomain of the DNS domain name. For example, if the DNS domain name is contoso.com, the NetBIOS domain name is contoso. If the DNS domain name is corp.contoso.com, the NetBIOS domain name is corp.

Note:

To find DNS and NetBIOS information for computers running Windows Server 2008, see View DNS and NetBIOS name-related information of a computer running Windows Server 2008.

A computer in an Active Directory domain also has a primary DNS suffix and can have additional DNS suffixes. By default, the primary DNS suffix is the same as the DNS domain name. For detailed steps about how to change the primary DNS suffix, see the procedures later in this topic.

You define the DNS domain name and NetBIOS domain name of an Active Directory domain when you configure the first domain controller in the domain. For more information about configuring domain controllers, see Domain Controller Roles and Active Directory Domain Services Overview.

Disjoint namespaces

In most domain topologies, the primary DNS suffix of the computers in the domain is the same as the DNS domain name.

In some cases, you may require these namespaces to be different. This is called a *disjoint namespace*. For example, a merger or acquisition may cause you to have a topology with a disjoint namespace. In addition, if DNS management in your company is split between administrators who manage Active Directory and administrators who manage networks, you may need to have a topology with a disjoint namespace.

A disjoint namespace scenario is one in which the primary DNS suffix of a computer doesn't match the DNS domain name where that computer resides. The computer with the primary DNS suffix that

doesn't match is said to be *disjoint*. Another disjoint namespace scenario occurs if the NetBIOS domain name of a domain controller doesn't match the DNS domain name.

Exchange 2013 and disjoint namespaces

Exchange 2013 supports the following three scenarios for deploying Exchange in a domain that has a disjoint namespace:

- **Primary DNS suffix and DNS domain name are different** The primary DNS suffix of the domain controller isn't the same as the DNS domain name. Computers that are members of the domain can be either disjoint or not disjoint.
- **Member computer is disjoint** A member computer in an Active Directory domain is disjoint, even though the domain controller is not disjoint.
- **NetBIOS name of domain controller differs from subdomain of its DNS domain name** The NetBIOS domain name of the domain controller isn't the same as the subdomain of the DNS domain name of that domain controller.

These scenarios are detailed in the following sections.

Note:

It's supported to run Exchange 2013 in the disjoint namespace scenarios described in this topic. However, if you have a disjoint namespace scenario that isn't one of the scenarios described in this topic, you must work with Microsoft Services to deploy Exchange 2013. For more information, see Microsoft Services.

Scenario: Primary DNS suffix and DNS domain name are different

In this scenario, the primary DNS suffix of the domain controller isn't the same as the DNS domain name. The domain controller is disjoint in this scenario. Computers that are members of the domain, including Exchange servers and Microsoft Outlook client computers, can have a primary DNS suffix that either matches the primary DNS suffix of the domain controller or matches the DNS domain name.

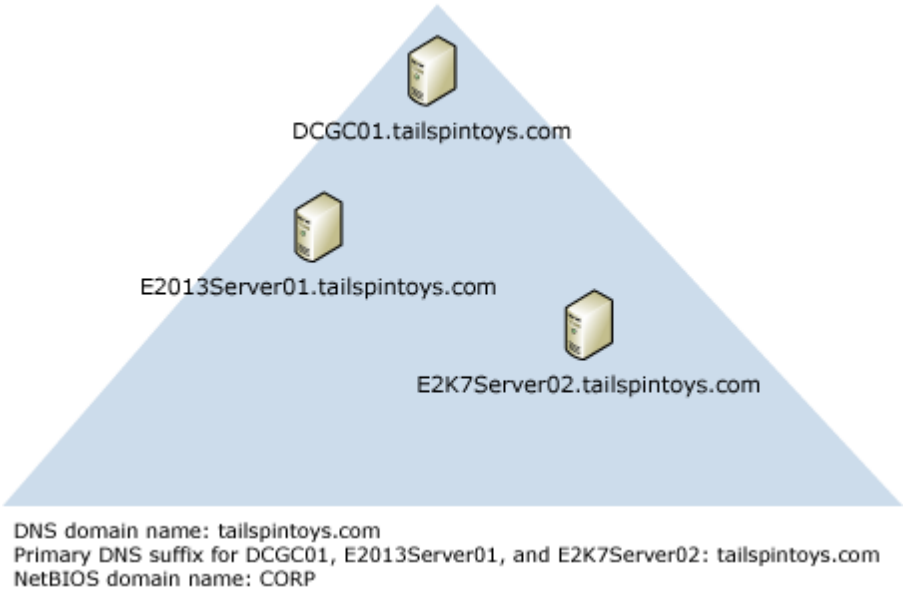
Scenario: Member computer is disjoint

In this scenario, the primary DNS suffix of a member computer on which Exchange 2013 is installed isn't the same as the DNS domain name, even though the primary DNS suffix of the domain controller is the same as the DNS domain name. In this scenario, you have a domain controller that isn't disjoint and a member computer that is disjoint. Member computers that are running Outlook can have a primary DNS suffix that either matches the primary DNS suffix of the disjoint Exchange server or matches the DNS domain name.

Scenario: NetBIOS name of domain controller differs from subdomain of its DNS domain name

In this scenario, the NetBIOS domain name of the domain controller isn't the same as the DNS domain name of the same domain controller.

NetBIOS domain name doesn't match DNS domain name



Allow Exchange 2013 servers to access domain controllers that are disjoint

To allow Exchange 2013 servers to access domain controllers that are disjoint, you must modify the **msDS-AllowedDNSSuffixes** Active Directory attribute on the domain object container. You must add both of the DNS suffixes to the attribute. For detailed steps about how to modify the attribute, see The computer's primary DNS suffix does not match the FQDN of the domain where it resides.

In addition, to make sure that the DNS suffix search list contains all DNS namespaces that are deployed within the organization, you must configure the search list for each computer in the domain that is disjoint. The list of namespaces should include not only the primary DNS suffix of the domain controller and the DNS domain name, but also any additional namespaces for other servers with which Exchange may interoperate (such as monitoring servers or servers for third-party applications). You can do this by setting Group Policy for the domain. For more information about Group Policy, see the following topics:

- Group Policy Frequently Asked Questions (FAQ)
- New group policies for DNS in Windows Server 2003
- Group Policy

For detailed steps about how to configure the DNS suffix search list Group Policy, see Configure the

DNS suffix search list for a disjoint namespace.

View DNS and NetBIOS name-related information of a computer running Windows Server 2008

1. Click **Start**, right-click **Computer**, and then click **Properties**.
2. In **System**, the DNS host name and primary DNS suffix are displayed under **Computer name, domain, and workgroup settings** next to **Full computer name**. The DNS domain name is displayed next to **Domain**.
3. Click **Change settings**.
4. In **System Properties**, on the **Computer Name** tab, click **Change**.
5. In **Computer Name/Domain Changes**, click **More**. The primary DNS suffix is displayed under **Primary DNS suffix of this computer**. The NetBIOS computer name is displayed under **NetBIOS computer name**.

To change the primary DNS suffix, type the new primary DNS suffix under **Primary DNS suffix of this computer**, and then click **OK**.

6. From a Command Prompt window, type **set**. The variable USERDNSDOMAIN displays the DNS domain name. The variable USERDOMAIN displays the NetBIOS domain name.

Configure the DNS suffix search list for a disjoint namespace

Planning and deployment > Active Directory > Disjoint namespace scenarios >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-08*

This topic explains how to use the Group Policy Management console (GPMC) to configure the Domain Name System (DNS) suffix search list. In some Microsoft Exchange 2013 scenarios, if you have a disjoint namespace, you must configure the DNS suffix search list to include multiple DNS suffixes.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes
- To perform this procedure, the account you use must be delegated membership in the Domain Admins group.
- Confirm that you have installed .NET Framework 3.0 on the computer on which you will install the GPMC.

Note:

The current version of the GPMC that you can download from the Microsoft Download Center operates on the 32-bit versions of the Windows Server 2003 and Windows XP operating systems and can remotely manage Group Policy objects on 32-bit and 64-bit domain controllers. This version of the GPMC doesn't include a 64-bit version, and the 32-bit version doesn't run on 64-bit platforms. The 32-bit version of Windows Server 2008 and the 32-bit version of Windows Vista both include a 32-bit version of the GPMC. The 64-bit version of Windows Server 2008 and the 64-bit version of Windows Vista both include a 64-bit version of the GPMC.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the GPMC to configure the DNS suffix search list

1. On a 32-bit computer in your domain, install GPMC with Service Pack 1 (SP1). For download information, see Group Policy Management Console with Service Pack 1.

Note:

If you have a computer in your domain running Windows Server 2008 or Windows Vista, you can skip this step.

2. Click **Start** > **Programs** > **Administrative Tools** > **Group Policy Management**.
3. In **Group Policy Management**, expand the forest and the domain in which you will apply Group Policy. Right-click **Group Policy Objects**, and then click **New**.
4. In **New GPO**, type a name for the policy, and then click **OK**.
5. Right-click the new policy that you created in Step 4, and then click **Edit**.
6. In **Group Policy Management Editor**, expand **Computer Configuration**, expand **Policies**, expand **Administrative Templates**, expand **Network**, and then click **DNS Client**.
7. Right-click **DNS Suffix Search List**, click **All Tasks**, and then click **Edit**.
8. On the **DNS Suffix Search List Properties** page, select **Enabled**. In the **DNS Suffixes** box, type the primary DNS suffix of the disjoint computer, the DNS domain name, and any additional namespaces for other servers with which Exchange may interoperate, such as monitoring servers or servers for third-party applications. Click **OK**.
9. In **Group Policy Management**, expand **Group Policy Objects**, and then select the policy that you created in Step 4. On the **Scope** tab, scope the policy so that it applies to only the computers that are disjoint.

How do you know this worked?

To verify that you have successfully completed your migration, do the following:

- After you install Exchange 2013, verify that you can send email messages inside and outside your

organization.

For more information

Windows Server Group Policy

Group Policy

Disjoint namespace scenarios

Exchange 2013 system requirements

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-25*

Before you install Microsoft Exchange Server 2013, we recommend that you review this topic to ensure that your network, hardware, software, clients, and other elements meet the requirements for Exchange 2013. In addition, make sure you understand the coexistence scenarios that are supported for Exchange 2013 and earlier versions of Exchange.

Supported coexistence scenarios

The following table lists the scenarios in which coexistence between Exchange 2013 and earlier versions of Exchange is supported.

Coexistence of Exchange 2013 and earlier versions of Exchange Server

Exchange version	Exchange organization coexistence
Exchange Server 2003 and earlier versions	Not supported
Exchange 2007	Supported with the following minimum versions of Exchange: • [1]Update Rollup 10 for Exchange 2007 Service Pack 3 (SP3) on all Exchange 2007 servers in the organization, including Edge Transport servers. • Exchange 2013 Cumulative Update 2 (CU2) or later on all Exchange 2013 servers in the

	organization.
Exchange 2010	Supported with the following minimum versions of Exchange: • [2]Exchange 2010 SP3 on all Exchange 2010 servers in the organization, including Edge Transport servers. • Exchange 2013 CU2 or later on all Exchange 2013 servers in the organization.
Mixed Exchange 2010 and Exchange 2007 organization	Supported with the following minimum versions of Exchange: • [1]Update Rollup 10 for Exchange 2007 SP3 on all Exchange 2007 servers in the organization, including Edge Transport servers. • [2]Exchange 2010 SP3 on all Exchange 2010 servers in the organization, including Edge Transport servers. • Exchange 2013 CU2 or later on all Exchange 2013 servers in the organization.

[1] If you want to create an EdgeSync Subscription between an Exchange 2007 Hub Transport server and an Exchange 2013 SP1 Edge Transport server, you need to install Exchange 2007 SP3 Update Rollup 13 or later on the Exchange 2007 Hub Transport server.

[2] If you want to create an EdgeSync Subscription between an Exchange 2010 Hub Transport server and an Exchange 2013 SP1 Edge Transport server, you need to install Exchange 2010 SP3 Update Rollup 5 or later on the Exchange 2010 Hub Transport server.

Supported hybrid deployment scenarios

Exchange 2013 supports hybrid deployments with Office 365 tenants that have been upgraded to the latest version of Office 365. For more information about specific hybrid deployments, see **Hybrid deployment prerequisites**.

Network and directory servers

The following table lists the requirements for the network and the directory servers in your Exchange 2013 organization.

Network and directory server requirements for Exchange 2013

Component	Requirement
Schema master	By default, the schema master runs on the first Active Directory domain controller installed in a forest. The schema master must be running one of the following: • Windows Server 2012 R2 Standard or Datacenter[1] • Windows Server 2012 Standard or Datacenter • Windows Server 2008 R2 Standard or Enterprise • Windows Server 2008 R2 Datacenter RTM or later • Windows Server 2008 Standard or Enterprise (32-bit or 64-bit) • Windows Server 2008 Datacenter RTM or later • Windows Server 2003 Standard Edition with Service Pack 2 (SP2) or later (32-bit or 64-bit) • Windows Server 2003 Enterprise Edition with SP2 or later (32-bit or 64-bit)
Global catalog server	In each Active Directory site where you plan to install Exchange 2013, you must have at least one global catalog server running one of the following: • Windows Server 2012 R2 Standard or Datacenter[1] • Windows Server 2012 Standard or Datacenter • Windows Server 2008 R2 Standard or Enterprise • Windows Server 2008 R2 Datacenter RTM or later • Windows Server 2008 Standard or Enterprise (32-bit or 64-bit)

	• Windows Server 2008 Datacenter RTM or later • Windows Server 2003 Standard Edition with Service Pack 2 (SP2) or later (32-bit or 64-bit) • Windows Server 2003 Enterprise Edition with SP2 or later (32-bit or 64-bit) For more information about global catalog servers, see What is the Global Catalog.
Domain controller	In each Active Directory site where you plan to install Exchange 2013, you must have at least one writeable domain controller running one of the following: • Windows Server 2012 R2 Standard or Datacenter[1] • Windows Server 2012 Standard or Datacenter • Windows Server 2008 R2 Standard or Enterprise SP1 or later • Windows Server 2008 R2 Datacenter RTM or later • Windows Server 2008 Standard or Enterprise SP1 or later (32-bit or 64-bit) • Windows Server 2008 Datacenter RTM or later • Windows Server 2003 Standard Edition with Service Pack 2 (SP2) or later (32-bit or 64-bit) • Windows Server 2003 Enterprise Edition with SP2 or later (32-bit or 64-bit)
Active Directory forest	Active Directory must be at Windows Server 2003 forest functionality mode or higher[2].
DNS namespace support	Exchange 2013 supports the following domain name system (DNS) namespaces: • Contiguous • Noncontiguous • Single label domains

	• Disjoint For more information about DNS namespaces supported by Exchange, see Microsoft Knowledge Base article 2269838, Microsoft Exchange compatibility with Single Label Domains, Disjoined Namespaces, and Discontiguous Namespaces.
IPv6 support	In Exchange 2013, IPv6 is supported only when IPv4 is also installed and enabled. If Exchange 2013 is deployed in this configuration, and the network supports IPv4 and IPv6, all Exchange servers can send data to and receive data from devices, servers, and clients that use IPv6 addresses. For more information, see IPv6 support in Exchange 2013.

[1] Windows Server 2012 R2 is supported only with Exchange 2013 SP1 or later.

[2] Windows Server 2012 R2 forest functionality mode is supported only with Exchange 2013 SP1 or later.

Directory server architecture

The use of 64-bit Active Directory domain controllers increases directory service performance for Exchange 2013.

Note:

In multi-domain environments, on Windows Server 2008 domain controllers that have the Active Directory language locale set to Japanese, your servers may not receive some attributes that are stored on an object during inbound replication. For more information, see Microsoft Knowledge Base article 949189, A Windows Server 2008 domain controller that is configured with the Japanese language locale may not apply updates to attributes on an object during inbound replication.

Installing Exchange 2013 on directory servers

For security and performance reasons, we recommend that you install Exchange 2013 only on member servers and not on Active Directory directory servers. However, you can't run DCPromo on a computer running Exchange 2013. After Exchange 2013 is installed, changing its role from a

member server to a directory server, or vice versa, isn't supported.

Hardware

The recommended hardware requirements for Exchange 2013 servers vary depending on a number of factors including the server roles that are installed and the anticipated load that will be placed on the servers.

Note:
For information about deploying Exchange in a virtualized environment, see Exchange 2013 virtualization.

Hardware requirements for Exchange 2013

Component	Requirement	Notes
Processor	• x64 architecture-based computer with Intel processor that supports Intel 64 architecture (formerly known as Intel EM64T) • AMD processor that supports the AMD64 platform • Intel Itanium IA64 processors not supported	See the "Operating system" section later in this topic for supported operating systems.
Memory	Varies depending on Exchange roles that are installed: • **Mailbox** 8GB minimum • **Client Access** 4GB minimum • **Mailbox and Client Access combined** 8GB minimum • **Edge Transport** 4GB minimum	None.
Paging file size	The page file size minimum and maximum must be set to physical RAM plus 10 MB	The recommended page file size also accounts for the memory that's needed to collect information if the operating system stops

		unexpectedly. On 64-bit operating systems, memory can be written as a dump file to the paging file. This file must reside on the boot volume of the server. For more information about the configuration options that are available for memory dump data, see Knowledge Base article 254649, Overview of memory dump file options for Windows Vista, Windows Server 2008 R2, Windows Server 2008, Windows Server 2003, Windows XP, and Windows 2000.
Disk space	• At least 30 GB on the drive on which you install Exchange • An additional 500 MB of available disk space for each Unified Messaging (UM) language pack that you plan to install • 200 MB of available disk space on the system drive • A hard disk that stores the message queue database on with at least 500 MB of free space.	The minimum space requirements detailed here don't account for disk subsystem requirements for adequate performance.
Drive	DVD-ROM drive, local or network accessible	None.

Screen resolution	1024 x 768 pixels or higher	None.
File format	Disk partitions formatted as NTFS file systems, which applies to the following partitions: • System partition • Partitions that store Exchange binary files or files generated by Exchange diagnostic logging Disk partitions containing the following types of files can be formatted as ReFS: • Partitions containing transaction log files • Partitions containing database files • Partitions containing content indexing files	None.

Operating system

The following table lists the supported operating systems for Exchange 2013.

Important:

We don't support the installation of Exchange 2013 on a computer that's running in Windows Server Core mode. The computer must be running the full installation of Windows Server. If you want to install Exchange 2013 on a computer that's running in Windows Server Core mode, you must convert the server to a full installation of Windows Server by doing one of the following:

- **Windows Server 2008 R2** Reinstall Windows Server and select the **Full Installation** option.
- **Windows Server 2012 R2** or **Windows Server 2012** Convert your Windows Server Core mode server to a full installation by running the following command.

```
Install-WindowsFeature Server-Gui-Mgmt-Infra, Server-Gui-Shell -Re
```

Supported operating systems for Exchange 2013

Component	Requirement

Mailbox, Client Access, and Edge Transport server roles	One of the following: • Windows Server 2012 R2 Standard or Datacenter[1] • Windows Server 2012 Standard or Datacenter • Windows Server 2008 R2 Standard with Service Pack 1 (SP1) • Windows Server 2008 R2 Enterprise with Service Pack 1 (SP1) • Windows Server 2008 R2 Datacenter RTM or later
Management tools	One of the following: • Windows Server 2012 R2 Standard or Datacenter[1] • Windows Server 2012 Standard or Datacenter • Windows Server 2008 R2 Standard with SP1 • Windows Server 2008 R2 Enterprise with SP1 • Windows Server 2008 R2 Datacenter RTM or later • 64-bit edition of Windows 8.1[2] • 64-bit edition of Windows 8 • 64-bit edition of Windows 7 with Service Pack 1

[1] Windows Server 2012 R2 is supported only with Exchange 2013 SP1 or later.

[2] Windows 8.1 is supported only with Exchange 2013 SP1 or later.

Supported Windows Management Framework versions for Exchange 2013

Exchange version	Windows Management Framework 3.0	Windows Management Framework 4.0
Exchange 2013 CU3	X	
Exchange 2013 SP1 and later	X	X

.NET Framework

We strongly recommend that you use the latest version of .NET Framework that's supported by the release of Exchange you're installing.

Exchange version	.NET Framework 4.5.1	.NET Framework 4.5
Exchange 2013 CU3		X
Exchange 2013 SP1 and later	X	X

Supported clients

Exchange 2013 and Exchange Online support the following minimum versions of Microsoft Outlook and Microsoft Entourage for Mac:

- Outlook 2013 (15.0.4420.1017)
- Outlook 2010 Service Pack 1 with the Outlook 2010 November 2012 update (14.0.6126.5000). For more information, see Description of the Outlook 2010 update: November 13, 2012.
- Outlook 2007 Service Pack 3 with the Outlook 2007 November 2012 update (12.0.6665.5000). For more information, see Description of the Outlook 2007 update: November 13, 2012.
- Entourage 2008 for Mac, Web Services Edition
- Outlook for Mac 2011

Important:

The information above provides the minimum versions required for a client to connect to Exchange and Exchange Online. We strongly recommend that you install the latest available service packs and updates available so that your users receive the best possible experience when connecting to Exchange and Exchange Online.

Outlook clients earlier than Outlook 2007 are not supported. Email clients on Mac operating systems that require DAV, such as Entourage 2008 for Mac RTM and Entourage 2004, are not supported.

Outlook Web App supports several browsers on a variety of operating systems and devices. For detailed information, see What's new for Outlook Web App in Exchange 2013.

Exchange 2013 prerequisites

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-25*

This topic provides the steps for installing the necessary Windows Server 2012 R2, Windows Server 2012 and Windows Server 2008 R2 with Service Pack 1 (SP1) operating system prerequisites for the

Microsoft Exchange 2013 Mailbox, Client Access, and Edge Transport server roles. It also provides the prerequisites required to install the Exchange 2013 management tools on Windows 8, Windows 8.1, and Windows 7 client computers.

What do you need to know before you begin?

- The Edge Transport server role is available starting with Exchange 2013 SP1.
- Make sure that the functional level of your forest is at least Windows Server 2003, and that the schema master is running Windows Server 2003 with Service Pack 2 or later. For more information about the Windows functional level, see Managing Domains and Forests.
- The full installation option of Windows Server 2012 R2, Windows Server 2012 and Windows Server 2008 R2 SP1 must be used for all servers running Exchange 2013 server roles or management tools.
- You must first join the computer to the appropriate internal Active Directory forest and domain.
- Some prerequisites require you to reboot the server to complete installation.
- Install the latest Windows updates on your computer. For more information, see Deployment security checklist.

Note:

If you're installing the Mailbox server role and you intend for the server to be a member of a database availability group (DAG), you must be running Windows Server 2012 R2 Standard or Datacenter Edition, Windows Server 2012 Standard or Datacenter Edition, or Windows Server 2008 R2 SP1 Enterprise Edition. Windows Server 2008 R2 SP 1 Standard Edition doesn't support the features needed for DAGs.
You can't upgrade Windows when Exchange is installed on the server.
To upgrade to Microsoft Unified Communications Managed API (UCMA) 4.0, you must first uninstall any previous versions of UCMA that are installed by using **Add/Remove programs**.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Active Directory preparation

The computer you want to use to prepare Active Directory for Exchange 2013 has specific prerequisites that must be met.

Install the following software on the computer that will be used to prepare Active Directory:

- Microsoft .NET Framework 4.5
- The version of Windows Management Framework that corresponds to the version of Exchange 2013 you're installing.
  - **Exchange 2013 CU3** Windows Management Framework 3.0
  - **Exchange 2013 SP1 or later** Windows Management Framework 4.0

Note:
.NET Framework 4.5 and Windows Management Framework 3.0 are included with Windows Server 2012 and don't need to be installed separately. .NET Framework 4.5 and Windows Management Framework 4.0 are included with Windows Server 2012 R2 and don't need to be installed separately.

After you've installed the software listed above, complete the following steps to install the Remote Tools Administration Pack. After you've installed the Remote Tools Administration Pack you'll be able to use the computer to prepare Active Directory. For more information about preparing Active Directory, see Prepare Active Directory and domains.

1. Open Windows PowerShell.
2. Install the Remote Tools Administration Pack.
   - On a Windows Server 2012 R2 or Windows Server 2012 computer, run the following command.

```
Install-WindowsFeature RSAT-ADDS
```

   - On a Windows Server 2008 R2 SP1 computer, run the following command.

```
Add-WindowsFeature RSAT-ADDS
```

Windows Server 2012 R2 and Windows Server 2012 prerequisites

The prerequisites that are needed to install Exchange 2013 on a Windows Server 2012 R2 or Windows Server 2012 computer depends on which Exchange roles you want to install. Read the section below that matches the roles you want to install.

Mailbox or Client Access server roles

Follow the instructions in this section to install the prerequisites on Windows Server 2012 R2 or Windows Server 2012 computers where you want to do one of the following:

- Install only the Mailbox server role on a computer.
- Install only the Client Access server role on a computer.
- Install both the Mailbox and Client Access server roles on the same computer.

Do the following to install the required Windows roles and features:

1. Open Windows PowerShell.
2. Run the following command to install the required Windows components.

```
Install-WindowsFeature AS-HTTP-Activation, Desktop-Experience, NET-Framework-45-Features, RPC-over-HTTP-proxy, RSAT-Clustering, RSAT-Clustering-CmdInterface, RSAT-Clustering-Mgmt, RSAT-Clustering-PowerShell, Web-Mgmt-Console, WAS-Process-Model, Web-Asp-Net45, Web-Basic-Auth, Web-Client-Auth, Web-Digest-Auth, Web-Dir-Browsing, Web-Dyn-Compression, Web-Http-Errors, Web-Http-Logging, Web-Http-Redirect, Web-Http-Tracing, Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Lgcy-Mgmt-Console, Web-Metabase, Web-Mgmt-Console, Web-Mgmt-Service, Web-Net-Ext45, Web-Request-Monitor, Web-Server, Web-Stat-Compression, Web-Static-Content, Web-Windows-Auth, Web-WMI, Windows-Identity-Foundation
```

After you've installed the operating system roles and features, install Microsoft Unified Communications Managed API 4.0, Core Runtime 64-bit.

Edge Transport server role

Follow the instructions in this section to install the prerequisites on Windows Server 2012 R2 or Windows Server 2012 computers where you want to install the Edge Transport server role on a computer.

Do the following to install the required Windows roles and features:

1. Open Windows PowerShell.
2. Run the following command to install the required Windows components.

```
Install-WindowsFeature ADLDS
```

Windows Server 2008 R2 SP1 prerequisites

The prerequisites that are needed to install Exchange 2013 on a Windows Server 2008 R2 SP1 computer depends on which Exchange roles you want to install. Read the section below that matches the roles you want to install.

Mailbox or Client Access server roles

Follow the instructions in this section to install the prerequisites on Windows Server 2008 R2 SP1

computers where you want to do one of the following:

- Install only the Mailbox server role on a computer.
- Install only the Client Access server role on a computer.
- Install both the Mailbox and Client Access server roles on the same computer.

Do the following to install the required Windows roles and features:

1. Open Windows PowerShell.
2. Run the following command to load the Server Manager module.

```
Import-Module ServerManager
```

3. Run the following command to install the required Windows components.

```
Add-WindowsFeature Desktop-Experience, NET-Framework, NET-HTTP-Activation, RPC-over-HTTP-proxy, RSAT-Clustering, RSAT-Web-Server, WAS-Process-Model, Web-Asp-Net, Web-Basic-Auth, Web-Client-Auth, Web-Digest-Auth, Web-Dir-Browsing, Web-Dyn-Compression, Web-Http-Errors, Web-Http-Logging, Web-Http-Redirect, Web-Http-Tracing, Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Lgcy-Mgmt-Console, Web-Metabase, Web-Mgmt-Console, Web-Mgmt-Service, Web-Net-Ext, Web-Request-Monitor, Web-Server, Web-Stat-Compression, Web-Static-Content, Web-Windows-Auth, Web-WMI
```

After you've installed the operating system roles and features, install the following software in the order shown:

1. Microsoft .NET Framework 4.5
2. The version of Windows Management Framework that corresponds to the version of Exchange 2013 you're installing.
   - **Exchange 2013 CU3** Windows Management Framework 3.0
   - **Exchange 2013 SP1 or later** Windows Management Framework 4.0
3. Microsoft Unified Communications Managed API 4.0, Core Runtime 64-bit
4. Microsoft Knowledge Base article KB974405 (Windows Identity Foundation)
5. Knowledge Base article KB2619234 (Enable the Association Cookie/GUID that is used by RPC over HTTP to also be used at the RPC layer in Windows 7 and in Windows Server 2008 R2)
6. Knowledge Base article KB2533623 (Insecure library loading could allow remote code execution)

Note:

This hotfix may already be installed if you've configured Windows Update to install security updates on your computer.

Edge Transport server role

Follow the instructions in this section to install the prerequisites on Windows Server 2008 R2 SP1

computers where you want to install the Edge Transport server role on a computer.

Do the following to install the required Windows roles and features:

1. Open Windows PowerShell.
2. Run the following command to load the Server Manager module.

```
Import-Module ServerManager
```

3. Run the following command to install the required Windows components.

```
Add-WindowsFeature NET-Framework, ADLDS
```

After you've installed the operating system roles and features, install the following software in the order shown:

1. Microsoft .NET Framework 4.5
2. The version of Windows Management Framework that corresponds to the version of Exchange 2013 you're installing.
   - **Exchange 2013 CU3** Windows Management Framework 3.0
   - **Exchange 2013 SP1 or later** Windows Management Framework 4.0

Windows 7 prerequisites

Follow the instructions in this section to install the prerequisites on domain-joined Windows 7 64-bit computers where you want to install the Exchange management tools.

1. Open **Control Panel**, and then select **Programs**.
2. Click **Turn Windows features on or off**.
3. Navigate to **Internet Information Services** > **Web Management Tools** > **IIS 6 Management Compatibility**.
4. Select the check box for **IIS 6 Management Console**, and then click **OK**.

After you've installed the operating system features, install the following software in the order shown:

1. Microsoft .NET Framework 4.5
2. The version of Windows Management Framework that corresponds to the version of Exchange 2013 you're installing.
   - **Exchange 2013 CU3** Windows Management Framework 3.0
   - **Exchange 2013 SP1 or later** Windows Management Framework 4.0
3. Knowledge Base article KB974405 (Windows Identity Foundation)

Windows 8 and Windows 8.1 prerequisites

The Exchange management tools can be installed on a domain-joined computer with a default install of Windows 8 or Windows 8.1 64-bit.

Prepare Active Directory and domains

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-25*

Before you install Microsoft Exchange Server 2013, you need to prepare your Active Directory forest and its domains. Exchange needs to prepare Active Directory so that it can store information about your users' mailboxes and the configuration of Exchange servers in the organization. If you aren't familiar with Active Directory forests or domains, check out Active Directory Domain Services Overview.

There are a couple of ways you can prepare Active Directory for Exchange. The first is to let the Exchange 2013 Setup wizard do it for you. If you don't have a large Active Directory deployment, and you don't have a separate team that manages Active Directory, we recommend using the wizard. The account you use will need to be a member of both the Schema Admins and Enterprise Admins security groups. For more information about how to use the Setup wizard, check out Install Exchange 2013 using the Setup wizard.

If you have a large Active Directory deployment, or if a separate team manages Active Directory, this topic is for you. Following the steps in this topic gives you much more control over each stage of preparation, and who can do each step. For example, Exchange administrators might not have the permissions needed to extend the Active Directory schema.

What do you need to know before you begin?

1. Extend the Active Directory schema

2. Prepare Active Directory

3. Prepare Active Directory domains

How do you know this worked?

Curious about what's happening when Active Directory is being prepared for Exchange? Check out What changes in Active Directory when Exchange 2013 is installed?

What do you need to know before you begin?

- Estimated time to complete: 10-15 minutes or more (not including Active Directory replication), depending on organization size and the number of child domains.
- The computer you use to run these steps needs to meet the Exchange 2013 system requirements. Also, your Active Directory forest needs to meet the requirements in the "Network and directory servers" section in Exchange 2013 system requirements.
- If your organization has multiple Active Directory domains, we recommend the following:

- Do the steps below from an Active Directory site that has an Active Directory server from every domain.
- Install the first Exchange server in an Active Directory site with a writeable global catalog server from every domain.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

1. Extend the Active Directory schema

The first step in getting your organization ready for Exchange 2013 is to extend the Active Directory schema. Exchange stores a lot of information in Active Directory but before it can do that, it needs to add and update classes, attributes, and other items. If you're curious about what's changed when your schema is extended, check out Exchange 2013 Active Directory schema changes.

Before you extend your schema, there are a few things to keep in mind:

- The account you're logged in as needs to be a member of the Schema Admins and Enterprise Admins security groups.
- The computer where you'll run the command to extend the schema needs to be in the same Active Directory domain and site as the schema master.
- If you use the *DomainController* parameter, make sure to use the name of the domain controller that's the schema master.
- The only way to extend the schema for Exchange is to use the steps in this topic or use Exchange 2013 Setup. Other ways of extending the schema aren't supported.

Tip:

If you don't have a separate team that manages your Active Directory schema, you can skip this step and go directly to Step 2. Prepare Active Directory. If the schema isn't extended in step 1, the commands in step 2 will extend the schema for you. If you decide to skip step 1, the information you need to keep in mind above still applies.

When you're ready, do the following to extend your Active Directory schema. If you have multiple Active Directory forests, make sure you're logged into the right one.

1. Make sure the computer is ready to run Exchange 2013 Setup. To see what you need to run Setup, check out the Active Directory preparation section in Exchange 2013 prerequisites.
2. Open a Windows Command Prompt window and go to where you downloaded the Exchange installation files.
3. Run the following command to extend the schema.

```
Setup.exe /PrepareSchema /IAcceptExchangeServerLicenseTerms
```

After Setup finishes extending the schema, you'll need to wait while Active Directory replicates the changes to all of your domain controllers. If you want to check on how replication is going, you can use the `repadmin` tool. `Repadmin` is included as part of the Active Directory Domain Services Tools

feature in Windows Server 2012 R2, Windows Server 2012, and Windows Server 2008 R2. For more information about how to use it, see Repadmin.

2. Prepare Active Directory

Now that the Active Directory schema has been extended, you can prepare other parts of Active Directory for Exchange 2013. During this step, Exchange will create containers, objects, and other items in Active Directory that it'll use to store information. The collection of all of the Exchange containers, objects, attributes, and so on, is called the *Exchange organization*.

Before you prepare Active Directory for Exchange, there are a few things to keep in mind:

- The account you're logged in as needs to be a member of the Enterprise Admins security group. If you skipped step 1 because you want the *PrepareAD* command to extend the schema, the account you use also needs to be a member of the Schema Admins security group.
- The computer where you'll run the command needs to be in the same Active Directory domain and site as the schema master. It'll also need to contact all of the domains in the forest on TCP port 389.
- Wait until Active Directory has replicated the changes made in step 1 to all of your domain controllers before you do this step.

When you run the command below to prepare Active Directory for Exchange, you'll need to name the Exchange organization. This name is used internally by Exchange and isn't normally seen by users. The name of the company where Exchange is being installed is often used for the organization name. The name you use won't affect the functionality of Exchange or determine what you can use for email addresses. You can name it anything you want, as long as you keep the following in mind:

- You can use any uppercase or lowercase letters from A to Z.
- You can use numbers 0 to 9.
- The name can contain spaces as long as they're not at the beginning or end of the name.
- You can use a hyphen or dash in the name.
- The name can be up to 64 characters but can't be blank.
- The name can't be changed after it's set.

When you're ready, do the following to prepare Active Directory for Exchange. If the organization name you want to use has spaces, enclose the name in quotation marks (").

1. Open a Windows Command Prompt window and go to where you downloaded the Exchange installation files.
2. Run the following command:

```
Setup.exe /PrepareAD /OrganizationName:"<organization name>" /IAcceptExchangeServerLicenseTerms
```

After Setup finishes preparing Active Directory for Exchange, you'll need to wait while Active Directory replicates the changes to all of your domain controllers. If you want to check on how

replication is going, you can use the `repadmin` tool. `repadmin` is included as part of the Active Directory Domain Services Tools feature in Windows Server 2012 R2, Windows Server 2012, and Windows Server 2008 R2. For more information about how to use the tool, see Repadmin.

3. Prepare Active Directory domains

The final step to get Active Directory ready for Exchange is to prepare each of the Active Directory domains where Exchange will be installed or where mail-enabled users will be located. This step creates additional containers and security groups, and sets permissions so that Exchange can access them.

If you have multiple domains in your Active Directory forest, you have a couple of choices in how you prepare them. Select the option that matches what you want to do. If you only have one domain, you can skip this step because the *PrepareAD* command in step 2 already prepared the domain for you.

Prepare all of the domains in my Active Directory forest

To prepare all of your Active Directory domains, you can use the *PrepareAllDomains* parameter when you run Setup. Setup will prepare every domain for Exchange in your Active Directory forest for you.

Before you prepare all of the domains in your Active Directory forest, keep the following in mind:

- The account you use needs to be a member of the Enterprise Admins security group.
- Wait until Active Directory has replicated the changes made in step 2 to all of your domain controllers. If you don't, you might get an error when you try to prepare the domain.

When you're ready, do the following to prepare all of the domains in your Active Directory forest for Exchange.

1. Open a Windows Command Prompt window and go to where you downloaded the Exchange installation files.
2. Run the following command:

```
Setup.exe /PrepareAllDomains /
IAcceptExchangeServerLicenseTerms
```

Let me choose which Active Directory domains I want to prepare

If you want to choose which Active Directory domains you want to prepare, you can use the *PrepareDomain* parameter when you run Setup. When you use the *PrepareDomain* parameter, you need to include the fully qualified domain name (FQDN) of the domain you want to prepare.

Before you prepare the domains in your Active Directory forest, keep the following in mind:

- The account you use needs permissions depending on when the domain was created.
  - **Domain created before PrepareAD was run** If the domain was created **before** you ran the *PrepareAD* command in step 2 above, then the account you use needs to be a member of the Domain Admins group in the domain you want to prepare.
  - **Domain created after PrepareAD was run** If the domain was created **after** you ran the *PrepareAD* command in step 2 above, then the account you use needs to 1) be a member of the Organization Management role group and 2) be a member of the Domain Admins group in the domain you want to prepare.
- Wait until Active Directory has replicated the changes made in step 2 to all of your domain controllers. If you don't, you might get an error when you try to prepare the domain.
- You need to prepare every domain where an Exchange server will be installed. You'll also need to prepare any domain that'll contain mail-enabled users, even if those domains won't contain any Exchange servers.
- You don't need to run the *PrepareDomain* command in the domain where the *PrepareAD* command was run. The *PrepareAD* command prepares that domain automatically.

When you're ready, do the following to prepare an individual domain in your Active Directory forest for Exchange.

1. Open a Windows Command Prompt window and go to where you downloaded the Exchange installation files.
2. Run the following command. Include the FQDN of the domain you want to prepare. If you want to prepare the domain you're running the command in, you don't have to include the FQDN.

```
Setup.exe /PrepareDomain:<FQDN of the domain you want to prepare> /IAcceptExchangeServerLicenseTerms
```

3. Repeat the steps for each Active Directory domain where you'll install an Exchange server or where mail-enabled users will be located.

How do you know this worked?

Once you've done all the steps above, you can check to make sure everything's gone smoothly. To do so, you'll use a tool called Active Directory Service Interfaces Editor (ADSI Edit). ADSI Edit is included as part of the Active Directory Domain Services Tools feature in Windows Server 2012 R2, Windows Server 2012, and Windows Server 2008 R2. If you want to know more about it, check out ADSI Edit (adsiedit.msc).

Warning:
Never change values in ADSI Edit unless you're told to do so by Microsoft support. Changing values in ADSI Edit can cause irreparable harm to your Exchange organization and Active Directory.

The easiest way to check whether Active Directory has been prepared correctly is to check whether the **msExchProductId** property has the right value. It'll have the right value if the schema has been

extended and the *PrepareAD* command completed successfully.

To check whether the **msExchProductId** property has the right value, do the following:

1. Press WIN+R, type `adsiedit.msc`, and then press Enter.
2. In ADSI Edit, right-click **ADSI Edit** in the navigation pane, and then click **Connect to...**
3. In **Connection Settings**, select **Select a well known Naming Context**, and then choose **Configuration**. Click **OK**.
4. Expand **Configuration [<domain FQDN>]**, **CN=Configuration,DC=domain,DC=com**, **CN=Services**, **CN=Microsoft Exchange**.
5. Right-click **CN=<your Exchange organization name>**, and then select **Properties**.
6. Make sure the value in **msExchangeProductId** matches the value in the Exchange 2013 Active Directory versions table for the version of Exchange 2013 you're installing.

If you're separating the steps to extend the schema and prepare Active Directory, you can use ADSI Edit to check additional properties to make sure each step has completed successfully. Use the information in the following list to make sure these properties have the right values.

- In the **Schema** naming context, verify that the **rangeUpper** property on **ms-Exch-Schema-Verision-Pt** is set to the value shown for your version of Exchange 2013 in the Exchange 2013 Active Directory versions table.

- In the **Configuration** naming context, verify that the **objectVersion** property in the CN=<*your organization*>,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=<*domain*> container is set to the value shown for your version of Exchange 2013 in the Exchange 2013 Active Directory versions table.

- In the **Default** naming context, verify that the **objectVersion** property in the **Microsoft Exchange System Objects** container under DC=<*root domain* is set to the value shown for your version of Exchange 2013 in the Exchange 2013 Active Directory versions table.

You can also check the Exchange setup log to verify that Active Directory preparation has completed successfully. For more information, see Verify an Exchange 2013 installation. You won't be able to use the **Get-ExchangeServer** cmdlet mentioned in the Verify an Exchange 2013 installation topic until you've completed the installation of at least one Mailbox server role and one Client Access server role in an Active Directory site.

Exchange 2013 Active Directory versions

The following table shows you the Exchange 2013 objects in Active Directory that get updated each time you install a new version of Exchange 2013. You can compare the object versions you see with the values in the table below to verify that the version of Exchange 2013 you installed successfully updated Active Directory during installation.

	Exchange	msExchProduc	rangeUpper	objectVersion	objectVersion

	version	tId			
Naming context		Configuration	Schema	Default	Configuration
Container		CN=<*your organization*>, CN=Microsoft Exchange, CN=Services, CN=Configuration, DC=<*domain*>	ms-Exch-Schema-Version-Pt	Microsoft Exchange System Objects	CN=<*your organization*>, CN=Microsoft Exchange, CN=Services, CN=Configuration, DC=<*domain*>
	Exchange 2013 CU6	15.00.0995.029	15303	13236	15965
	Exchange 2013 CU5	15.00.0913.022	15300	13236	15870
	Exchange 2013 SP1	15.00.0847.032	15292	13236	15844
	Exchange 2013 CU3	15.00.0775.038	15283	13236	15763
	Exchange 2013 CU2	15.00.0712.024 **Important:** If **msExchProductId** is 15.00.712.022, you have an out-of-date version of Exchange 2013 CU2. To avoid problems moving public folder mailboxes and	15281	13236	15688

		to make sure you can install future updates, you need to install the latest version of Exchange 2013 CU2. For more information, see Public folders in Release notes for Exchange 2013.			
	Exchange 2013 CU1	15.00.0620.029	15254	13236	15614
	Exchange 2013 RTM	15.00.0516.032	15137	13236	15449

Deploy a new installation of Exchange 2013

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-31*

Before you begin your installation of Microsoft Exchange Server 2013, see Planning and deployment for important planning information, and information about system requirements and prerequisites.

The following topics provide information about deploying a new installation of Exchange 2013 in your organization:

Checklist: Perform a new installation of Exchange 2013

Install Exchange 2013 using the Setup wizard

Install Exchange 2013 using unattended mode

Install the Exchange 2013 Edge Transport role using the Setup wizard

Delegate the installation of an Exchange 2013 server

After you've completed your installation, see Exchange 2013 post-Installation tasks.

Checklist: Perform a new installation of Exchange 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-28*

Use this checklist to deploy Microsoft Exchange Server 2013. Before you start working with this checklist, make sure you're familiar with the concepts discussed in:

- Planning and deployment
- Deployment security checklist

This checklist is generic in that it provides guidance for a typical scenario.

Note:

The Exchange Server Deployment Assistant provides you with customized step-by-step guidance about how to deploy Exchange Server. The Deployment Assistant can help you deploy a new installation of Exchange Server 2013, upgrade a previous version to Exchange 2013, or configure a hybrid deployment of Exchange 2013 and Exchange Online. To learn more, see Exchange Server Deployment Assistant.

Checklist for a new installation of Exchange 2013

Done?	Task	Topic
	1. Read the release notes.	Release notes for Exchange 2013
2. Verify system requirements.	Exchange 2013 system requirements	
3. Confirm prerequisite steps are done.	Exchange 2013 prerequisites	

4. Configure disjoint namespace. **Note:** This step is optional. It's only necessary if your organization is running a disjoint namespace.	Disjoint namespace scenarios	
5. Install the Mailbox server role.	Install Exchange 2013 using the Setup wizard	
6. Install the Client Access server role.	Install Exchange 2013 using the Setup wizard	
	7. Install the Edge Transport server role. **Note:** This step is optional. It's only necessary if you want to install an Edge Transport server. For more information, see Edge Transport servers.	Install the Exchange 2013 Edge Transport role using the Setup wizard
	8. Create an EdgeSync subscription. This step is optional. It's only necessary if you've installed an Edge Transport server and want to configure an EdgeSync subscription between your Edge Transport server and a Hub Transport server. For more information, see Edge Subscriptions.	Configure Internet mail flow through a subscribed Edge Transport server
	9. Configure Transport.	Create a Send connector
	10. Add additional accepted	Add additional accepted

	domains.	domains
	11. Configure email address policies.	Configure the default email address policy
12. Configure settings on virtual directories, including the offline address book, Exchange Web Services, Exchange Administration Center (EAC), Outlook Web App, and Exchange ActiveSync virtual directories. **Note:** This step is necessary if you want to use Exchange Web Services, Outlook Anywhere, or the offline address book. It also may be required if you need to change any of the default settings for EAC, Outlook Web App, or Exchange ActiveSync.	Configure external URLs Configure internal URLs	
13. Add digital certificates on the Client Access server.	Configure an SSL certificate	
14. Configure Unified Messaging. **Note:** This step is optional. It's only necessary if you want to use Unified Messaging in your organization.	Deploying voice mail and UM	
	15. Configure additional Unified Messaging and Lync Server settings. **Note:** This step is optional. It's only	Deploying Exchange 2013 UM and Lync Server overview

	necessary if you've configured Unified Messaging in your organization and want to integrate it with Lync Server.	
16. Post-installation tasks.	Exchange 2013 post-Installation tasks	

Install Exchange 2013 using the Setup wizard

Exchange Server 2013 > Planning and deployment > Deploy a new installation of Exchange 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-19*

This topic explains how to use the Microsoft Exchange Server 2013 Setup wizard to install the Exchange 2013 Mailbox and Client Access roles on a computer. For more information about planning and deploying Exchange 2013, see Planning and deployment.

If you want to install the Exchange 2013 Edge Transport role on a computer, see Install the Exchange 2013 Edge Transport role using the Setup wizard. The Edge Transport role can't be installed on the same computer as the Mailbox or Client Access server roles.

Tip:

Have you heard about the Exchange Server Deployment Assistant? It's a free online tool that helps you quickly deploy Exchange 2013 in your organization by asking you a few questions and creating a customized deployment checklist just for you. If you want to learn more about it, go to Exchange Server Deployment Assistant.

Note:

After you install any server roles on a computer running Exchange 2013, you can't use the Exchange 2013 Setup wizard to add any additional server roles to this computer. If you want to add more server roles to a computer, you must either use Add or Remove Programs from Control Panel or use Setup.exe from a Command Prompt window.

For information about tasks to complete after installation, see Exchange 2013 post-Installation tasks.

What do you need to know before you begin?

- Estimated time to complete: 60 minutes

- Make sure you've read the release notes prior to installing Exchange 2013. For more information, see Release notes for Exchange 2013.
- Each organization requires at a minimum one Client Access server and one Mailbox server in the Active Directory forest. Additionally, each Active Directory site that contains a Mailbox server must also contain at least one Client Access server. If you're separating your server roles, we recommend installing the Mailbox server role first.
- The computer you install Exchange 2013 on must have a supported operating system (such as Windows Server 2008 R2 with Service Pack 1 (SP1) or Windows Server 2012), have enough disk space, be a member of an Active Directory domain, and satisfy other requirements. For information about system requirements, see Exchange 2013 system requirements.
- To run Exchange 2013 setup, you must install Microsoft .NET Framework 4.5, Windows Management Framework 3.0, and other required software. To understand the prerequisites for all server roles, see Exchange 2013 prerequisites.
- You must ensure the account you use is delegated membership in the Schema Admins group if you haven't previously prepared the Active Directory schema. If you're installing the first Exchange 2013 server in the organization, the account you use must have membership in the Enterprise Admins group. If you've already prepared the schema and aren't installing the first Exchange 2013 server in the organization, the account you use must be a member of the Exchange 2013 Organization Management role group.

Administrators who are members of the Delegated Setup role group can deploy Exchange 2013 servers that have been previously provisioned by a member of the Organization Management management role group.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Caution:

After you install Exchange 2013 on a server, you must not change the server name. Renaming a server after you have installed an Exchange 2013 server role is not supported.

Install Exchange Server 2013

If you're installing the first Exchange 2013 server in the organization, and the Active Directory preparation steps have not been performed, the account you use must have membership in the Enterprise Administrators group. If you haven't previously prepared the Active Directory Schema, the account must also be a member of the Schema Admins group. For information about preparing Active Directory for Exchange 2013, see Prepare Active Directory and domains. If you have already performed the Schema and Active Directory preparation steps, the account you use must be a member of the Delegated Setup management role group or the Organization Management role group.

Note:

To download the latest version of Exchange 2013, see Updates for Exchange 2013.

1. Log on to the computer on which you want to install Exchange 2013.

2. Navigate to the network location of the Exchange 2013 installation files.
3. Start Exchange 2013 Setup by double-clicking `Setup.exe`

Important:

If you have User Access Control (UAC) enabled, you must right-click `Setup.exe` and select **Run as administrator**.

4. On the **Check for Updates?** page, choose whether you want Setup to connect to the Internet and download product and security updates for Exchange 2013. If you select **Connect to the Internet and check for updates**, Setup will download updates and apply them prior to continuing. If you select **Don't check for updates right now**, you can download and install updates manually later. We recommend that you download and install updates now. Click **Next** to continue.
5. The **Introduction** page begins the process of installing Exchange into your organization. It will guide you through the installation. Several links to helpful deployment content are listed. We recommend that you visit these links prior to continuing setup. Click **Next** to continue.
6. On the **License Agreement** page, review the software license terms. If you agree to the terms, select **I accept the terms in the license agreement**, and then click **Next**.
7. On the **Recommended settings** page, select whether you want to use the recommended settings. If you select **Use recommended settings**, Exchange will automatically send error reports and information about your computer hardware and how you use Exchange to Microsoft. If you select **Don't use recommended settings**, these settings remain disabled but you can enable them at any time after Setup completes. For more information about these settings and how information sent to Microsoft is used, click **?**.
8. On the **Server Role Selection** page, choose whether you want to install the **Mailbox role**, the **Client Access role**, both roles, or just the **Management Tools** on this computer. You can add additional server roles later if you choose not to install them during this installation. An organization must have at least one Mailbox role and at least one Client Access server role installed. They can be installed on the same computer or on separate computers. The management tools are installed automatically if you install any server role.

Select **Automatically install Windows Server roles and features that are required to install Exchange Server** to have the Setup wizard install required Windows prerequisites. You may need to reboot the computer to complete the installation of some Windows features. If you don't select this option, you must install the Windows features manually.

Note:

This option installs only the Windows features required by Exchange. You must manually install other prerequisites manually. For more information, see Exchange 2013 prerequisites.

Click **Next** to continue.

9. On the **Installation Space and Location** page, either accept the default installation location or click **Browse** to choose a new location. Make sure that you have enough disk space available in the location where you want to install Exchange. Click **Next** to continue.
10. If this is the first Exchange server in your organization, on the **Exchange Organization** page, type a name for your Exchange organization. The Exchange organization name can contain only the

following characters:

- A through Z
- a through z
- 0 through 9
- Space (not leading or trailing)
- Hyphen or dash

Note:

The organization name can't contain more than 64 characters. The organization name can't be blank.

If you want to use the Active Directory split permissions model, select **Apply Active Directory split permission security model to the Exchange organization**.

Caution:

Most organizations don't need to apply the Active Directory split permissions model. If you need to separate management of Active Directory security principals and Exchange configuration, Role Based Access Control (RBAC) split permissions might work for you. For more information, click **?**.

Click **Next** to continue.

11. If you're installing the Mailbox role, on the **Malware Protection Settings** page, choose whether you want to enable or disable malware scanning. If you disable malware scanning, it can be enabled in the future. Click **Next** to continue.
12. On the **Readiness Checks** page, view the status to determine if the organization and server role prerequisite checks completed successfully. If they haven't completed successfully, you must resolve any reported errors before you can install Exchange 2013. You don't need to exit Setup when resolving some of the prerequisite errors. After resolving a reported error, click **back** and then click **Next** to run the prerequisite check again. Be sure to also review any warnings that are reported. If all readiness checks have completed successfully, click **Next** to install Exchange 2013.
13. On the **Completion** page, click **Finish**.
14. Restart the computer after Exchange 2013 has completed.
15. Complete your deployment by performing the tasks provided in Exchange 2013 post-Installation tasks.

How do you know this worked?

To verify that you've successfully installed Exchange 2013, see Verify an Exchange 2013 installation.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Install Exchange 2013 using unattended mode

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-19*

To perform an unattended setup, you must install Microsoft Exchange Server 2013 from the command prompt. For more information about planning and deploying Exchange 2013, see Planning and deployment.

We recommend that the Edge Transport role be installed in a perimeter network outside of your organization's internal Active Directory forest. While you can install the Edge Transport server role on a domain-joined computer, doing so will only enable domain management of Windows features and settings. The Edge Transport role itself doesn't use Active Directory. Instead, it uses the Active Directory Lightweight Directory Services (AD LDS) Windows feature to store configuration and recipient information. For more information about the Edge Transport role, see Edge Transport servers.

Tip:

Have you heard about the Exchange Server Deployment Assistant? It's a free online tool that helps you quickly deploy Exchange 2013 in your organization by asking you a few questions and creating a customized deployment checklist just for you. If you want to learn more about it, go to Exchange Server Deployment Assistant.

Note:

After you install any server roles on a computer running Exchange 2013, you can't use the Exchange 2013 Setup wizard to add any additional server roles to this computer. If you want to add more server roles to a computer, you must either use Add or Remove Programs from Control Panel or use Setup.exe from a Command Prompt window.
The Edge Transport role can't be installed on the same computer as the Mailbox or Client Access server roles.

For information about tasks to complete after installation, see Exchange 2013 post-Installation tasks.

What do you need to know before you begin?

The following information applies to all Exchange 2013 server roles.

- Make sure you've read the release notes prior to installing Exchange 2013. For more information, see Release notes for Exchange 2013.

- The computer you install Exchange 2013 on must have a supported operating system (such as Windows Server 2008 R2 with Service Pack 1 (SP1), Windows Server 2012 R2, or Windows Server 2012), have enough disk space, and satisfy other requirements. For information about system requirements, see Exchange 2013 system requirements.
- To run Exchange 2013 setup, you must install Microsoft .NET Framework 4.5, Windows Management Framework, and other required software. To understand the prerequisites for all server roles, see Exchange 2013 prerequisites.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Caution:

After you install Exchange 2013 on a server, you must not change the server name. Renaming a server after you have installed an Exchange 2013 server role is not supported.

The following information applies to the Exchange 2013 Mailbox and Client Access server roles.

- Estimated time to complete: 60 minutes
- Each organization requires at a minimum one Client Access server and one Mailbox server in the Active Directory forest. Additionally, each Active Directory site that contains a Mailbox server must also contain at least one Client Access server. If you're separating your server roles, we recommend installing the Mailbox server role first.
- The computer you install Exchange 2013 on must be a member of an Active Directory domain.
- You must ensure the account you use is delegated membership in the Schema Admins group if you haven't previously prepared the Active Directory schema. If you're installing the first Exchange 2013 server in the organization, the account you use must have membership in the Enterprise Admins group. If you've already prepared the schema and aren't installing the first Exchange 2013 server in the organization, the account you use must be a member of the Exchange 2013 Organization Management management role group.

Administrators who are members of the Delegated Setup role group can deploy Exchange 2013 servers that have been previously provisioned by a member of the Organization Management role group.

The following information applies to the Exchange 2013 Edge Transport server role.

- Estimated time to complete: 40 minutes
- The Edge Transport role is available with Exchange 2013 SP1 or later.
- You need to configure the primary DNS suffix on the computer. For example, if the fully qualified domain name of your computer is edge.contoso.com, the DNS suffix for the computer is contoso.com. For more information, see Primary DNS Suffix is missing.
- Exchange 2007 and Exchange 2010 Hub Transport servers need an update before you can create an EdgeSync Subscription between them and an Exchange 2013 Edge Transport server. If you don't install this update, the EdgeSync Subscription won't work correctly. For more information, see the "Supported coexistence scenarios" section in Exchange 2013 system requirements.
- Make sure the account you use is a member of the local Administrators group on the computer you're installing the Edge Transport role.

Use Setup.exe to install Exchange 2013 in unattended mode

Note:
To download the latest version of Exchange 2013, see Updates for Exchange 2013.

1. Log on to the computer on which you want to install Exchange 2013.
2. Navigate to the network location of the Exchange 2013 installation files.
3. At the command prompt, run the applicable command for your organization.

Important:
If you have User Access Control (UAC) enabled, you must run `Setup.exe` from an elevated command prompt.

```
Setup.exe [/Mode:<setup mode>] [/
IAcceptExchangeServerLicenseTerms]
[/Roles:<server roles to install>] [/
InstallWindowsComponents]
[/OrganizationName:<name for the new Exchange
organization>]
[/TargetDir:<target directory>] [/SourceDir:<source
directory>]
[/UpdatesDir:<directory from which to install updates>]
[/DomainController:<FQDN of domain controller>] [/
DisableAMFiltering]
[/AnswerFile:<filename>] [/DoNotStartTransport]
[/EnableErrorReporting] [/CustomerFeedbackEnabled:<True |
False>]
[/AddUmLanguagePack:<UM language pack name>]
[/RemoveUmLanguagePack:<UM language pack name>]
[/NewProvisionedServer:<server>] [/
RemoveProvisionedServer:<server>]
[/MdbName:<mailbox database name>] [/DbFilePath:<Edb file
path>]
[/LogFolderPath:<log folder path>] [/
ActiveDirectorySplitPermissions:<True | False>]
[/TenantOrganizationConfig:<path>]
```

4. Setup copies the setup files locally to the computer on which you're installing Exchange 2013.
5. Setup checks the prerequisites, including all prerequisites specific to the server roles that you're installing. If you haven't met all the prerequisites, Setup fails and returns an error message that

explains the reason for the failure. If you've met all the prerequisites, Setup installs Exchange 2013.

6. Restart the computer after Exchange 2013 has completed.
7. Complete your deployment by performing the tasks provided in Exchange 2013 post-Installation tasks.

Examples

The following are examples of using Setup.exe:

- **Setup.exe /mode:Install /role:ClientAccess,Mailbox /OrganizationName:MyOrg /IAcceptExchangeServerLicenseTerms**

This command creates an Exchange 2013 organization in Active Directory called MyOrg, installs the Client Access server role, Mailbox server role, and the management tools and also accepts the Exchange 2013 licensing terms.

- **Setup.exe /mode:Install /role:ClientAccess,Mailbox /TargetDir:"C:\Exchange Server" /IAcceptExchangeServerLicenseTerms**

This command installs the Client Access server role, the Mailbox server role, and the management tools to the "C:\Exchange Server" directory. This command assumes an Exchange 2013 organization has already been prepared.

- **Setup.exe /mode:Install /r:CA,MB /IAcceptExchangeServerLicenseTerms**

This command installs the Client Access server role, the Mailbox server role, and the management tools to the default installation location.

- **Setup.exe /mode:Install /r:EdgeTransport /IAcceptExchangeServerLicenseTerms**

This command installs the Edge Transport server role and the management tools to the default installation location.

- **Setup.exe /mode:Install /r:ET /IAcceptExchangeServerLicenseTerms**

This command installs the Edge Transport server role and the management tools to the default installation location.

- **Setup.exe /mode:Uninstall /IAcceptExchangeServerLicenseTerms**

This command completely removes Exchange 2013 from the server and removes this server's Exchange configuration from Active Directory.

- **Setup.exe /PrepareAD /on:"My Org" /IAcceptExchangeServerLicenseTerms**

This command creates an Exchange organization called My Org and prepares Active Directory for Exchange 2013.

- **C:\ExchangeServer\bin\Setup.exe /m:Install /r:ClientAccess /SourceDir:d:\amd64 /IAcceptExchangeServerLicenseTerms**

This command adds the Client Access server role to an existing Exchange 2013 server using D:\amd64 as the source directory.

- **Setup.exe /role:ClientAccess,Mailbox /UpdatesDir:"C:\ExchangeServer\New Patches" /IAcceptExchangeServerLicenseTerms**

This command updates ExchangeServer.msi with patches from the specified directory, and then

installs the Client Access server role, Mailbox server role, and the management tools. If a language pack bundle is included in this directory, the language pack is also installed.

- **Setup.exe /mode:Install /role:ClientAccess,Mailbox /DomainController:DC01 /IAcceptExchangeServerLicenseTerms**

This command uses the domain controller DC01 to query and make changes to Active Directory while installing the Client Access server role, Mailbox server role, and the management tools.

- **Setup.exe /mode:Install /role:ClientAccess /AnswerFile:c:\ExchangeConfig.txt /IAcceptExchangeServerLicenseTerms**

This command installs the Client Access server role by using the settings in the ExchangeConfig.txt file.

- **Setup.exe /rprs:Exchange03 /IAcceptExchangeServerLicenseTerms**

This command removes the object Exchange03 from Active Directory.

- **Setup.exe /AddUmLanguagePack:ko-KR /IAcceptExchangeServerLicenseTerms**

This command installs the Korean Unified Messaging language pack from the %ExchangeSourceDir%\ServerRoles\UnifiedMessaging directory.

How do you know this worked?

To verify that you've successfully installed Exchange 2013, see Verify an Exchange 2013 installation.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Install the Exchange 2013 Edge Transport role using the Setup wizard

Exchange Server 2013 > Planning and deployment > Deploy a new installation of Exchange 2013 >

Applies to: *Exchange Server*

Topic Last Modified: *2014-06-19*

This topic explains how to use the Microsoft Exchange Server 2013 Setup wizard to install the Exchange 2013 Edge Transport server role on a computer. The Edge Transport role is available with Exchange 2013 Service Pack 1 (SP1) or later. For more information about planning and deploying Exchange 2013, see Planning and deployment.

We recommend that the Edge Transport role be installed in a perimeter network outside of your

organization's internal Active Directory forest. While you can install the Edge Transport server role on a domain-joined computer, doing so will only enable domain management of Windows features and settings. The Edge Transport role itself doesn't use Active Directory. Instead, it uses the Active Directory Lightweight Directory Services (AD LDS) Windows feature to store configuration and recipient information. For more information about the Edge Transport role, see Edge Transport servers.

If you want to install the Exchange 2013 Mailbox or Client Access roles on a computer, see Install Exchange 2013 using the Setup wizard. The Edge Transport role can't be installed on the same computer as the Mailbox or Client Access server roles.

Tip:

Have you heard about the Exchange Server Deployment Assistant? It's a free online tool that helps you quickly deploy Exchange 2013 in your organization by asking you a few questions and creating a customized deployment checklist just for you. If you want to learn more about it, go to Exchange Server Deployment Assistant.

For information about tasks to complete after installation, see Exchange 2013 post-Installation tasks.

What do you need to know before you begin?

- Estimated time to complete: 40 minutes
- Make sure you've read the release notes prior to installing Exchange 2013. For more information, see Release notes for Exchange 2013.
- The computer you install Exchange 2013 on must have a supported operating system (such as Windows Server 2008 R2 with SP1, Windows Server 2012 R2, or Windows Server 2012), have enough disk space, and satisfy other requirements. For information about system requirements, see Exchange 2013 system requirements.
- To run Exchange 2013 setup, you must install Microsoft .NET Framework 4.5, Windows Management Framework, and other required software. To understand the prerequisites for all server roles, see Exchange 2013 prerequisites.
- You need to configure the primary DNS suffix on the computer. For example, if the fully qualified domain name of your computer is edge.contoso.com, the DNS suffix for the computer is contoso.com. For more information, see Primary DNS Suffix is missing.
- Exchange 2007 and Exchange 2010 Hub Transport servers need an update before you can create an EdgeSync Subscription between them and an Exchange 2013 Edge Transport server. If you don't install this update, the EdgeSync Subscription won't work correctly. For more information, see the "Supported coexistence scenarios" section in Exchange 2013 system requirements.
- Make sure the account you use is a member of the local Administrators group on the computer you're installing the Edge Transport role.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Caution:

After you install Exchange 2013 on a server, you must not change the server name. Renaming a server after you have installed an Exchange 2013 server role is not supported.

Install Exchange Server 2013

Note:

To download the latest version of Exchange 2013, see Updates for Exchange 2013.

1. Log on to the computer on which you want to install Exchange 2013.
2. Navigate to the network location of the Exchange 2013 installation files.
3. Start Exchange 2013 Setup by double-clicking `Setup.exe`

Important:

If you have User Access Control (UAC) enabled, you must right-click `Setup.exe` and select **Run as administrator**.

4. On the **Check for Updates?** page, choose whether you want Setup to connect to the Internet and download product and security updates for Exchange 2013. If you select **Connect to the Internet and check for updates**, Setup will download updates and apply them prior to continuing. If you select **Don't check for updates right now**, you can download and install updates manually later. We recommend that you download and install updates now. Click **Next** to continue.
5. The **Introduction** page begins the process of installing Exchange into your organization. It will guide you through the installation. Several links to helpful deployment content are listed. We recommend that you visit these links prior to continuing setup. Click **Next** to continue.
6. On the **License Agreement** page, review the software license terms. If you agree to the terms, select **I accept the terms in the license agreement**, and then click **Next**.
7. On the **Recommended settings** page, select whether you want to use the recommended settings. If you select **Use recommended settings**, Exchange will automatically send error reports and information about your computer hardware and how you use Exchange to Microsoft. If you select **Don't use recommended settings**, these settings remain disabled but you can enable them at any time after Setup completes. For more information about these settings and how information sent to Microsoft is used, click **?**.
8. On the **Server Role Selection** page, select **Edge Transport**. Remember that you can't add the Mailbox or Client Access server roles to a computer that has the Edge Transport role installed. The management tools are installed automatically if you install any server role.

Select **Automatically install Windows Server roles and features that are required to install Exchange Server** to have the Setup wizard install required Windows prerequisites. You may need to reboot the computer to complete the installation of some Windows features. If you don't select this option, you must install the Windows features manually.

Note:

This option installs only the Windows features required by Exchange. You must manually install other prerequisites manually. For more information, see Exchange 2013 prerequisites.

Click **Next** to continue.

9. On the **Installation Space and Location** page, either accept the default installation location or click **Browse** to choose a new location. Make sure that you have enough disk space available in the location where you want to install Exchange. Click **Next** to continue.
10. On the **Readiness Checks** page, view the status to determine if the organization and server role prerequisite checks completed successfully. If they haven't completed successfully, you must resolve any reported errors before you can install Exchange 2013. You don't need to exit Setup when resolving some of the prerequisite errors. After resolving a reported error, click **back** and then click **Next** to run the prerequisite check again. Be sure to also review any warnings that are reported. If all readiness checks have completed successfully, click **Next** to install Exchange 2013.
11. On the **Completion** page, click **Finish**.
12. Restart the computer after Exchange 2013 has completed.
13. Complete your deployment by performing the tasks provided in Exchange 2013 post-Installation tasks.

How do you know this worked?

To verify that you've successfully installed Exchange 2013, see Verify an Exchange 2013 installation.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Delegate the installation of an Exchange 2013 server

Exchange Server 2013 > Planning and deployment > Deploy a new installation of Exchange 2013 >

Applies to: *Exchange Server 2013, Exchange Server, Exchange Online*

Topic Last Modified: *2014-07-31*

Exchange Server 2013 lets you delegate the installation of Exchange servers to people who aren't members of the Exchange 2013 Organization Management role group. This is often helpful in large companies where the people who install and set up servers aren't the same people who manage services, like Exchange. If this sounds like something you want to do, this topic is for you.

Normally, when Exchange is installed, the people installing it need to be members of the Organization Management role group. This is because when Exchange is installed, changes are made to Active Directory, and only Exchange administrators, who are members of the Organization Management role group, can make those changes. The following list shows the changes that are

made:

- A server object is created in the **CN=Servers,CN=Exchange Administrative Group (FYDIBOHF23SPDLT),CN=Administrative Groups,CN=<Organization Name>,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=<Root Domain>** configuration partition.
- The following access control entries (ACEs) are added to the server object within the configuration partition for the Delegated Setup role group:
  - Full Control on the server object and its child objects
  - Deny access control entry for the Send As extended right
  - Deny access control entry for the Receive As extended right
  - Deny CreateChild and DeleteChild permissions for Exchange Public Folder Store objects

Note:

Public folders are administered at an organizational level; therefore, the creation and deletion of public folder stores is restricted to Exchange administrators.

- The Active Directory computer account for the server is added to the Exchange Servers group.
- The server is added as a provisioned server in the Exchange Admin Center.

In large companies, the people who install and set up new servers often aren't Exchange administrators. To enable them to install Exchange, an Exchange administrator can *provision* the server in Active Directory. When a server is provisioned, all of the changes needed for the new Exchange server to function are made to Active Directory separately from the actual installation of Exchange on a computer. An Exchange administrator can provision a new server in Active Directory hours or even days before Exchange is installed on the new computer. After a server has been provisioned, the person doing the installation needs only to be a member of the Delegated Setup role group to install Exchange. The Delegated Setup role group only allows members to install provisioned servers.

Keep the following in mind when thinking about using delegated setup:

- At least one Exchange 2013 server has to already be installed before you can delegate the installation of additional servers. The person who installs the first server needs to be an Exchange administrator. For more information, check out Checklist: Perform a new installation of Exchange 2013.
- A delegated user can't uninstall an Exchange server. To uninstall an Exchange server, you need to be an Exchange administrator.

How do I provision an Exchange 2013 server?

To provision a server for Exchange, you need to use Exchange 2013 command-line Setup. If you're not very familiar with the Windows Command Prompt, don't worry. This topic steps you through exactly what you need to do. Before we start, here are a couple of things to keep in mind:

- You need to be a member of the Organization Management role group to provision a server.
- You should have the latest release of Exchange 2013. You can get the download link from Updates for Exchange 2013.

The command that you need to use to provision the server depends on whether you're running

Setup from the computer you're provisioning or whether you're running it from another computer. Choose the command in the following steps that matches where you're running Setup:

1. Press the Windows key + 'R' to open the **Run** window.
2. In **Open**, type**cmd.exe**, and then press Enter to open a **Windows Command Prompt**.
3. Change directories to where you downloaded and expanded the Exchange 2013 install files. If the install files are located in `C:\Downloads\Exchange 2013`, use the following command.

```
CD "C:\Downloads\Exchange 2013"
```

4. Choose the command that matches where you're running Setup:
   - **If you're running Setup on the computer that's being provisioned**, run the following command:

```
Setup.exe /NewProvisionedServer /
IAcceptExchangeServerLicenseTerms
```

   - **If you're running Setup on another computer**, run the following command:

```
Setup.exe /NewProvisionedServer:<ComputerName> /
IAcceptExchangeServerLicenseTerms
```

5. After you provision the server, you need to make sure that you've added the users who should be able to install Exchange on provisioned servers to the Delegated Setup role group. To see how to add users to a role group, see Manage Role Group Members.

When you're done with these steps, the computer will be ready for Exchange to be installed. Exchange 2013 can be installed on a provisioned server by using the steps in Install Exchange 2013 using the Setup wizard.

How do I know this worked?

To make sure the server was properly provisioned for Exchange, you can do the following:

1. Go to **Start** > **Administrative Tools**, and then open **Active Directory Users and Computers**.
2. Select **Microsoft Exchange Security Groups**, double-click **Exchange Servers**, and then select the **Members** tab.
3. On the **Members** tab, check to see if the server you just provisioned is listed as a member of the security group.

If your server is listed as a member of the Exchange Servers security group, it was properly provisioned. Someone who's a member of the Delegated Setup role group can now install Exchange on that server.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you’re looking for? Please take a minute to send us feedback about the

information you were hoping to find.

Upgrade Exchange 2013 to the latest cumulative update or service pack

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013, Exchange Server, Exchange Online*

Topic Last Modified: *2014-06-12*

If you have Microsoft Exchange Server 2013 installed, you can upgrade it to the latest Exchange 2013 cumulative update or service pack. You can use the Exchange 2013 Setup wizard to upgrade your current version of Exchange 2013. For more information about the latest Exchange 2013 cumulative update or service pack, see Updates for Exchange 2013. To learn more about Exchange 2013, see What's new in Exchange 2013.

Warning:

After you upgrade Exchange 2013 to a newer cumulative update or service pack, you can't uninstall the new version to revert to the previous version. If you uninstall the new version, you remove Exchange from the server.

What do you need to know before you begin?

- Estimated time to complete: 60 minutes
- Make sure you read the release notes before you install Exchange 2013. For more information, see Release notes for Exchange 2013.
- Make sure that any server on which you plan to install the cumulative update or service pack meets the system requirements and prerequisites. For more information, see Exchange 2013 system requirements and Exchange 2013 prerequisites.

Warning:

Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.

- After you install a cumulative update or service pack, you must restart the computer so that changes can be made to the registry and operating system.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Install the Exchange 2013 cumulative update or service pack

You can install the cumulative update or service pack for Exchange 2013 by using either the Setup wizard or via unattended mode. For specific instructions, see the following topics:

- Install Exchange 2013 using the Setup wizard
- Install Exchange 2013 using unattended mode

Upgrade from Exchange 2010 to Exchange 2013

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013, Exchange Server, Exchange Online*

Topic Last Modified: *2014-06-19*

Microsoft Exchange Server 2010 and Exchange Server 2007 have multiple server roles: Client Access, Mailbox, Hub Transport, Unified Messaging, and Edge Transport. With Exchange Server 2013, we reduced the number of server roles from five to three: Client Access, Mailbox, and Edge Transport. Unified Messaging is now considered a component or sub-feature of the voice-related features that are offered in Exchange 2013. (For more details about the changes, see “Exchange 2013 architecture” in What's new in Exchange 2013.)

When you're upgrading your existing Exchange 2010 organization to Exchange 2013, there's a period of time when Exchange 2010 and Exchange 2013 servers will coexist within your organization. You can maintain this mode for an indefinite period of time, or you can immediately complete the upgrade to Exchange 2013 by moving all resources from Exchange 2010 to Exchange 2013, and then decommissioning the Exchange 2010 servers. You have a coexistence scenario if the following conditions are true:

- Exchange 2013 is deployed in an existing Exchange organization.
- More than one version of Microsoft Exchange provides messaging services to the organization.

You can't upgrade an existing Exchange 2003 organization directly to Exchange 2013. You must first upgrade the Exchange 2003 organization to either an Exchange 2007 or Exchange 2010 organization, and then you can upgrade the Exchange 2007 or Exchange 2010 organization to Exchange 2013. We recommend that you upgrade your organization from Exchange 2003 to Exchange 2010, and then upgrade from Exchange 2010 to Exchange 2013.

⚠Warning:

You need to remove all instances of Exchange 2003 from your organization before you can upgrade to Exchange 2013.

You can migrate all your Exchange 2003 mailboxes to Exchange Online. For more information about this approach, see **Mailbox Migration to Exchange Online**.

The following table lists the scenarios in which coexistence between Exchange 2013 and earlier versions of Exchange is supported.

Coexistence of Exchange 2013 and earlier versions of Exchange Server

Exchange version	Exchange organization coexistence
Exchange Server 2003 and earlier versions	Not supported
Exchange 2007	Supported with the following minimum versions of Exchange: • [1]Update Rollup 10 for Exchange 2007 Service Pack 3 (SP3) on all Exchange 2007 servers in the organization, including Edge Transport servers. • Exchange 2013 Cumulative Update 2 (CU2) or later on all Exchange 2013 servers in the organization.
Exchange 2010	Supported with the following minimum versions of Exchange: • [2]Exchange 2010 SP3 on all Exchange 2010 servers in the organization, including Edge Transport servers. • Exchange 2013 CU2 or later on all Exchange 2013 servers in the organization.
Mixed Exchange 2010 and Exchange 2007 organization	Supported with the following minimum versions of Exchange: • [1]Update Rollup 10 for Exchange 2007 SP3 on all Exchange 2007 servers in the organization, including Edge Transport servers. • [2]Exchange 2010 SP3 on all Exchange 2010 servers in the organization, including Edge

	Transport servers. • Exchange 2013 CU2 or later on all Exchange 2013 servers in the organization.

[1] If you want to create an EdgeSync Subscription between an Exchange 2007 Hub Transport server and an Exchange 2013 SP1 Edge Transport server, you need to install Exchange 2007 SP3 Update Rollup 13 or later on the Exchange 2007 Hub Transport server.

[2] If you want to create an EdgeSync Subscription between an Exchange 2010 Hub Transport server and an Exchange 2013 SP1 Edge Transport server, you need to install Exchange 2010 SP3 Update Rollup 5 or later on the Exchange 2010 Hub Transport server.

Mixed mode coexistence of Exchange 2013 and Exchange 2007 with Exchange 2010

If you have Active Directory sites with both Exchange 2010and Exchange 2007 installed, follow the upgrade instructions from both Exchange 2010 and Exchange 2007, and perform the upgrade steps required by both.

Overview of the upgrade process

To help you get an overview of the Exchange 2010 to Exchange 2013 upgrade process, we've gathered resources related to each key task in the following table. For specific step-by-step guidance, see Checklist: Upgrade from Exchange 2010.

Task	Topic
Learn about Exchange 2013 roles and components	What's new in Exchange 2013 Client Access server Mailbox server Mail flow Unified Messaging
Install Exchange 2013	Install Exchange 2013 using the Setup wizard Install the Exchange 2013 Edge Transport role using the Setup wizard (optional) Verify an Exchange 2013 installation

Add digital certificates on the Client Access server	Exchange 2013 Client Access server configuration Digital certificates and SSL Create a digital certificate request
Configure Exchange-related virtual directories	Default settings for Exchange virtual directories
Move mailboxes from Exchange 2010	Mailbox moves in Exchange 2013
Configure transport components	Edge Subscriptions (only necessary if you've installed an Edge Transport server) Mail routing Shadow redundancy Delivery reports for administrators
Configure and deploy UM	Planning for Unified Messaging Deploying voice mail and UM

Checklist: Upgrade from Exchange 2010

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-28*

Use this checklist to upgrade from Microsoft Exchange 2010 to Exchange 2013. Before you start working with this checklist, make sure you're familiar with the concepts discussed in:

- Planning and deployment
- What's new in Exchange 2013

This checklist is generic in that it provides guidance for a typical upgrade scenario.

Note:

The Exchange Server Deployment Assistant provides you with customized step-by-step guidance about how to deploy Exchange Server. The Deployment Assistant can help you deploy a new installation of Exchange Server 2013, upgrade a previous version to Exchange 2013, or configure a hybrid deployment of Exchange 2013 and Exchange Online. To learn

more, see Exchange Server Deployment Assistant.

Checklist for upgrading from Exchange 2010 to Exchange 2013

Done?		Task	Topic(s)
	1. Read the release notes.	Release notes for Exchange 2013	
		2. Verify system requirements	Exchange 2013 system requirements
		3. Confirm prerequisite steps are done	Exchange 2013 prerequisites Deployment security checklist
		4. Configure disjoint namespace **Note:** This step is optional. It's only necessary if your organization is running a disjoint namespace.	Configure the DNS suffix search list for a disjoint namespace
	5. Select an offline address book for all Exchange 2010 mailbox databases	Set mailbox database properties in Manage mailbox databases in Exchange 2013	
		6. Install Exchange 2013	Install Exchange 2013 using the Setup wizard Install the Exchange 2013 Edge Transport role using the Setup wizard

			Verify an Exchange 2013 installation
	7. Create an Exchange 2013 mailbox	Create user mailboxes	
		8. Configure Exchange-related virtual directories **Note:** This step is necessary if you want to use Exchange Web Services, Outlook Anywhere, or the offline address book. It also may be required if you need to change any of the default settings for Exchange Control Panel, Microsoft Office Outlook Web App, or Exchange ActiveSync.	Exchange 2013 Client Access server configuration
		9. Add digital certificates on the Client Access server	Digital certificates and SSL
	10. Move arbitration mailbox	Move the Exchange 2010 system mailbox to Exchange 2013	
		11. Configure Unified Messaging **Note:** This step is optional. It's only necessary if you want to use Unified Messaging in your organization.	Upgrade Exchange 2010 UM to Exchange 2013 UM
		12. Configure Edge Transport server **Note:** This step is optional. It's only	Configure Internet mail flow through a subscribed Edge Transport server

		necessary if your organization is uses an Edge Transport server.		
	13. Enable and configure Outlook Anywhere	Outlook Anywhere		
	14. Configure service connection point	Exchange 2013 Client Access server configuration		
	15. Configure DNS records	**Configure DNS records for Exchange 2010 multiple-server install**		
		16. Move mailboxes from Exchange 2010 to Exchange 2013	Mailbox moves in Exchange 2013	
		17. Move public folder data from Exchange 2013 to Exchange 2013	Public folders Migrate public folders to Exchange 2013 from previous versions	
		18. Post-installation tasks	Exchange 2013 post-Installation tasks	

Upgrade from Exchange 2007 to Exchange 2013

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013, Exchange Server, Exchange Online*

Topic Last Modified: *2014-06-19*

Microsoft Exchange Server 2010 and Exchange Server 2007 have multiple server roles: Client Access, Mailbox, Hub Transport, Unified Messaging, and Edge Transport. With Exchange Server 2013, we reduced the number of server roles from five to three: Client Access, Mailbox, and Edge Transport. Unified Messaging is now considered a component or sub feature of the voice-related features that are offered in Exchange 2013. (For more details about the changes, see "Exchange 2013 architecture" in What's new in Exchange 2013.)

When you're upgrading your existing Exchange 2007 organization to Exchange 2013, there's a period of time when Exchange 2007 and Exchange 2013 servers will coexist within your organization. You can maintain this mode for an indefinite period of time, or you can immediately complete the upgrade to Exchange 2013 by moving all resources from Exchange 2007 to Exchange 2013, and then decommissioning the Exchange 2007 servers. You have a coexistence scenario if the following conditions are true:

- Exchange 2013 is deployed in an existing Exchange organization.
- More than one version of Microsoft Exchange provides messaging services to the organization.

You can't upgrade an existing Exchange 2003 organization directly to Exchange 2013. You must first upgrade the Exchange 2003 organization to either an Exchange 2007 or Exchange 2010 organization, and then you can upgrade the Exchange 2007 or Exchange 2010 organization to Exchange 2013. We recommend that you upgrade your organization from Exchange 2003 to Exchange 2010, and then upgrade from Exchange 2010 to Exchange 2013.

Warning:
You need to remove all instances of Exchange 2003 from your organization before you can upgrade to Exchange 2013.

You can migrate all your Exchange 2003 mailboxes to Exchange Online. For more information about this approach, see **Mailbox Migration to Exchange Online**.

The following table lists the scenarios in which coexistence between Exchange 2013 and earlier versions of Exchange is supported.

Coexistence of Exchange 2013 and earlier versions of Exchange Server

Exchange version	Exchange organization coexistence
Exchange Server 2003 and earlier versions	Not supported
Exchange 2007	Supported with the following minimum versions of Exchange: • [1]Update Rollup 10 for Exchange 2007 Service Pack 3 (SP3) on all Exchange 2007 servers in the organization, including Edge Transport servers.

	• Exchange 2013 Cumulative Update 2 (CU2) or later on all Exchange 2013 servers in the organization.
Exchange 2010	Supported with the following minimum versions of Exchange: • [2]Exchange 2010 SP3 on all Exchange 2010 servers in the organization, including Edge Transport servers. • Exchange 2013 CU2 or later on all Exchange 2013 servers in the organization.
Mixed Exchange 2010 and Exchange 2007 organization	Supported with the following minimum versions of Exchange: • [1]Update Rollup 10 for Exchange 2007 SP3 on all Exchange 2007 servers in the organization, including Edge Transport servers. • [2]Exchange 2010 SP3 on all Exchange 2010 servers in the organization, including Edge Transport servers. • Exchange 2013 CU2 or later on all Exchange 2013 servers in the organization.

[1] If you want to create an EdgeSync Subscription between an Exchange 2007 Hub Transport server and an Exchange 2013 SP1 Edge Transport server, you need to install Exchange 2007 SP3 Update Rollup 13 or later on the Exchange 2007 Hub Transport server.

[2] If you want to create an EdgeSync Subscription between an Exchange 2010 Hub Transport server and an Exchange 2013 SP1 Edge Transport server, you need to install Exchange 2010 SP3 Update Rollup 5 or later on the Exchange 2010 Hub Transport server.

Mixed mode coexistence of Exchange 2013 and Exchange 2007 with Exchange 2010

If you have Active Directory sites with both Exchange 2010 and Exchange 2007 installed, follow the upgrade instructions from both Exchange 2010 and Exchange 2007, and perform the upgrade steps required by both.

Overview of the upgrade process

To help you get an overview of the Exchange 2007 to Exchange 2013 upgrade process, we've gathered resources related to each key task in the following table. For specific step-by-step guidance, see Checklist: Upgrade from Exchange 2007.

Task	Topic
Learn about Exchange 2013 roles and components	What's new in Exchange 2013 Client Access server Mailbox server Mail flow Unified Messaging
Install Exchange 2013	Install Exchange 2013 using the Setup wizard Install the Exchange 2013 Edge Transport role using the Setup wizard (optional) Verify an Exchange 2013 installation
Add digital certificates on the Client Access server	Exchange 2013 Client Access server configuration Digital certificates and SSL Create a digital certificate request
Configure Exchange-related virtual directories	Default settings for Exchange virtual directories
Move mailboxes from Exchange 2010	Mailbox moves in Exchange 2013
Configure transport components	Edge Subscriptions (only necessary if you've installed an Edge Transport server) Mail routing Shadow redundancy Delivery reports for administrators
Configure and deploy UM	Planning for Unified Messaging

	Deploying voice mail and UM

Checklist: Upgrade from Exchange 2007

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-28*

Use this checklist to upgrade from Microsoft Exchange Server 2007 to Exchange Server 2013. Before you start working with this checklist, make sure you're familiar with the concepts discussed in:

- Planning and deployment
- What's new in Exchange 2013

This checklist is generic in that it provides guidance for a typical upgrade scenario.

Note:

The Exchange Server Deployment Assistant provides you with customized step-by-step guidance about how to deploy Exchange Server. The Deployment Assistant can help you deploy a new installation of Exchange Server 2013, upgrade a previous version to Exchange 2013, or configure a hybrid deployment of Exchange 2013 and Exchange Online. To learn more, see Exchange Server Deployment Assistant.

Checklist for upgrading from Exchange 2007 to Exchange 2013

Done?		Task	Topic(s)	
	1. Read the release notes.	Release notes for Exchange 2013		
2. Verify system requirements		Exchange 2013 system requirements		
3. Confirm prerequisite steps are done		Exchange 2013 prerequisites Deployment security checklist		
4. Configure disjoint namespace		Configure the DNS suffix		

Note: This step is optional. It's only necessary if your organization is running a disjoint namespace.		search list for a disjoint namespace		
	5. Select an offline address book for all Exchange 2007 mailbox databases	How to Provision Recipients for Offline Address Book Downloads		
	6. Create legacy Exchange hostname	**Create a legacy Exchange host name**		
7. Install Exchange 2013		Install Exchange 2013 using the Setup wizard Install the Exchange 2013 Edge Transport role using the Setup wizard Verify an Exchange 2013 installation		
	8. Create an Exchange 2013 mailbox	Create user mailboxes		
	9. Configure Exchange-related virtual directories **Note:** This step is necessary if you want to use Exchange Web	Exchange 2013 Client Access server configuration		

	Services, Outlook Anywhere, or the offline address book. It also may be required if you need to change any of the default settings for Exchange Control Panel, Microsoft Office Outlook Web App, or Exchange ActiveSync.			
10. Configure Exchange 2013 certificates		Digital certificates and SSL		
	11. Configure Exchange 2007 certificates	Managing SSL for a Client Access Server		
	12. Configure Edge Transport server **Note:** This step is optional. It's only necessary if your organization is uses an Edge Transport server.	Configure Internet mail flow through a subscribed Edge Transport server		
13. Configure Unified Messaging **Note:** This step is optional. It's only necessary if you want to use		Planning for Unified Messaging Deploy Exchange 2013 UM		

Unified Messaging in your organization.				
	14. Enable and configure Outlook Anywhere	Outlook Anywhere		
	15. Configure service connection point	Exchange 2013 Client Access server configuration		
	16. Configure Exchange 2007 URLs	**Configure Exchange 2007 external URLs**		
	17. Configure DNS records	**Configure DNS records for Exchange 2007 multiple-server install**		
18. Move mailboxes from Exchange 2007 to Exchange 2013		Mailbox moves in Exchange 2013		
19. Move public folder data from Exchange 2013 to Exchange 2013 **Note:** This step is optional. It's only necessary if your organization is uses public folders.		Public folders Migrate public folders to Exchange 2013 from previous versions		
20. Post-installation tasks		Exchange 2013 post-Installation tasks		

Deploy multiple forest topologies for Exchange 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-20*

This topic provides an overview of deploying Microsoft Exchange Server 2013 in multiple forest topologies. You'll find information about the following subjects:

- **Supported multiple forest topologies** Exchange 2013 supports two types of multiple forest topologies: cross-forest and resource forest.
- **GAL synchronization** If you have a cross-forest environment, you need to ensure that the GAL in any given forest contains mail recipients from other forests.
- **Moving mailboxes across forests** The **New-MoveRequest** and **New-MigrationBatch** cmdlets in the Exchange Management Shell can help move mailboxes from one forest to another.
- **Understanding multiple forest administration** Learn about the permissions model to configure and manage the permissions between your forests.

Supported multiple forest topologies

Exchange 2013 supports the following types of multiple forest topologies:

- **Cross-forest** A cross-forest topology is one with multiple Exchange forests. Here is an overview of what you need to do to deploy Exchange 2013 in a topology with a multiple forest:
  1. You must first install Exchange 2013 in each forest. For more information, see Deploy a new installation of Exchange 2013.
  2. Next, you must synchronize the recipients in each of the forests, so that the Global Address List (GAL) in each forest contains users from all the synchronized forests. See the "GAL Synchronization" section below for more details.
  3. Finally, you must configure the Availability service so that users in one forest can view availability data for users in another forest. For more information, see Configure the Availability service for cross-forest topologies.

For details about deploying Exchange 2013 in a cross-forest topology, see Deploy Exchange 2013 in a cross-forest topology.

- **Resource forest** A resource forest topology is one with an Exchange forest and one or more user accounts forests. Here is an overview of what you need to do to deploy Exchange 2013 in a topology with a resource forest:
  1. You must have a forest with Exchange installed. In the Exchange forest, you must have disabled the user accounts that have Exchange mailboxes.

2. You must have at least one forest that contains user accounts. This forest should *not* have Exchange installed.
3. Then, you must associate the disabled user accounts in the Exchange forest with the user accounts in the accounts forest.

For details about deploying Exchange 2013 in a resource forest topology, see Deploy Exchange 2013 in an Exchange resource forest topology.

GAL synchronization

By default, a GAL contains mail recipients from a single forest. If you have a cross-forest environment, we recommend using Microsoft Identity Lifecycle Manager (ILM) 2007 Feature Pack 1 (FP1) to ensure that the GAL in any given forest contains mail recipients from other forests. ILM 2007 FP1 creates mail users that represent recipients from other forests, thereby allowing users to view them in the GAL and send mail. For example, users in Forest A appear as a mail user in Forest B and vice versa. Users in the target forest can then select the mail user object that represents a recipient in another forest to send mail.

To enable GAL synchronization, you create management agents that import mail-enabled users, contacts, and groups from designated Active Directory services into a centralized metadirectory. In the metadirectory, mail-enabled objects are represented as mail users. Groups are represented as contacts without any associated membership. The management agents then export these mail users to an organizational unit in the specified target forest.

For more information about Forefront Identity Manager (FIM), see Forefront Identity Manager 2010.

Moving mailboxes across forests

In a cross-forest topology, you may want to move mailboxes from one forest to another. To do this you must use the **New-MoveRequest** or **New-MigrationBatch** cmdlet in the Exchange Management Shell. For more information about moving mailboxes across forests, see the following topics:

- Prepare mailboxes for cross-forest move requests
- Prepare mailboxes for cross-forest moves using the Prepare-MoveRequest.ps1 script in the Shell
- Prepare mailboxes for cross-forest moves using sample code

Understanding multiple forest administration

Exchange 2013 uses new permissions functionality to manage your multiple forest environments.

Exchange 2013 uses a Role Based Access Control (RBAC) permissions model. The management role groups that administrators are members of, and the management role assignment policies that end-users are assigned, determine what each administrator and end-user can do. To understand multiple forest permissions, you need to be familiar with RBAC. For more information about RBAC and role groups and role assignment policies in particular, see Understanding Role Based Access

Control.

You can use the RBAC permissions model to configure and manage the permissions between your forests. For more information about multiple forest permissions, see the following topics:

- Understanding multiple-forest permissions
- Manage linked role groups
- Create linked role groups that mirror built-in role groups

Deploy Exchange 2013 in a cross-forest topology

Exchange Server 2013 > Planning and deployment > Deploy multiple forest topologies for Exchange 2013 >

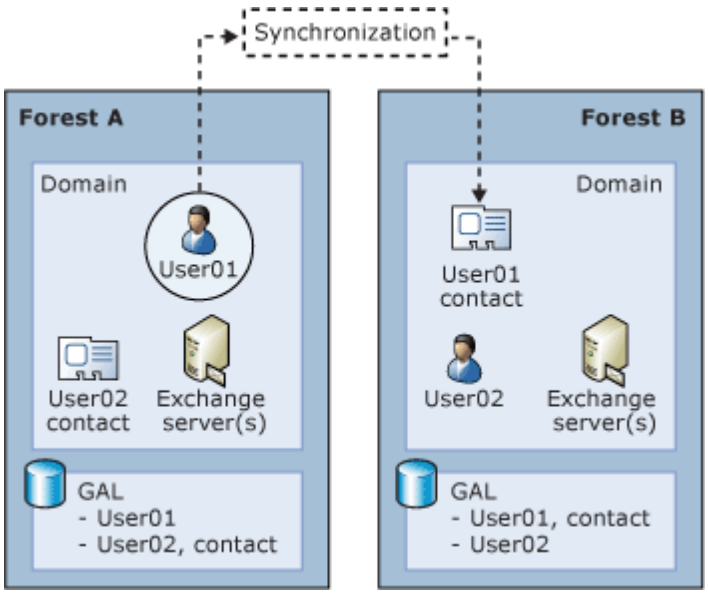
Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-13*

This topic explains how to deploy Exchange 2013 in a cross-forest topology using Microsoft Forefront Identity Manager 2010 R2 SP1. To deploy Exchange 2013 in a cross-forest topology, you must first install Exchange 2013 in each forest, and then connect the forests so that users can see address and availability data across the forests.

The following figure illustrates user synchronization between two Exchange 2013 forests.

Example of Exchange 2013 cross-forest synchronization



This topic does *not* describe how to deploy Exchange 2013 in a dedicated Exchange forest (or resource forest) topology. For more information about how to deploy Exchange 2013 in a resource forest topology, see Deploy Exchange 2013 in an Exchange resource forest topology.

What do you need to know before you begin?

To perform the following procedure in Exchange 2013, confirm the following:

- You have correctly configured Domain Name System (DNS) for name resolution across forests in your organization. To verify that DNS is configured correctly, use the Ping tool to test connectivity to each forest from the other forests in your organization and from the server on which you will run the GALSync agent.
- The GALSync management agent (MA) communicates with the Exchange 2013 forest using Windows PowerShell V2.0 RTM. Make sure Windows PowerShell v1.0 isn't installed on this computer by going to Control Panel, and then clicking Programs and Features.
- Ensure that Windows Remote Management has not been installed by Windows Update.
- Install Windows PowerShell and Windows Remote Management. For details, see Microsoft Knowledge Base article 968930, Windows Management Framework Core package (Windows PowerShell 2.0 and WinRM 2.0).
- Download Forefront Identity Manager 2010 R2 SP1. See Download of Microsoft Forefront Identity Manager 2010 R2 SP1.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Deploy Exchange 2013 in a cross-forest topology with Forefront Identity Manager 2010 R2 SP1

1. In each forest, install Exchange 2013 separately. To install Exchange 2013, perform the same steps that you would if you were installing Exchange 2013 in a single forest topology. For detailed steps, see one of the following topics:
   - Deploy a new installation of Exchange 2013
   - Install Exchange 2013 using the Setup wizard

Note:

This topic assumes that you don't have an existing Exchange 2007 or Exchange 2010 topology. If you do have an existing Exchange topology and you want to upgrade, see Upgrade from Exchange 2010 to Exchange 2013 or Upgrade from Exchange 2007 to Exchange 2013.

2. In each forest, use Active Directory Users and Computers to create a container in which FIM 2010 R2 SP1 will create contacts for each mailbox from the other forest. We recommend that you name this container **FromFIM**. To create the container, select the domain in which you want to create the container, right-click the domain, select **New** > **Organizational Unit**. In **New Object - Organizational Unit**, type **FromFIM**, and then click **OK**.
3. Create a GALSync management agent for each forest by using Forefront Identify Manager. This

allows you to synchronize the users in each forest and create a common GAL. For detailed steps, see the following resources:

- Configuring Global Address List (GAL) Synchronization with Forefront Identity Manager (FIM) 2010
- Work with Management Agents
- Forefront Identity Manager 2010 R2 Documentation Roadmap

Important:

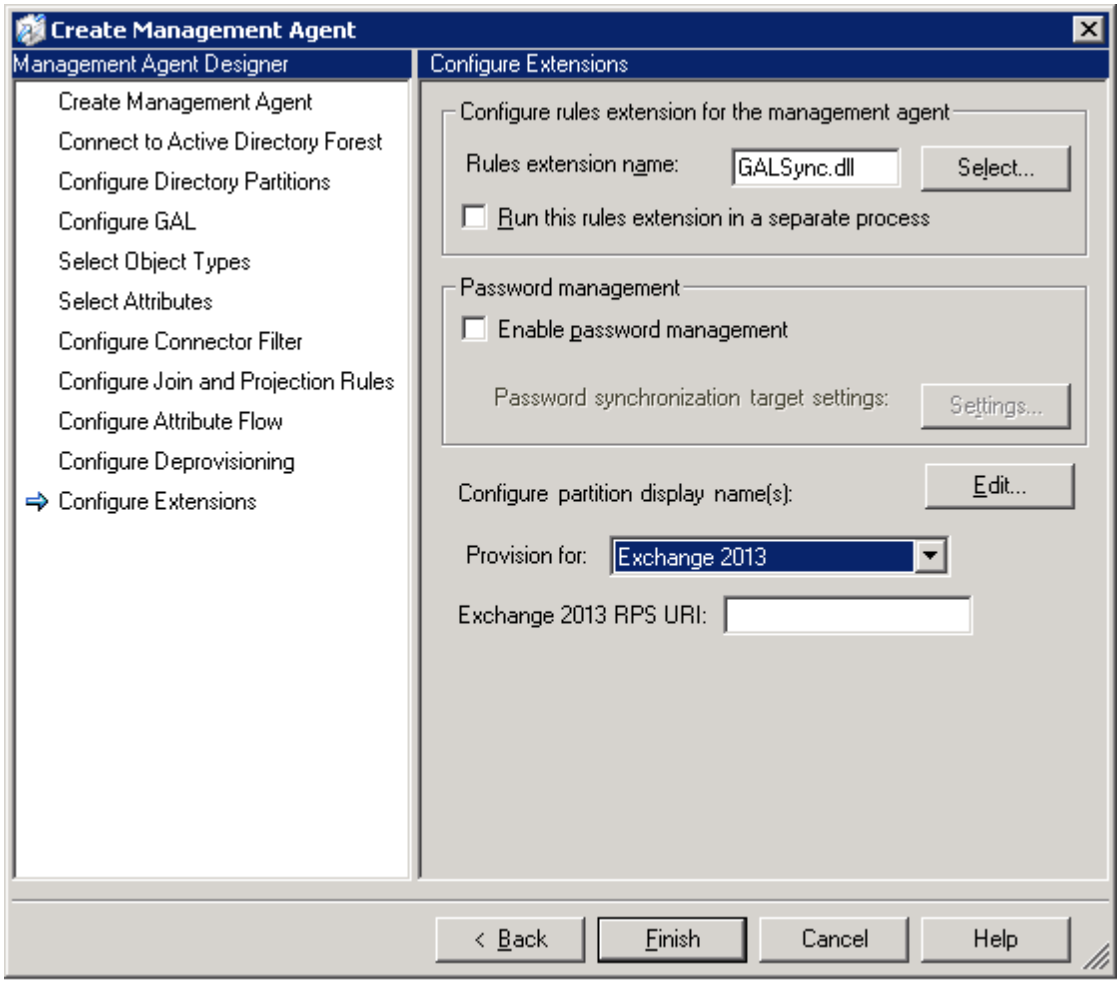
While the resources discuss Exchange 2010, Exchange 2013 is supported for FIM 2010 R2 SP1. Make sure that you configure **Extensions** in FIM 2010 R2 SP1 for Exchange 2013.

- On the **Configure Extensions** page, under **Configure partition display name(s)**, next to **Provision for**, select **Exchange 2013**. You will see the **Exchange 2013 RPS URI** field. Enter the URI of an Exchange 2013 Client Access server to make sure the remote PowerShell connection is functioning. The **Exchange 2013 RPS URI** should be in the following format: http://CAS_Server_FQDN/Powershell. Click **OK**.

Note:

Make sure that the administrator credentials used to connect to the Exchange 2013 forest can also make remote PowerShell connections to that forest.
The following figure shows how to select provisioning for Exchange 2013.

Provision GalSync Management Agent for Exchange 2013



4. Create an SMTP Send connector in each of the forests. For detailed steps, see Configure a cross-forest Send connector.

5. In each forest, enable the Availability service so that users in each forest can view free/busy data about users in the other forest. For more information, see Availability service in Exchange 2013.
6. If you want mail relayed through any forest in your organization, you must configure a domain in that forest as an authoritative domain. For detailed steps, see Configure Exchange to accept mail for multiple authoritative domains.

Deploy Exchange 2013 in an Exchange resource forest topology

Exchange Server 2013 > Planning and deployment > Deploy multiple forest topologies for Exchange 2013 >

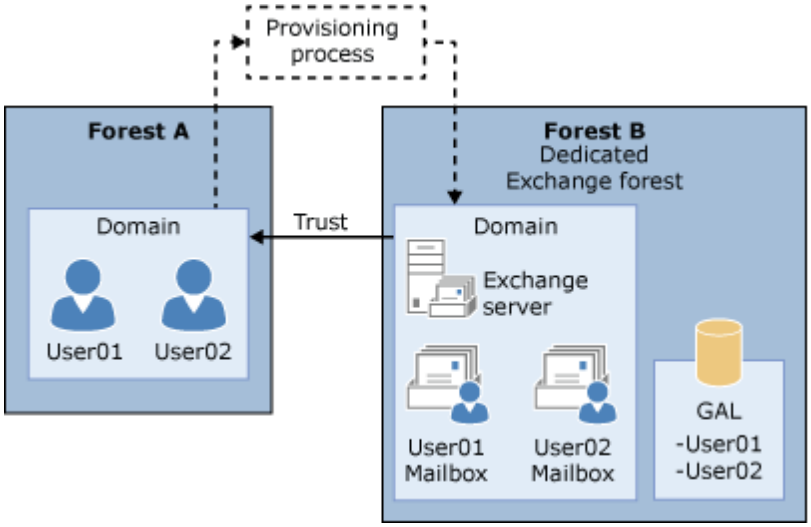
Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-05-08*

This topic explains how to deploy Microsoft Exchange 2013 in an Exchange resource forest topology. An Exchange resource forest is also called a dedicated Exchange forest. This topic assumes that you don't have an existing Exchange 2013 topology.

The following figure shows an Exchange organization with a resource forest.

Example of an Exchange organization with an Exchange resource forest



What do you need to know before you begin?

To perform the following procedure in Exchange 2013, confirm you have the following:

- You have the following two Active Directory forests:
  - One forest contains the user accounts for your organization. In this procedure, this forest is called the *accounts forest*.
  - One forest does not contain user accounts and does not yet have Exchange installed. In this

procedure, this forest is called the *Exchange forest*. You will use the procedure to install Exchange 2013 in this forest.

- You have correctly configured Domain Name System (DNS) for name resolution across forests in your organization. To check that you have DNS configured correctly, ping each forest from the other forest or forests in your organization. For more information about configuring DNS, see the DNS Servers Operations Guide.

Deploy Exchange 2013 in an Exchange resource forest topology

1. From a domain controller in the Exchange forest, create a one-way outgoing trust so that the Exchange forest trusts the accounts forest. For detailed steps, see Create a one-way, outgoing, forest trust for both sides of the trust.

Note:
Although we recommend that you create a forest trust, you can create either a forest trust or an external trust. If you create an external trust, when you create linked mailboxes in Step 3, on the **Master Account** page of the New Mailbox wizard, you must specify a user account that can access the domain controller in the trusted forest. You can't use the credentials with which you are currently logged on. If you create linked mailboxes by using the **New-Mailbox** cmdlet, you must specify a user account that can access the domain controller in the trusted forest by using the *LinkedCredential* parameter.

2. In the Exchange forest, install Exchange 2013. Install Exchange the same way that you would in a single forest scenario. For detailed steps about how to install Exchange 2013, see one of the following topics:
   - Deploy a new installation of Exchange 2013
   - Install Exchange 2013 using the Setup wizard
3. In the Exchange forest, for each user in the accounts forest that will have a mailbox in the Exchange forest, create a mailbox that is associated with an external account. For detailed steps, see Manage linked mailboxes.

Exchange 2013 post-Installation tasks

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-30*

After you've completed the installation of Microsoft Exchange Server 2013, read the following topics to help you configure your new Exchange 2013 organization.

Topic	Description

Enter your Exchange 2013 product key	Read this topic to license an Microsoft Exchange server.
Configure mail flow and client access	Read this topic to configure mail flow to and from the Internet and configure Microsoft Exchange to accept client connections from the Internet.
Configure Internet mail flow through a subscribed Edge Transport server	Read this topic if you're installing an Edge Transport server and you want to configure an EdgeSync Subscription between that server and a Hub Transport server.
Verify an Exchange 2013 installation	Read this topic to verify that Exchange 2013 was installed successfully in your organization.
Install the Exchange 2013 management tools	Read this topic to install the Exchange Management Shell and Exchange Toolbox on client workstations or other non-Exchange servers in your organization.

If you want to configure additional features, such as permissions, compliance, high availability, and more, see Exchange Server 2013.

Enter your Exchange 2013 product key

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-14*

A product key tells Exchange Server 2013 that you've purchased a Standard or Enterprise Edition license. If the product key you purchased is for an Enterprise Edition license, it lets you mount more than five databases per server in addition to everything that's available with a Standard Edition license. If you want to read more about Exchange licensing, see Exchange 2013: editions and versions.

If you don't enter a product key, your server is automatically licensed as a trial edition. The trial edition functions just like an Exchange Standard Edition server and is helpful if you want to try out

Exchange before you buy it, or to run tests in a lab. The only difference is that you can only use an Exchange server licensed as a trial edition for up to 180 days. If you want to keep using the server beyond 180 days, you'll need to enter a product key or the Exchange Admin Center (EAC) will start to show reminders that you need to enter a product key to license the server.

Tip:

We've noticed some visitors to this page are looking for information on how to install or activate Office. If that's you, check out these pages:

- Install Office
- Need help with your Office product key?

If you want to enter a product key on an Exchange 2010 server, go to Enter an Exchange 2010 product key.
If you want to enter a product key on an Exchange 2013 server, you're in the right place! Read on.

What do you need to know before you begin?

- Estimated time to complete this procedure: less than 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Product key" entry in the Exchange and Shell infrastructure permissions topic.
- If you're licensing an Exchange server that's running the Mailbox server role, you'll need to restart the Microsoft Exchange Information Store service on the server after you enter the product key.
- If you want to upgrade an Exchange server from a Standard Edition license to an Enterprise Edition license, follow the steps in this topic.
- If you want to downgrade an Exchange server from an Enterprise Edition license to a Standard Edition license, you need to reinstall Exchange.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to enter the product key

1. Open the EAC by browsing to https://<*Client Access server name*>/ecp.
2. Enter your user name and password in **Domain\user name** and **Password**, and then click **Sign in**.
3. Go to **Servers** > **Servers**. Select the server you want to license, and then click **Edit** ✎.
4. (Optional) If you want to upgrade the server from a Standard Edition license to an Enterprise Edition license, on the **General** page, select **Change product key**. You'll only see this option if the server is already licensed.

5. On the **General** page, enter your product key in the **Enter a valid product key** text boxes.
6. Click **Save**.
7. If you licensed an Exchange server running the Mailbox server role, do the following to restart the Microsoft Exchange Information Store service:
   a. Open **Control Panel**, go to **Administrative Tools**, and then open **Services**.
   b. Right-click on **Microsoft Exchange Information Store** and click **Restart**.

Use the Shell to enter the product key

This example uses the **set-ExchangeServer** cmdlet to enter the product key.

Note:
You can run this command again on the same server to upgrade it from a Standard Edition license to an Enterprise Edition license.

```
Set-ExchangeServer ExServer01 -ProductKey aaaaa-aaaaa-aaaaa-aaaaa-aaaaa
```

For detailed syntax and parameter information, see Set-ExchangeServer.

If you licensed an Exchange server running the Mailbox server role, do the following to restart the Microsoft Exchange Information Store service:

1. Open **Control Panel**, go to **Administrative Tools**, and then open **Services**.
2. Right-click **Microsoft Exchange Information Store** and click **Restart**.

How do you know this worked?

To use the EAC to verify that you've successfully licensed the server as Standard Edition or Enterprise Edition, do the following:

1. Enter your user name and password in **Domain\user name** and **Password**, and then click **Sign in**.
2. Go to **Servers** > **Servers**.
3. Select the server you want to view, and then look in the server details pane. If the product key has been accepted, **Licensed** will appear along with the Exchange 2013 edition.

To use the Shell to verify that you've successfully licensed the server as Standard Edition or Enterprise Edition, do the following:

1. Open the Shell.
2. Run the following command to view the licensing status of a specific Exchange server.

```
Get-ExchangeServer ExServer01 | Format-Table Edition,*Trial*
```

3. (Optional) Run the following command to view the licensing status of all Exchange servers in your organization.

```
Get-ExchangeServer | Format-Table Name, Edition, *Trial* -
Auto
```

Configure mail flow and client access

Exchange Server 2013 > Planning and deployment > Exchange 2013 post-Installation tasks >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-20*

After you've installed Microsoft Exchange Server 2013 in your organization, you need to configure Exchange Server 2013 for mail flow and client access. Without these additional steps, you won't be able to send mail to the Internet and external clients such as Microsoft Office Outlook and Exchange ActiveSync devices won't be able to connect to your Exchange organization.

The steps in this topic assume a basic Exchange deployment with a single Active Directory site and a single simple mail transport protocol (SMTP) namespace.

Important:

This topic uses example values such as Ex2013CAS, contoso.com, mail.contoso.com, and 172.16.10.11. Replace the example values with the server names, FQDNs, and IP addresses for your organization.

For additional management tasks related to mail flow and clients and devices, see Mail flow and Clients and mobile.

What do you need to know before you begin?

- Estimated time to complete this task: 50 minutes
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- You might receive certificate warnings when you connect to the Exchange admin center (EAC) website until you configure a secure sockets layer (SSL) certificate on the Client Access server. You'll be shown how to do this later in this topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Important:

Each organization requires at a minimum one Client Access server and one Mailbox server in the Active Directory forest. Additionally, each Active Directory site that contains a Mailbox server must also contain at least one Client Access server. If you're separating your server roles, we recommend installing the Mailbox server role first.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Create a Send connector

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Send connectors" entry in the Mail flow permissions topic.

Before you can send mail to the Internet, you need to create a Send connector on the Mailbox server. Do the following.

1. Open the EAC by browsing to the URL of your Client Access server. For example, https://Ex2013CAS/ECP.
2. Enter your user name and password in **Domain\user name** and **Password** and then click **Sign in**.
3. Go to **Mail flow** > **Send connectors**. On the **Send connectors** page, click **New +**.
4. In the **New send connector** wizard, specify a name for the Send connector and then select **Internet**. Click **Next**.
5. Verify that **MX record associated with recipient domain** is selected. Click **Next**.
6. Under **Address space**, click **Add +**. In the **Add domain** window, make sure **SMTP** is selected in the **Type** field. In the **Fully Qualified Domain Name (FQDN)** field, enter ***. Click **Save**.
7. Make sure **Scoped send connector** isn't selected and then click **Next**.
8. Under **Source server**, click **Add +**. In the **Select a Server** window, select a Mailbox server. After you've selected the server, click **Add** and then click **OK**.
9. Click **Finish**.

Note:

A default inbound Receive connector is created when Exchange 2013 is installed. This Receive connector accepts anonymous SMTP connections from external servers. You don't need to do any additional configuration if this is the functionality you want. If you want to restrict inbound connections from external servers, modify the **Default Frontend <Client Access server>** Receive connector on the Client Access server.

How do you know this step worked?

To verify that you have successfully created an outbound Send connector, do the following:

1. In the EAC, verify the new Send connector appears in **Mail flow** > **Send connectors**.
2. Open Outlook Web App and send an email message to an external recipient. If the recipient receives the message, you've successfully configured the Send connector.

Step 2: Add additional accepted domains

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Accepted domains" entry in the Mail flow permissions topic.

By default, when you deploy a new Exchange 2013 organization in an Active Directory forest, Exchange uses the domain name of the Active Directory domain where Setup /PrepareAD was run. If you want recipients to receive and send messages to and from another domain, you must add the domain as an accepted domain. This domain is also added as the primary SMTP address on the default email address policy in the next step.

Important:
A public Domain Name System (DNS) MX resource record is required for each SMTP domain for which you accept email from the Internet. Each MX record should resolve to the Internet-facing server that receives email for your organization.

1. Open the EAC by browsing to the URL of your Client Access server. For example, https://Ex2013CAS/ECP.
2. Enter your user name and password in **Domain\user name** and **Password** and then click **Sign in**.
3. Go to **Mail flow** > **Accepted domains**. On the **Accepted domains** page, click **New +**.
4. In the **New accepted domain** wizard, specify a name for the accepted domain.
5. In the **Accepted domain** field, specify the SMTP recipient domain you want to add. For example, contoso.com.
6. Select **Authoritative domain** and then click **Save**.

How do you know this step worked?

To verify that you have successfully created an accepted domain, do the following:

- In the EAC, verify the new accepted domain appears in **Mail flow** > **Accepted domains**.

Step 3: Configure the default email address policy

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Email address policies" entry in the Email address and address book permissions topic.

If you added an accepted domain in the previous step and you want that domain to be added to every recipient in the organization, you need to update the default email address policy.

1. Open the EAC by browsing to the URL of your Client Access server. For example, https://Ex2013CAS/ECP.
2. Enter your user name and password in **Domain\user name** and **Password** and then click **Sign in**.
3. Go to **Mail flow** > **Email address policies**. On the **Email address policies** page, select **Default Policy** and then click **Edit** ✎.
4. On the **Default Policy Email Address Policy** page, click **Email Address Format**.
5. Under **Email address format**, click the SMTP address you want to change and then click **Edit** ✎.
6. On the **Email address format** page in the **Email address parameters** field, specify the SMTP recipient domain you want to apply to all recipients in the Exchange organization. This domain must match the accepted domain you added in the previous step. For example, @contoso.com. Click **Save**.
7. Click **Save**

8. In the **Default Policy** details pane, click **Apply**.

Note:

We recommend that you configure a user principal name (UPN) that matches the primary email address of each user. If you don't provide a UPN that matches the email address of a user, the user will be required to manually provide their domain\user name or UPN in addition to their email address. If their UPN matches their email address, Outlook Web App, ActiveSync, and Outlook will automatically match their email address to their UPN.

How do you know this step worked?

To verify that you have successfully configured the default email address policy, do the following:

1. In the EAC, go to **Recipients** > **Mailboxes**.
2. Select a mailbox and then, in the recipient details pane, verify that the **User mailbox** field has been set to <*alias*>@<*new accepted domain*>. For example, david@contoso.com.
3. Optionally, create a new mailbox and verify the mailbox is given an email address with the new accepted domain by doing the following:
   a. Go to **Recipients** > **Mailboxes**, click **New +** and then select **User mailbox**.
   b. On the new user mailbox page, provide the information required to create a new mailbox. Click **Save**.
   c. Select the new mailbox and then, in the recipient details pane, verify that the **User mailbox** field has been set to <*alias*>@<*new accepted domain*>. For example, david@contoso.com.

Step 4: Configure external URLs

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "<*Service*> virtual directory settings" entry in the Clients and mobile devices permissions topic.

Before clients can connect to your new server from the Internet, you need to configure the external domains, or URLs, on the Client Access server's virtual directories and then configure your public domain name service (DNS) records. The steps below configure the same external domain on the external URL of each virtual directory. If you want to configure different external domains on one or more virtual directory external URLs, you need to configure the external URLs manually. For more information, see Virtual directory management.

1. Open the EAC by browsing to the URL of your Client Access server. For example, https://Ex2013CAS/ECP.
2. Enter your user name and password in **Domain\user name** and **Password** and then click **Sign in**.
3. Go to **Servers** > **Servers**, select the name of the Internet-facing Client Access server and then click **Edit** ✎.
4. Click **Outlook Anywhere**.
5. In the **Specify the external hostname** field, specify the externally accessible FQDN of the Client Access server. For example, mail.contoso.com.
6. While you're here, let's also set the internally accessible FQDN of the Client Access server. In the **Specify the internal hostname** field, insert the FQDN you used in the previous step. For

example, mail.contoso.com.

7. Click **Save**.
8. Go to **Servers** > **Virtual directories** and then click **Configure external access domain**.
9. Under **Select the Client Access servers to use with the external URL**, click **Add +**
10. Select the Client Access servers you want to configure and then click **Add**. After you've added all of the Client Access servers you want to configure, click **OK**.
11. In **Enter the domain name you will use with your external Client Access servers**, type the external domain you want to apply. For example, mail.contoso.com. Click **Save**.

Note:

Some organizations make the Outlook Web App FQDN unique to protect users against changes to underlying server FQDN changes. Many organizations use owa.contoso.com for their Outlook Web App FQDN instead of mail.contoso.com. If you want to configure a unique Outlook Web App FQDN, do the following after you completed the previous step. This checklist assumes you have configured a unique Outlook Web App FQDN.

a. Select **owa (Default Web Site)** and click **Edit**.
b. In **External URL**, type **https://**, then the unique Outlook Web App FQDN you want to use, and then append **/owa**. For example, https://owa.contoso.com/owa.
c. Click **Save**.
d. Select **ecp (Default Web Site)** and click **Edit**.
e. In **External URL**, type **https://**, then the same Outlook Web App FQDN that you specified in the previous step, and then append **/ecp**. For example, https://owa.contoso.com/ecp.
f. Click **Save**.

After you've configured the external URL on the Client Access server virtual directories, you need to configure your public DNS records for Autodiscover, Outlook Web App, and mail flow. The public DNS records should point to the external IP address or FQDN of your Internet-facing Client Access server and use the externally accessible FQDNs that you've configured on your Client Access server. The following are examples of recommended DNS records that you should create to enable mail flow and external client connectivity.

FQDN	DNS record type	Value
Contoso.com	MX	Mail.contoso.com
Mail.contoso.com	A	172.16.10.11
Owa.contoso.com	CNAME	Mail.contoso.com
Autodiscover.contoso.com	CNAME	Mail.contoso.com

How do you know this step worked?

To verify that you have successfully configured the external URL on the Client Access server virtual directories, do the following:

1. In the EAC, go to **Servers** > **Virtual directories**.
2. In the **Select server** field, select the Internet-facing Client Access server.
3. Select a virtual directory and then, in the virtual directory details pane, verify that the **External URL** field is populated with the correct FQDN and service as shown below:

Virtual directory	External URL value
Autodiscover	No external URL displayed
ECP	https://owa.contoso.com/ecp
EWS	https://mail.contoso.com/EWS/Exchange.asmx
Microsoft-Server-ActiveSync	https://mail.contoso.com/Microsoft-Server-ActiveSync
OAB	https://mail.contoso.com/OAB
OWA	https://owa.contoso.com/owa
PowerShell	http://mail.contoso.com/PowerShell

To verify that you have successfully configured your public DNS records, do the following:

1. Open a command prompt and run `nslookup.exe`.
2. Change to a DNS server that can query your public DNS zone.
3. In `nslookup`, look up the record of each FQDN you created. Verify that the value that's returned for each FQDN is correct.
4. In `nslookup`, type `set type=mx` and then look up the accepted domain you added in Step 1. Verify that the value returned matches the FQDN of the Client Access server.

Step 5: Configure internal URLs

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "<*Service*> virtual directory settings" entry in the Clients and mobile devices permissions topic.

Before clients can connect to your new server from yourintranet, you need to configure the internal domains, or URLs, on the Client Access server's virtual directories and then configure your private domain name service (DNS) records.

The procedure below lets you choose whether you want users to use the same URL on your intranet and on the Internet to access your Exchange server or whether they should use a different URL. What you choose depends on the addressing scheme you have in place already or that you want to implement. If you're implementing a new addressing scheme, we recommend that you use the same URL for both internal and external URLs. Using the same URL makes it easier for users to access your Exchange server because they only have to remember one address. Regardless of the choice you make, you need to make sure you configure a private DNS zone for the address space you configure. For more information about administering DNS zones, see Administering DNS Server.

For more information about internal and external URLs on virtual directories, see Virtual directory management.

Configure internal and external URLs to be the same

1. Open the Exchange Management Shell on your Client Access server.
2. Store the host name of your Client Access server in a variable that will be used in the next step. For example, Ex2013CAS.

```
$HostName = "Ex2013CAS"
```

3. Run each of the following commands in the Shell to configure each internal URL to match the virtual directory's external URL.

```
Set-EcpVirtualDirectory "$HostName\ECP (Default Web Site)"
-InternalUrl ((Get-EcpVirtualDirectory "$HostName\ECP
(Default Web Site)").ExternalUrl)
Set-WebServicesVirtualDirectory "$HostName\EWS (Default Web
Site)" -InternalUrl ((get-WebServicesVirtualDirectory
"$HostName\EWS (Default Web Site)").ExternalUrl)
Set-ActiveSyncVirtualDirectory "$HostName\Microsoft-Server-
ActiveSync (Default Web Site)" -InternalUrl ((Get-
ActiveSyncVirtualDirectory "$HostName\Microsoft-Server-
ActiveSync (Default Web Site)").ExternalUrl)
Set-OabVirtualDirectory "$HostName\OAB (Default Web Site)"
-InternalUrl ((Get-OabVirtualDirectory "$HostName\OAB
(Default Web Site)").ExternalUrl)
Set-OwaVirtualDirectory "$HostName\OWA (Default Web Site)"
-InternalUrl ((Get-OwaVirtualDirectory "$HostName\OWA
(Default Web Site)").ExternalUrl)
Set-PowerShellVirtualDirectory "$HostName\PowerShell
(Default Web Site)" -InternalUrl ((Get-
PowerShellVirtualDirectory "$HostName\PowerShell (Default
Web Site)").ExternalUrl)
```

After you've configured the internal URL on the Client Access server virtual directories, you need to configure your private DNS records for Outlook Web App, and other connectivity. Depending on your configuration, you'll need to configure your private DNS records to point to the internal or external IP address or fully qualified domain name (FQDN) of your Client Access server. The following are examples of recommended DNS records that you should create to enable internal client connectivity.

FQDN	DNS record type	Value

Mail.contoso.com	CNAME	Ex2013CAS.corp.contoso.com
Owa.contoso.com	CNAME	Ex2013CAS.corp.contoso.com

How do you know this step worked?

To verify that you have successfully configured the internal URL on the Client Access server virtual directories, do the following:

1. In the EAC, go to **Servers** > **Virtual directories**.
2. In the **Select server** field, select the Internet-facing Client Access server.
3. Select a virtual directory and then click **Edit** .
4. Verify that the **Internal URL** field is populated with the correct FQDN and service as shown below:

Virtual directory	Internal URL value
Autodiscover	No internal URL displayed
ECP	https://owa.contoso.com/ecp
EWS	https://mail.contoso.com/EWS/Exchange.asmx
Microsoft-Server-ActiveSync	https://mail.contoso.com/Microsoft-Server-ActiveSync
OAB	https://mail.contoso.com/OAB
OWA	https://owa.contoso.com/owa
PowerShell	http://mail.contoso.com/PowerShell

To verify that you have successfully configured your private DNS records, do the following:

1. Open a command prompt and run `nslookup.exe`.
2. Change to a DNS server that can query your private DNS zone.
3. In `nslookup`, look up the record of each FQDN you created. Verify that the value that's returned for each FQDN is correct.

Configure different internal and external URLs

1. Open the EAC by browsing to the URL of your Client Access server. For example, https://Ex2013CAS/ECP.
2. Go to **Servers** > **Virtual directories**.
3. In the **Select server** field, select the Internet-facing Client Access server.
4. Select the virtual directory you want to change and click **Edit** .
5. In **Internal URL**, replace the host name between **https://** and the first forward slash (**/**) with the new FQDN you want to use. For example, if you want to change the EWS virtual directory FQDN from Ex2013CAS.corp.contoso.com to internal.contoso.com, change the internal URL from https://Ex2013CAS.corp.contoso.com/ews/exchange.asmx to https://internal.contoso.com/ews/exchange.asmx.

6. Click **Save**.
7. Repeat steps 5 and 6 for each virtual directory you want to change.

Note:
The ECP and OWA virtual directory internal URLs must be the same. You can't set an internal URL on the Autodiscover virtual directory.

After you've configured the internal URL on the Client Access server virtual directories, you need to configure your private DNS records for Outlook Web App, and other connectivity. Depending on your configuration, you'll need to configure your private DNS records to point to the internal or external IP address or FQDN of your Client Access server. The following is an example of recommended DNS record that you should create to enable internal client connectivity if you've configured your virtual directory internal URLs to use internal.contoso.com.

FQDN	DNS record type	Value
internal.contoso.com	CNAME	Ex2013CAS.corp.contoso.com

How do you know this step worked?

To verify that you have successfully configured the internal URL on the Client Access server virtual directories, do the following:

1. In the EAC, go to **Servers** > **Virtual directories**.
2. In the **Select server** field, select the Internet-facing Client Access server.
3. Select a virtual directory and then click **Edit** .
4. Verify that the **Internal URL** field is populated with the correct FQDN. For example, you may have set the internal URLs to use internal.contoso.com.

Virtual directory	Internal URL value
Autodiscover	No internal URL displayed
ECP	https://internal.contoso.com/ecp
EWS	https://internal.contoso.com/EWS/Exchange.asmx
Microsoft-Server-ActiveSync	https://internal.contoso.com/Microsoft-Server-ActiveSync
OAB	https://internal.contoso.com/OAB
OWA	https://internal.contoso.com/owa
PowerShell	http://internal.contoso.com/PowerShell

To verify that you have successfully configured your private DNS records, do the following:

1. Open a command prompt and run `nslookup.exe`.
2. Change to a DNS server that can query your private DNS zone.
3. In `nslookup`, look up the record of each FQDN you created. Verify that the value that's returned for

each FQDN is correct.

Step 6: Configure an SSL certificate

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Certificate management" entry in the Mail flow permissions topic.

Some services, such as Outlook Anywhere and Exchange ActiveSync, require certificates to be configured on your Exchange 2013 server. The following steps show you how to configure an SSL certificate from a third-party certificate authority (CA):

1. Open the EAC by browsing to the URL of your Client Access server. For example, https://Ex2013CAS/ECP.
2. Enter your user name and password in **Domain\user name** and **Password** and then click **Sign in**.
3. Go to **Servers** > **Certificates**. On the **Certificates** page, make sure your Client Access server is selected in the **Select server** field, and then click **New +**.
4. In the **New Exchange certificate** wizard, select **Create a request for a certificate from a certification authority** and then click **Next**.
5. Specify a name for this certificate and then click **Next**.
6. If you want to request a wildcard certificate, select **Request a wild-card certificate** and then specify the root domain of all subdomains in the **Root domain** field. If you don't want to request a wildcard certificate and instead want to specify each domain you want to add to the certificate, leave this page blank. Click **Next**.
7. Click **Browse** and specify an Exchange server to store the certificate on. The server you select should be the Internet-facing Client Access server. Click **Next**.
8. For each service in the list shown, verify that the external or internal server names that users will use to connect to the Exchange server are correct. For example:
   - If you configured your internal and external URLs to be the same, **Outlook Web App (when accessed from the Internet)** and **Outlook Web App (when accessed from the Intranet)** should show owa.contoso.com. **OAB (when accessed from the Internet)** and **OAB (when accessed from the Intranet)** should show mail.contoso.com.
   - If you configured the internal URLs to be internal.contoso.com, **Outlook Web App (when accessed from the Internet)** should show owa.contoso.com and **Outlook Web App (when accessed from the Intranet)** should show internal.contoso.com.

These domains will be used to create the SSL certificate request. Click **Next**.

9. Add any additional domains you want included on the SSL certificate.
10. Select the domain that you want to be the common name for the certificate, and then click **Set as common name**. For example, contoso.com. Click **Next**.
11. Provide information about your organization. This information will be included with the SSL certificate. Click **Next**.
12. Specify the network location where you want this certificate request to be saved. Click **Finish**.

After you've saved the certificate request, submit the request to your certificate authority (CA). This can be an internal CA or a third-party CA, depending on your organization. Clients that connect to

the Client Access server must trust the CA that you use. After you receive the certificate from the CA, complete the following steps:

1. On the **Server** > **Certificates** page in the EAC, select the certificate request you created in the previous steps.
2. In the certificate request details pane, click **Complete** under **Status**.
3. On the complete pending request page, specify the path to the SSL certificate file and then click **OK**.
4. Select the new certificate you just added, and then click **Edit** .
5. On the certificate page, click **Services**.
6. Select the services you want to assign to this certificate. At minimum, you should select **SMTP** and **IIS**. Click **Save**.
7. If you receive the warning **Overwrite the existing default SMTP certificate?**, click **Yes**.

How do you know this step worked?

To verify that you have successfully added a new certificate, do the following:

1. In the EAC, go to **Servers** > **Certificates**.
2. Select the new certificate and then, in the certificate details pane, verify that the following are true:
   - **Status** shows **Valid**
   - **Assigned to services** shows, at minimum, **IIS** and **SMTP**.

How do you know this task worked?

To verify that you have configured mail flow and external client access, do the following:

1. In Outlook, on an Exchange ActiveSync device, or on both, create a new profile. Verify that Outlook or the mobile device successfully creates the new profile.
2. In Outlook, or on the mobile device, send a new message to an external recipient. Verify the external recipient receives the message.
3. In the external recipient's mailbox, reply to the message you just sent from the Exchange mailbox. Verify the Exchange mailbox receives the message.
4. Go to https://owa.contoso.com/owa and verify that there are no certificate warnings.

Verify an Exchange 2013 installation

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-10*

After you install Microsoft Exchange Server 2013, we recommend that you verify the installation by running the **Get-ExchangeServer** cmdlet and by reviewing the setup log file. If the setup process

fails or errors occur during installation, you can use the setup log file to track down the source of the problem.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Run Get-ExchangeServer

To verify that Exchange 2013 installed successfully, run the **Get-ExchangeServer** cmdlet in the Exchange Management Shell. A list is displayed of all Exchange 2013 server roles that are installed on the specified server when this cmdlet is run.

For detailed syntax and parameter information, see Get-ExchangeServer.

Review the setup log file

You can also learn more about the installation and configuration of Exchange 2013 by reviewing the setup log file created during the setup process.

During installation, Exchange Setup logs events in the **Application** log of **Event Viewer** on computers that are running Windows Server 2008 R2 with Service Pack 1 (SP1) and Windows Server 2012. Review the **Application** log, and make sure there are no warning or error messages related to Exchange setup. These log files contain a history of each action that the system takes during Exchange 2013 setup and any errors that may have occurred. By default, the logging method is set to verbose. Information is available for each installed server role.

You can find the setup log file at <*system drive*>\ExchangeSetupLogs\ExchangeSetup.log. The <*system drive*> variable represents the root directory of the drive where the operating system is installed.

The setup log file tracks the progress of every task that is performed during the Exchange 2013 installation and configuration. The file contains information about the status of the prerequisite and system readiness checks that are performed before installation starts, the application installation progress, and the configuration changes that are made to the system. Check this log file to verify that the server roles were installed as expected.

We recommend that you start your review of the setup log file by searching for any errors. If you find an entry that indicates that an error occurred, read the associated text to determine the cause of the error.

Install the Exchange 2013 management

tools

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-28*

With the Microsoft Exchange Server 2013 management tools, you can configure and manage your Exchange organization remotely. Exchange 2013 management tools include the Exchange Management Shell and the Exchange Toolbox. This topic explains how you can either use Setup.exe or unattended setup mode to install the Exchange 2013 management tools.

Note:
You don't need to perform this procedure to use the Exchange Administration Center (EAC) remotely. The EAC is a web-based console that's hosted on computers running the Exchange 2013 Client Access server role. For more information about accessing the EAC remotely, see Exchange admin center in Exchange 2013.

For more information about managing Exchange 2013, see Exchange admin center in Exchange 2013 and Exchange Management Shell.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes
- Make sure you've read the release notes prior to installing Exchange 2013. For more information, see Release notes for Exchange 2013.
- The computer you install the management tools on must have a supported operating system (such as Windows Server 2012 or Windows 8), have enough disk space, be a member of an Active Directory domain, and satisfy other requirements. For information about system requirements, see Exchange 2013 system requirements.
- To run Exchange 2013 Setup, you must install Microsoft .NET Framework 4.5, Windows Management Framework 3.0, and other required software. To understand the prerequisites for all server roles, see Exchange 2013 prerequisites.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use Setup to install the Exchange 2013 management tools

1. Log on to the computer on which you want to install Exchange 2013.
2. Navigate to the network location of the Exchange 2013 installation files.
3. Start Exchange 2013 Setup by double-clicking `setup.exe`

Important:

If you have User Access Control (UAC) enabled, you must right-click `Setup.exe` and select **Run as administrator**.

4. On the **Check for Updates** page, choose whether you want Setup to connect to the Internet and download product and security updates for Exchange 2013. If you select **Connect to the Internet and check for updates**, Setup will download updates and apply them prior to continuing. If you select **Don't check for updates right now**, you can download and install updates manually later. We recommend that you download and install updates now. Click **Next** to continue.
5. The **Introduction** page begins the process of installing Exchange into your organization. It will guide you through the installation. Several links to helpful deployment content are listed. We recommend that you visit these links prior to continuing setup. Click **Next** to continue.
6. On the **License Agreement** page, review the software license terms. If you agree to the terms, select **I accept the terms in the license agreement**, and then click **Next**.
7. On the **Recommended settings** page, select whether you want to use the recommended settings. If you select **Use recommended settings**, Exchange will automatically send error reports and information about your computer hardware and how you use Exchange to Microsoft. If you select **Don't use recommended settings**, these settings remain disabled but you can enable them at any time after Setup completes. For more information about these settings and how information sent to Microsoft is used, click **?**.
8. On the **Server Role Selection** page, verify that **Management Tools** is selected.

Select **Automatically install Windows Server roles and features that are required to install Exchange Server** to have the Setup wizard install required Windows prerequisites. You may need to reboot the computer to complete the installation of some Windows features. If you don't select this option, you must install the Windows features manually.

Note:

This option installs only the Windows features required by Exchange. You must manually install other prerequisites manually. For more information, see Exchange 2013 prerequisites.

Click **Next** to continue.

9. On the **Installation Space and Location** page, either accept the default installation location or click **Browse** to choose a new location. Make sure that you have enough disk space available in the location where you want to install Exchange. Click **Next** to continue.
10. If this is the first time you've run Exchange 2013 Setup in your organization, on the **Exchange Organization** page, type a name for your Exchange organization. The Exchange organization name can contain only the following characters:
    - A through Z
    - a through z
    - 0 through 9
    - Space (not leading or trailing)
    - Hyphen or dash

Note:

The organization name can't contain more than 64 characters. The organization name can't be blank.

If you want to use the Active Directory split permissions model, select **Apply Active Directory split permission security model to the Exchange organization**.

Caution:
Most organizations don't need to apply the Active Directory split permissions model. If you need to separate management of Active Directory security principals and Exchange configuration, Role Based Access Control (RBAC) split permissions might work for you. For more information, click **?**.

Click **Next** to continue.

11. On the **Readiness Checks** page, view the status to determine if the organization and server role prerequisite checks completed successfully. If they haven't completed successfully, you must resolve any reported errors before you can install Exchange 2013. You don't need to exit Setup when resolving some of the prerequisite errors. After resolving a reported error, click **back** and then click **Next** to run the prerequisite check again. Be sure to also review any warnings that are reported. If all readiness checks have completed successfully, click **Next** to install Exchange 2013.
12. On the **Completion** page, click **Finish**.
13. Restart the computer after Exchange 2013 has completed.

Use unattended Setup mode to install the Exchange 2013 management tools

1. Log on to the computer on which you want to install the Exchange 2013 management tools.
2. Navigate to the network location of the Exchange 2013 installation files.
3. At the command prompt, run the following command.

Important:
If you have User Access Control (UAC) enabled, you must run `Setup.exe` from an elevated command prompt.

```
Setup.exe /Role:ManagementTools /
IAcceptExchangeServerLicenseTerms
```

For more information, see Install Exchange 2013 using unattended mode.

Exchange 2013 virtualization

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-23*

You can deploy Microsoft Exchange Server 2013 in a virtualized environment. This topic provides an overview of the scenarios that are supported for deploying Exchange 2013 on hardware virtualization software.

Contents

The following terms are used in this discussion of Exchange virtualization:

- **Cold boot** When bringing a system from a power-off state into a clean start of the operating system, the action is a *cold boot*. No operating system state has been persisted in this case.
- **Saved state** When a virtual machine is powered off, hypervisors typically have the ability to save the state of the virtual machine, so when the machine is powered back on, it returns to that *saved state* rather than going through a cold boot startup.
- **Planned migration** When a system administrator initiates the move of a virtual machine from one hypervisor host to another, the action is a *planned migration*. The action could be a single migration, or a system administrator could configure automation to move the virtual machine on a timed basis. A planned migration could also be the result of some other event that occurs in the system, other than hardware or software failure. The key point is the Exchange virtual machine is operating normally and needs to be relocated for some reason. This relocation can be done via technology, like Live Migration or vMotion. However, if the Exchange virtual machine or the hypervisor host where the virtual machine is located experiences some sort of failure condition, the outcome isn't characterized as a planned migration.

Requirements for hardware virtualization

Microsoft supports Exchange 2013 in production on hardware virtualization software only when all the following conditions are true:

- The hardware virtualization software is running one of the following:
  - Any version of Windows Server with Hyper-V technology or Microsoft Hyper-V Server
  - Any third-party hypervisor that has been validated under the Windows Server Virtualization Validation Program.

Note:
Deployment of production Exchange servers on Windows Azure virtual machines isn't supported.

- The Exchange guest virtual machine has the following conditions:
  - It's running Exchange 2013.

- It's deployed on Windows Server 2008 R2 SP1 (or later versions), Windows Server 2012, or on Windows Server 2012 R2.

For deployments of Exchange 2013:

- All Exchange 2013 server roles are supported in a virtual machine.
- Exchange server virtual machines (including Exchange Mailbox virtual machines that are part of a database availability group, or DAG), may be combined with host-based failover clustering and migration technology, as long as the virtual machines are configured such that they won't save and restore state on disk when moved or taken offline. All failover activity occurring at the hypervisor level must result in a cold boot when the virtual machine is activated on the target node. All planned migration must either result in shutdown and cold boot, or an online migration that makes use of a technology like Hyper-V Live Migration. Hypervisor migration of virtual machines is supported by the hypervisor vendor; therefore, you must ensure that your hypervisor vendor has tested and supports migration of Exchange virtual machines. Microsoft supports Hyper-V Live Migration of these virtual machines.
- Only management software (for example, antivirus software, backup software, or virtual machine management software) can be deployed on the physical host machine. No other server-based applications (for example, Exchange, SQL Server, Active Directory, or SAP) should be installed on the host machine. The host machine should be dedicated to running guest virtual machines.
- Some hypervisors include features for taking snapshots of virtual machines. Virtual machine snapshots capture the state of a virtual machine while it's running. This feature enables you to take multiple snapshots of a virtual machine and then revert the virtual machine to any of the previous states by applying a snapshot to the virtual machine. However, virtual machine snapshots aren't application aware, and using them can have unintended and unexpected consequences for a server application that maintains state data, such as Exchange. As a result, making virtual machine snapshots of an Exchange guest virtual machine isn't supported.
- Many hardware virtualization products allow you to specify the number of virtual processors that should be allocated to each guest virtual machine. The virtual processors located in the guest virtual machine share a fixed number of logical processors in the physical system. Exchange supports a virtual processor-to-logical processor ratio no greater than 2:1, although we recommend a ratio of 1:1. For example, a dual processor system using quad core processors contains a total of 8 logical processors in the host system. On a system with this configuration, don't allocate more than a total of 16 virtual processors to all guest virtual machines combined.
- When calculating the total number of virtual processors required by the host machine, you must also account for both I/O and operating system requirements. In most cases, the equivalent number of virtual processors required in the host operating system for a system hosting Exchange virtual machines is 2. This value should be used as a baseline for the host operating system virtual processor when calculating the overall ratio of physical cores to virtual processors. If performance monitoring of the host operating system indicates you're consuming more processor utilization than the equivalent of 2 processors, you should reduce the count of virtual processors assigned to guest virtual machines accordingly and verify that the overall virtual processor-to-physical core ratio is no greater than 2:1.
- The operating system for an Exchange guest machine must use a disk that has a size equal to at

least 15 gigabytes (GB) plus the size of the virtual memory that's allocated to the guest machine. This requirement is necessary to account for the operating system and paging file disk requirements. For example, if the guest machine is allocated 16 GB of memory, the minimum disk space needed for the guest operating system disk is 31 GB.

In addition, it's possible that guest virtual machines may be prevented from directly communicating with Fibre Channel or SCSI host bus adapters (HBAs) installed in the host machine. In this event, you must configure the adapters in the host machine's operating system and present the logical unit numbers (LUNs) to guest virtual machines as either a virtual disk or a pass-through disk.

Return to top

Host machine storage requirements

The minimum disk space requirements for each host machine are as follows:

- Host machines in some hardware virtualization applications may require storage space for an operating system and its components. For example, when running Windows Server 2008 R2 with Hyper-V, you will need a minimum of 10 GB to meet the requirements for Windows Server 2008. For more details, see Windows Server 2008 R2 System Requirements. Additional storage space is also required to support the operating system's paging file, management software, and crash recovery (dump) files.
- Some hypervisors maintain files on the host machine that are unique to each guest virtual machine. For example, in a Hyper-V environment, a temporary memory storage file (BIN file) is created and maintained for each guest machine. The size of each BIN file is equal to the amount of memory allocated to the guest machine. In addition, other files may also be created and maintained on the host machine for each guest machine.
- If your host machine is running Windows Server 2012 Hyper-V or Hyper-V 2012, and you are configuring a host-based failover cluster that will host Exchange Mailbox servers in a database availability group, then we recommend following the guidance documented in Microsoft Knowledge Base article, 2872325, Guest Cluster nodes in Hyper-V may not be able to create or join.

Return to top

Exchange storage requirements

Requirements for storage connected to a virtualized Exchange server are as follows:

- Each Exchange guest machine must be allocated sufficient storage space on the host machine for the fixed disk that contains the guest's operating system, any temporary memory storage files in use, and related virtual machine files that are hosted on the host machine. In addition, for each Exchange guest machine, you must also allocate sufficient storage for the message queues and sufficient storage for the databases and log files on Mailbox servers.
- The storage used by the Exchange guest machine for storage of Exchange data (for example, mailbox databases and transport queues) can be virtual storage of a fixed size (for example,

fixed virtual hard disks (VHDs) in a Hyper-V environment), SCSI pass-through storage, or Internet SCSI (iSCSI) storage. Pass-through storage is storage that's configured at the host level and dedicated to one guest machine. All storage used by an Exchange guest machine for storage of Exchange data must be block-level storage because Exchange 2013 doesn't support the use of network attached storage (NAS) volumes, other than in the SMB 3.0 scenario outlined later in this topic. Also, NAS storage that's presented to the guest as block-level storage via the hypervisor isn't supported.

- Fixed VHDs may be stored on SMB 3.0 files that are backed by block-level storage if the guest machine is running on Windows Server 2012 Hyper-V (or a later version of Hyper-V). The only supported usage of SMB 3.0 file shares is for storage of fixed VHDs. Such file shares can't be used for direct storage of Exchange data. When using SMB 3.0 file shares to store fixed VHDs, the storage backing the file share should be configured for high availability to ensure the best possible availability of the Exchange service.
- Storage used by Exchange should be hosted in disk spindles that are separate from the storage that's hosting the guest virtual machine's operating system.
- Configuring iSCSI storage to use an iSCSI initiator inside an Exchange guest virtual machine is supported. However, there is reduced performance in this configuration if the network stack inside a virtual machine isn't full-featured (for example, not all virtual network stacks support jumbo frames).

Return to top

Exchange memory requirements and recommendations

Some hypervisors have the ability to oversubscribe or dynamically adjust the amount of memory available to a specific guest machine based on the perceived usage of memory in the guest machine as compared to the needs of other guest machines managed by the same hypervisor. This technology makes sense for workloads in which memory is needed for brief periods of time and then can be surrendered for other uses. However, it doesn't make sense for workloads that are designed to use memory on an ongoing basis. Exchange, like many server applications with optimizations for performance that involve caching of data in memory, is susceptible to poor system performance and an unacceptable client experience if it doesn't have full control over the memory allocated to the physical or virtual machine on which it's running. As a result, using dynamic memory features for Exchange isn't supported.

Return to top

Host-based failover clustering and migration for Exchange

The following are answers to some frequently asked questions about host-based failover clustering and migration technology with Exchange 2013 DAGs:

- **Does Microsoft support third-party migration technology?**

Microsoft can't make support statements for the integration of third party hypervisor products using

these technologies with Exchange, because these technologies aren't part of the Server Virtualization Validation Program (SVVP). The SVVP covers the other aspects of Microsoft support for third-party hypervisors. You need to ensure that your hypervisor vendor supports the combination of their migration and clustering technology with Exchange. If your hypervisor vendor supports their migration technology with Exchange, Microsoft supports Exchange with their migration technology.

- **How does Microsoft define host-based failover clustering?**

Host-based failover clustering refers to any technology that provides the automatic ability to react to host-level failures and start affected virtual machines on alternate servers. Use of this technology is supported given that, in a failure scenario, the virtual machine is coming up from a cold boot on the alternate host. This technology helps to make sure that the virtual machine never comes up from a saved state that's persisted on disk because it will be stale relative to the rest of the DAG members.

- **What does Microsoft mean by migration support?**

Migration technology refers to any technology that allows a planned move of a virtual machine from one host machine to another host machine. This move could also be an automated move that occurs as part of resource load balancing, but it isn't related to a failure in the system. Migrations are supported as long as the virtual machines never come up from a saved state that's persisted on disk. This means that technology that moves a virtual machine by transporting the state and virtual machine memory over the network with no perceived downtime is supported for use with Exchange. A third-party hypervisor vendor must provide support for the migration technology, while Microsoft provides support for Exchange when used in this configuration.

Return to top

Integration with SharePoint and Lync

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-11*

Microsoft Exchange Server 2013 includes many features that integrate with Microsoft SharePoint 2013 and Microsoft Lync 2013. Together, these products offer a suite of features that make scenarios such as enterprise eDiscovery and collaboration using site mailboxes possible.

Archiving, hold, and eDiscovery

Archiving of email and documents, preserving them for the duration required to meet regulatory compliance and business requirements, and ability to search them quickly to fulfill eDiscovery requests is critical in most organizations. Exchange 2013, SharePoint 2013 and Lync Server 2013

together provide integrated archiving, hold and eDiscovery functionality, allowing you to preserve important data in-place across Exchange mailboxes, SharePoint documents and web sites, and archived Lync content. The eDiscovery Center in SharePoint 2013 allows authorized discovery managers to perform an eDiscovery search for content across these stores, preview search results, and export the data for legal review.

Archive Lync Server 2013 content in Exchange 2013

With Lync Server 2013 deployed in an organization with Exchange 2013, you can configure Lync to archive instant messaging and on-line meeting content such as shared presentations or documents in the user's Exchange 2013 mailbox. Archiving Lync data in Exchange 2013 allows you to apply retention policies to it. Archived Lync content also surfaces in any eDiscovery searches. For more details about Lync Archiving and how to deploy it, see the following topics:

- Planning for Archiving
- Deploying Archiving

Search Exchange, SharePoint and Lync data using the SharePoint 2013 eDiscovery Center

Exchange 2013 allows SharePoint 2013 to search Exchange mailbox content using Federated search API. SharePoint 2013 provides an eDiscovery Center to allow authorized personnel to perform eDiscovery. Microsoft Search Foundation provides a common indexing and search infrastructure to both Exchange 2013 and SharePoint 2013 and allows you to use the same query syntax across both applications. This ensures an eDiscovery search performed in SharePoint 2013 will return the same Exchange 2013 content as the same search performed using In-Place eDiscovery in Exchange 2013. SharePoint 2013 eDiscovery Center also allows you to export content returned in an eDiscovery search, including export of Exchange 2013 content to a PST file.

For more details, see the following topics:

- In-Place eDiscovery
- In-Place Hold
- Configure eDiscovery in SharePoint 2013
- What's new in eDiscovery in SharePoint Server 2013
- Configure Exchange for SharePoint eDiscovery Center

Site mailboxes

In many organizations, information resides in two different stores - email in Microsoft Exchange and documents in SharePoint, with two different interfaces to access them. This causes a disjointed user experience and impedes effective collaboration. Site mailboxes allow users to collaborate effectively by bringing together Exchange emails and SharePoint documents. For users, a site mailbox serves as a central filing cabinet, providing a place to file project emails and documents

that can only be accessed and edited by site members. Site mailboxes are surfaced in Outlook 2013 and give users easy access to the email and documents for the projects they care about. Additionally, the same set of content can be accessed directly from the SharePoint site itself.

Under the covers of a site mailbox, the content is kept where it belongs. Exchange stores the email, providing users with the same message view for email conversations that they use every day for their own mailboxes. SharePoint stores the documents, bringing document coauthoring and versioning to the table. Exchange synchronizes just enough metadata from SharePoint to create the document view in Outlook (e.g. document title, last modified date, last modified author, size).

Site mailboxes are provisioned and managed from SharePoint 2013. For more details, see the following topics:.

- Site mailboxes
- Configure site mailboxes in SharePoint Server 2013

Unified contact store

Unified contact store (UCS) is a feature that provides a consistent contact experience across Microsoft Office products. This feature enables users to store all contact information in their Exchange 2013 mailbox so that the same contact information is available globally across Lync, Exchange, Outlook and Outlook Web App.

After Lync Server 2013 is installed in an environment with Exchange 2013 and you have configured server-to-server authentication between the two, users can initiate the migration of existing contacts from Lync Server 2013 to Exchange 2013. For details, see Planning and Deploying Unified Contact Store.

User photos

User photos is a feature that allows you to store high resolution user photos in Exchange 2013 that can be accessed by client applications, including Outlook, Outlook Web App, SharePoint 2013, Lync 2013, and mobile email clients. A low-resolution photo is also stored in Active Directory. As with Unified contact stores, user photos allow your organization to maintain a consistent user profile photo that can be consumed by client applications without requiring each application to have its own user photos and different ways to add and manage them. Users can manage their own photos using Outlook Web App, SharePoint 2013 or Lync 2013. For detail about managing photos on Outlook Web App, see My account.

Lync presence in Outlook Web App

In Exchange 2013 environments with Lync Server 2013 installed, you can configure them to enable users to see presence information in Outlook Web App. Users can see their instant messaging contacts and groups in the Navigation Pane of Outlook Web App, respond to or initiate instant

messaging sessions from Outlook Web App and manage their instant messaging contacts and groups.

OAuth authentication

Exchange 2013, SharePoint 2013 and Lync Server 2013 provide the rich cross-product functionality described above using OAuth authorization protocol for server-to-server authentication. Using the same authentication protocol allows these applications to seamlessly and securely authenticate to each other. The authentication mechanism supports authentication as an application using the context of a linked account and user impersonation where the access request is made in the user context.

OAuth is a standard authorization protocol used by many web sites and web services. It allows clients to access resources provided by a resource server without having to provide a username and password. Authentication is performed by an authorization server trusted by the resource owner, which provides the client with an access token. The token grants access to a specific set of resources for a specified period. For more details about Exchange 2013's implementation of OAuth, see [MS-XOAUTH]: OAuth 2.0 Authorization Protocol Extensions.

OAuth in on-premises deployments

Within an on-premises deployment, Exchange 2013, SharePoint 2013 and Lync Server 2013 do not require an authorization server to issue tokens. Each of these applications issue self-signed tokens to access resources provided by other application. The application that provides access to resources, for example Exchange 2013, must trust the self-signed tokens presented by the calling application. Trust is established by creating a *partner application* configuration for the calling application, which includes the calling application's ApplicationID, certificate, and AuthMetadataUrl. Exchange 2013, SharePoint 2013 and Lync Server 2013 publish their auth metadata document in a well-known URL.

Server	**AuthMetadataUrl**
Exchange 2013	`https://<serverfqdn>/autodiscover/metadata/json/1`
Lync Server 2013	`https://<serverfqdn>/metadata/json/1`
SharePoint 2013	`https://<serverfqdn>/_layouts/15/metadata/json/1`

Exchange 2013 Server Auth Certificate

Exchange 2013 Setup creates a self-signed certificate with the friendly name Microsoft Exchange Server Auth Certificate. The certificate is replicated to all front-end servers in the Exchange 2013 organization. The certificate's thumbprint is specified in Exchange 2013's authorization configuration, along with its service name, a well-known GUID that represents on-premises Exchange 2013. Exchange uses the authorization configuration to publish its auth metadata

document.

Important:
The default Server Auth Certificate created by Exchange 2013 is valid for five years. You must ensure the authorization configuration includes a current certificate.

When Exchange 2013 receives an access request from a partner application via Exchange Web Services (EWS), it parses the `www-authenticate` header of the https request, which contains the access token signed by the calling server using its private key. The auth module validates the access token using the partner application configuration. It then grants access to resources based on the RBAC permissions granted to the application. If the access token is on behalf of a user, the RBAC permissions granted to the user are checked. For example, if a user performs an eDiscovery search using the eDiscovery Center in SharePoint 2013, Exchange checks whether the user is a member of the Discovery Management role group or has the Mailbox Search role assigned and the mailboxes being searched are within the scope of the RBAC role assignment. For more details, see Permissions.

Managing OAuth Authentication

In Exchange 2013, there are two configuration objects you must manage for OAuth authentication with partner applications:

- **AuthConfig** The AuthConfig is created by Exchange 2013 Setup and is used to publish the auth metadata. You don't need to manage the auth config except to provision a new certificate when the existing certificate is close to expiration. When this happens, you can renew the existing certificate and configure the new certificate as the next certificate in the AuthConfig along with its effective date. The new certificate is automatically replicated to other Exchange 2013 in the organization, the auth metadata document is refreshed with details of the new certificate, and the AuthConfig rolls over to the new certificate on the effective date.
- **Partner applications** To enable partner applications to request access tokens from Exchange 2013, you must create a partner application configuration. Exchange 2013 provides the `Configure-EnterprisePartnerApplication.ps1` script, which allows you to quickly and easily create partner application configurations and minimize configuration errors. For details, see Configure OAuth authentication with SharePoint 2013 and Lync 2013.

Configure OAuth authentication with SharePoint 2013 and Lync 2013

Exchange Server 2013 > Planning and deployment > Integration with SharePoint and Lync >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-03*

Exchange Server 2013 allows other applications to use OAuth to authenticate to Exchange. The applications must be configured as partner applications in Exchange 2013.

In Exchange 2013, OAuth configuration with partner applications such as SharePoint 2013 and Lync Server 2013 is supported only by using the `Configure-EnterpriseApplication.ps1` script. By automating the task, the script makes it easier to configure authentication with partner applications and reduces configuration errors. The script performs the following tasks:

1. Configures an Enterprise partner application that self-issues OAuth tokens to successfully authenticate to Exchange.
2. Assigns Role Based Access Control (RBAC) roles to the partner application to authorize it for calling specific Exchange Web Services APIs.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- The partner application must publish an auth metadata document for Exchange 2013 to establish a direct trust to this application and accept authentication requests.
- Examples in this topic use the following default location of the `\Scripts` directory: `C:\Program Files\Microsoft\Exchange Server\V15\Scripts`.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Configure OAuth authentication with a partner application

This procedure uses the `Configure-EntepririseApplication.ps1` script to configure OAuth authentication with partner applications. Access to resources depends on the permissions assigned to the partner application and/or the user it impersonates by using RBAC.

After configuring OAuth authentication from Exchange, the partner application can use Exchange 2013 resources. If Exchange 2013 also needs to access resources offered by the partner application, you must also configure OAuth authentication in the partner application.

This example configures OAuth authentication for SharePoint 2013.

```
Cd C:\Program Files\Microsoft\Exchange Server\V15\Scripts
Configure-EnterprisePartnerApplication.ps1 -AuthMetaDataUrl
```

```
https://sharepoint.contoso.com/_layouts/15/metadata/json/1
-ApplicationType SharePoint
```

This example configures OAuth authentication for Lync Server 2013.

```
Cd C:\Program Files\Microsoft\Exchange Server\V15\Scripts
Configure-EnterprisePartnerApplication.ps1 -AuthMetaDataUrl
https://lync.contoso.com/metadata/json/1 -ApplicationType
Lync
```

How do you know this worked?

To verify that you have successfully configured an enterprise partner application to authenticate to Exchange 2013 , run the Get-PartnerApplication cmdlet in the Shell to retrieve the configuration. You can also run the Test-OAuthConnectivity cmdlet to test OAuth connectivity with a partner application for a user.

More information

- In hybrid deployments, you can use OAuth authentication between your on-premises Exchange 2013 organization and the Exchange Online organization. For more information, see Using OAuth authentication to support eDiscovery in an Exchange hybrid deployment.
- In on-premises deployments, you can configure server-to-server authentication between Exchange 2013 and SharePoint 2013 so administrators and compliance officers can use the eDiscovery Center in SharePoint 2013 to search Exchange 2013 mailboxes. For more information, see Configure Exchange for SharePoint eDiscovery Center.

Deployment reference

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-08-02*

Exchange 2013: editions and versions

Exchange Server Updates: build numbers and release dates

Exchange Server Supportability Matrix

Exchange 2013 deployment permissions reference

Deployment security checklist

Exchange 2013 sizing and capacity planning

Exchange 2013 storage configuration options

IPv6 support in Exchange 2013

Exchange 2013 language support

Exchange 2013 readiness checks

What changes in Active Directory when Exchange 2013 is installed?

Applies to: *Exchange Server*

Topic Last Modified: *2014-05-26*

When you install Exchange 2013, changes are made to your Active Directory forest and domains. Exchange does this so that it can store information about the Exchange servers, mailboxes, and other objects related to Exchange in your organization. These changes are made for you when you run the Exchange 2013 Setup wizard or when you run the *PrepareSchema*, *PrepareAD*, and *PrepareDomains* commands (see how to use these commands in Prepare Active Directory and domains) during Exchange 2013 command-line Setup. If you're curious about the changes that Exchange makes to Active Directory, this topic is for you. It explains what Exchange does at each step of Active Directory preparation.

There are three steps that need to be done to prepare Active Directory for Exchange:

- Extend the Active Directory schema
- Prepare Active Directory containers, objects, and other items
- Prepare Active Directory domains

After all three steps are done, your Active Directory forest is ready for Exchange 2013. You can find out more about how to install Exchange 2013 by reading Install Exchange 2013 using the Setup wizard.

Extend the Active Directory schema

Extending the Active Directory schema adds and updates classes, attributes, and other items. These changes are needed so that Exchange can create containers and objects to store information about the Exchange organization. Because Exchange makes a lot of changes to the Active Directory schema, there's a topic dedicated to this step. To see all of the changes made to the schema, see Exchange 2013 Active Directory schema changes.

This step is done automatically when you run the Exchange 2013 Setup wizard on the first Exchange 2013 server in the Active Directory forest. It's also done when you run Exchange 2013 command line Setup with the *PrepareSchema* command (or optionally with the *PrepareAD* command) on the first Exchange 2013 server in the forest. If you want to find out more information about how to extend the schema, see Extend the Active Directory schema in Prepare Active Directory and domains.

After Exchange is finished extending the schema, it sets the schema version, which is stored in the **ms-Exch-Schema-Version-Pt** attribute. If you want to make sure that the Active Directory schema was extended successfully, you can check the value stored in this attribute. If the value in the attribute matches the schema version listed for the release of Exchange 2013 you installed, extending the schema was successful. For a list of Exchange releases and how to check the value of this attribute, check out the How do you know this worked? section in Prepare Active Directory and domains.

Prepare Active Directory containers, objects, and other items

With the schema extended, the next step is to add all of the containers, objects, attributes, and other items that Exchange uses to store information in Active Directory. Most of the changes made in this step are applied to the entire Active Directory forest. A smaller set of changes are made to the local Active Directory domain where the *PrepareAD* command was run during Setup.

These are the changes that are made to the Active Directory forest:

- The Microsoft Exchange container is created under CN=Services,CN=Configuration,DC=<*root domain*> if it doesn't already exist.
- The following containers and objects are created under CN=<*organization name*>,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=<*root domain*> if they don't already exist:
  - CN=Address Lists Container
  - CN=AddressBook Mailbox Policies
  - CN=Addressing
  - CN=Administrative Groups
  - CN=Approval Applications
  - CN=Auth Configuration
  - CN=Availability Configuration
  - CN=Client Access
  - CN=Connections
  - CN=ELC Folders Container
  - CN=ELC Mailbox Policies
  - CN=ExchangeAssistance
  - CN=Federation
  - CN=Federation Trusts
  - CN=Global Settings

    - CN=Hybrid Configuration
    - CN=Mobile Mailbox Policies
    - CN=Mobile Mailbox Settings
    - CN=Monitoring Settings
    - CN=OWA Mailbox Policies
    - CN=Provisioning Policy Container
    - CN=Push Notification Settings
    - CN=RBAC
    - CN=Recipient Policies
    - CN=Remote Accounts Policies Container
    - CN=Retention Policies Container
    - CN=Retention Policy Tag Container
    - CN=ServiceEndpoints
    - CN=System Policies
    - CN=Team Mailbox Provisioning Policies
    - CN=Transport Settings
    - CN=UM AutoAttendant Container
    - CN=UM DialPlan Container
    - CN=UM IPGateway Container
    - CN=UM Mailbox Policies
    - CN=Workload Management Settings
- The following containers and objects are created under CN=Transport Settings,CN=<*Organization Name*>,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=<*root domain*> if they don't already exist:
    - CN=Accepted Domains
    - CN=ControlPoint Config
    - CN=DNS Customization
    - CN=Interceptor Rules
    - CN=Malware Filter
    - CN=Message Classifications
    - CN=Message Hygiene
    - CN=Rules
    - CN=MicrosoftExchange329e71ec88ae4615bbc36ab6ce41109e
- Permissions are set throughout the configuration partition in Active Directory.
- The Rights.ldf file is imported. This file adds permissions that are needed to install Exchange and configure Active Directory.
- The Microsoft Exchange Security Groups organizational unit (OU) is created in the root domain of the forest, and permissions are assigned to it.
- The following management role groups are created within the Microsoft Exchange Security Groups OU if they don't already exist:
    - Compliance Management
    - Delegated Setup

    - Discovery Management
    - Help Desk
    - Hygiene Management
    - Organization Management
    - Public Folder Management
    - Recipient Management
    - Records Management
    - Server Management
    - UM Management
    - View-Only Organization Management
- The new management role groups (which appear as universal security groups (USGs) in Active Directory) that were created in the Microsoft Exchange Security Groups OU are added to the **otherWellKnownObjects** attribute stored on the CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=<*root domain*> container.
- The Unified Messaging Voice Originator contact is created in the Microsoft Exchange System Objects container of the root domain.
- The domain where the *PrepareAD* command was run is prepared for Exchange 2013. For information about what's done to prepare the Active Directory domain for Exchange, check out Preparing Active Directory domains.
- The **msExchProductId** property on the Exchange organization object is set. If you want to make sure that the Active Directory schema was extended successfully, you can check the value stored in this property. If the value in the property matches the schema version listed for the release of Exchange 2013 you installed, extending the schema was successful. For a list of Exchange releases and how to check the value of this property, check out the How do you know this worked? section in Prepare Active Directory and domains.

Prepare Active Directory domains

The final step of preparing Active Directory for Exchange is to prepare all of the Active Directory domains where Exchange servers will be installed or where mailbox-enabled users will be located. This step is done automatically in the domain where the *PrepareAD* command was run.

These are the changes that are made to the Active Directory domains:

- The Microsoft Exchange System Objects container is created in the root domain partition in Active Directory if it doesn't already exist.
- Permissions are set on the Microsoft Exchange System Objects container for the Exchange Servers, Organization Management, and Authenticated Users security groups.
- The Exchange Install Domain Servers domain global group is created in the current domain and placed in the Microsoft Exchange System Objects container.
- The Exchange Install Domain Servers group is added to the Exchange Servers USG in the root domain.
- Permissions are assigned at the domain level for the Exchange Servers USG and the Organization Management USG.

- The **objectVersion** property in the Microsoft Exchange System Objects container under DC=<*root domain*> is set. If you want to make sure that the Active Directory schema was extended successfully, you can check the value stored in this property. If the value in the property matches the schema version listed for the release of Exchange 2013 you installed, extending the schema was successful. For a list of Exchange releases and how to check the value of this property, check out the How do you know this worked? section in Prepare Active Directory and domains.

Exchange 2013: editions and versions

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-24*

Microsoft Exchange Server 2013 is available in two server editions: Standard Edition and Enterprise Edition. Enterprise Edition can scale to 50 mounted databases per server in the Release to Manufacturing (RTM) and Cumulative Update 1 (CU1) versions, and 100 mounted databases per server in Cumulative Update 2 (CU2) and later versions; Standard Edition is limited to 5 mounted databases per server. A mounted database is a database that is in use. A mounted database can be an active mailbox database that is mounted for use by clients, or a passive mailbox database that is mounted in recovery for log replication and replay. While you can create more databases than the limits described above, you can only mount the maximum number specified above. The recovery database does not count towards this limit.

These licensing editions are defined by a product key. When you enter a valid license product key, the supported edition for the server is established. Product keys can be used for the same edition key swaps and upgrades only; they can't be used for downgrades. You can use a valid product key to move from the evaluation version (Trial Edition) of Exchange 2013 to either Standard Edition or Enterprise Edition. You can also use a valid product key to move from Standard Edition to Enterprise Edition.

You can also license the server again using the same edition product key. For example, if you had two Standard Edition servers with two keys, but you accidentally used the same key on both servers, you can change the key for one of them to the other key that you were issued. You can take these actions without having to reinstall or reconfigure anything. After you enter the product key and restart the Microsoft Exchange Information Store service, the edition corresponding to that product key will be reflected.

No loss of functionality will occur when the Trial Edition expires, so you can maintain lab, demo, training, and other non-production environments beyond 120 days without having to reinstall the Trial Edition of Exchange 2013.

As mentioned earlier, you can't use product keys to downgrade from Enterprise Edition to Standard Edition, nor can you use them to revert to the Trial Edition. These types of downgrades can only be

done by uninstalling Exchange 2013, reinstalling Exchange 2013, and entering the correct product key.

Exchange 2013 versions

For a list of Exchange 2013 versions and information on how to download and upgrade to the latest version of Exchange 2013, see the following topics:

- Exchange Server Updates: build numbers and release dates
- Upgrade Exchange 2013 to the latest cumulative update or service pack

To view the build number for the version of Exchange 2013 that you're running, run the following command in the Exchange Management Shell.

```
Get-ExchangeServer | fl name,edition,admindisplayversion
```

Exchange 2013 license types

Exchange 2013 is licensed in the Server/Client Access License (CAL) model similar to how Exchange 2010 was licensed. Following are the types of licenses:

- **Server licenses** A license must be assigned for each instance of the server software that is being run. The Server license is sold in two server editions: Standard Edition and Enterprise Edition.
- **Client Access licenses (CALs)** Exchange 2013 also comes in two client access license (CAL) editions, which are referred to as a Standard CAL and an Enterprise CAL. You can mix and match the server editions with the CAL types. For example, you can use Enterprise CALs with Exchange 2013 Standard Edition. Similarly, you can use Standard CALs with Exchange 2013 Enterprise Edition.

For more information about Exchange license types, see Licensing.

Exchange Server Supportability Matrix

Applies to: *Exchange Server 2013, Exchange Server 2010, Exchange Server 2007*

Topic Last Modified: *2014-08-26*

The Exchange Server Supportability Matrix provides a central source for Microsoft Exchange administrators to easily locate information about the level of support available for any configuration or required component for supported versions of Microsoft Exchange.

Support lifecycle

For more information about the support lifecycle for a specific version of Exchange, or of the Microsoft Windows server or client operating systems, see the Microsoft Support Lifecycle page. For more information about the Microsoft Support Lifecycle, see the Microsoft Support Lifecycle Policy FAQ.

Exchange Server 2003 End-of-life

As of April 8, 2014, Microsoft no longer provides security updates, offers free or paid support options, or updated online content such as KB articles for Exchange 2003. Online content may remain available as long as Exchange 2003 remains in the Self-Help online support phase.

Companies running Exchange 2003 after April 8, 2014 will be responsible for their own for support. More importantly, because Microsoft will no longer provide security updates, companies that choose to continue running Exchange 2003 accept the risk associated with that.

Release model

The following table identifies the release model for each supported version of Exchange. The release model is identified by an X character.

For versions of Exchange prior to Exchange 2013, each update rollup package is cumulative with regard to the whole product. Therefore, if you apply an update rollup package to Exchange Server 2010, you apply all the fixes contained in that update rollup package. This includes all the fixes contained in each earlier update rollup package. When an update or a hotfix for earlier versions of Exchange is created, one or more of the binary files included in the update or included in the hotfix are cumulative. They are cumulative with regard to the contents of the files. However, they aren't cumulative with regard to the whole Exchange product. For more information, see Exchange 2010 Servicing.

With Exchange 2013, we changed the way we deliver hotfixes and service packs. Instead of the priority-driven hotfix release and update rollup model used by previous versions of Exchange, Exchange 2013 now adheres to a scheduled delivery model. In this model, cumulative updates are released every three months. A cumulative update (CU) for Exchange 2013 is released as a full refresh of Exchange 2013, similar to a product upgrade or a service pack release. For more information about the new servicing model, see Updates for Exchange 2013.

Servicing release model	Exchange 2013	Exchange 2010	Exchange 2007	
Cumulative updates	X			
Update rollups		X	X	
Security Hotfixes	X			

delivered separately				

Supported operating system platforms

The following table identifies the operating system platforms on which each version of Exchange can run. Supported platforms are identified by an X character.

Operating system platform	Exchange 2013 SP1 and later	Exchange 2013 CU3	Exchange 2010 SP3	Exchange 2010 SP2	Exchange 2007 SP3
Windows Vista SP2			X[1]	X[1]	X[1]
Windows Server 2003 SP2					X
Windows Server 2003 R2 SP2					X
Windows Server 2008 SP2			X	X	X
Windows Server 2008 R2 SP1	X	X	X	X	X
Windows 7 SP1	X[1]	X[1]	X[1]	X[1]	X[1]
Windows 8	X[1]	X[1]	X[1]		
Windows 8.1	X[1]				
Windows Server 2012	X	X	X		

Windows Server 2012 R2	X				

[1]Only for Exchange management tools

Supported Active Directory environments

The following table identifies the Active Directory environments with which each version of Exchange can communicate. Supported environments are identified by an X character. An Active Directory server refers to both writable global catalog servers and to writable domain controllers. Read-only global catalog servers and read-only domain controllers aren't supported.

Operating system environment	Exchange 2013 SP1 and later	Exchange 2013 CU3	Exchange 2010 SP3 RU5 or later	Exchange 2010 SP2	Exchange 2007 SP3 RU13 or later	
Windows Server 2003 SP1 Active Directory servers					X	
Windows Server 2003 SP2 Active Directory servers	X	X	X	X	X	
Windows Server 2008 SP2 Active Directory servers	X	X	X	X	X	
Windows Server 2008 R2 SP1 Active	X	X	X	X	X	

Directory servers						
Windows Server 2012 Active Directory servers	X	X	X	X	X	
Windows Server 2012 R2 Active Directory servers	X	X	X		X	

Domain and forest functional level	**Exchange 2013 SP1 and later**	**Exchange 2013 CU3**	**Exchange 2010 SP3 RU5 or later**	**Exchange 2010 SP2**	**Exchange 2007 SP3 RU13 or later**	
Windows Server 2003 domain functional level	X	X	X	X	X	
Windows Server 2008 R2 SP1 domain functional level	X	X	X	X	X	
Windows Server 2012 domain functional	X	X	X	X	X	

level						
Windows Server 2012 R2 domain functional level	X		X			
Windows Server 2003 forest functional level	X	X	X	X	X	
Windows Server 2008 R2 SP1 forest functional level	X	X	X	X	X	
Windows Server 2012 forest functional level	X	X	X	X	X	
Windows Server 2012 R2 forest functional level	X		X			

Web browsers supported for use with the premium version of Outlook Web App or Outlook Web Access

The following table identifies the Web browsers supported for use together with the premium version of Microsoft Office Outlook Web App for Exchange 2013, Microsoft Exchange Server 2010 and Office Outlook Web Access for Exchange 2007. Supported browsers are identified by an X character.

Browser	Exchange 2013 CU3 or later	Exchange 2010 SP3	Exchange 2010 SP2	Exchange 2007 SP3	
Internet Explorer 11	X	X			
Internet Explorer 10	X	X			
Internet Explorer 9	X	X	X	X	
Internet Explorer 8	X	X	X	X	
Internet Explorer 7		X	X	X	
Firefox 3.0.1 or later		X	X		
Firefox 12 or later	X	X	X		
Safari 3.1 or later		X	X		
Safari 5.0 or later	X	X	X		
Chrome 3.0.195 or later		X	X		
Chrome 18 or later	X	X	X		

Web browsers supported for use with the basic version of Outlook Web App or Outlook Web Access

The following table identifies the Web browsers that are supported for use together with the light (basic) version of Outlook Web App for Exchange 2013, Exchange 2010 or Microsoft Outlook Web Access for Microsoft Exchange Server 2007. Supported browsers are identified by an X character.

Note:

Outlook Web App Basic (Outlook Web App Light) is supported for use in mobile browsers. However, if rendering or authentication issues occur in a mobile browser, determine whether the issue can be reproduced by using Outlook Web App Light in the full client of a supported browser. For example, test the use of Outlook Web App Light in Safari, Chrome, or Internet Explorer. If the issue can't be reproduced in the full client, we recommend that you contact the mobile device vendor for help. In these cases, we collaborate with the vendor as appropriate.

Browser	Exchange 2013 CU3 or later	Exchange 2010 SP3	Exchange 2010 SP2	Exchange 2007 SP3	
Internet Explorer 11	X	X			
Internet Explorer 10	X	X			
Internet Explorer 9	X	X[1]	X[1]	X	
Internet Explorer 8	X	X	X	X	
Internet Explorer 7	X	X	X	X	
Safari	X	X	X	X	
Firefox	X	X	X	X	
Netscape				X	
Opera	X	X	X	X	

[1]Requires Exchange 2010 SP2 RU5-v2, described in Microsoft Knowledge Base article 2785908, Description of Update Rollup 5 version 2 for Exchange Server 2010 Service Pack 2.

Web browsers supported for use of S/MIME with Outlook Web App or Outlook Web Access

The following table identifies the Web browsers that are supported for the use of S/MIME together with Outlook Web App for Exchange 2013, Exchange 2010 or Outlook Web Access for Exchange 2007. Supported browsers are identified by an X character.

Browser	Exchange 2013 CU3 and later	Exchange 2010 SP3	Exchange 2010 SP2	Exchange 2007 SP3	
Internet Explorer 11	X	X			
Internet Explorer 10	X	X			
Internet Explorer 9	X	X	X	X	
Internet Explorer 8		X	X	X	
Internet Explorer 7		X	X	X	

Clients

The following table identifies the mailbox clients that are supported for use together with each version of Exchange. Supported clients are identified by an X character.

Client	Exchange 2013 CU3 or later	Exchange 2010 SP3	Exchange 2010 SP2	Exchange 2007 SP3
Outlook 2003		X	X	X
Outlook 2007	X[2]	X	X	X

Outlook 2010	X[3]	X	X	X
Outlook 2013	X	X	X	X
Windows Mobile 5.0	X	X	X	X
Windows Mobile 6.0	X	X	X	X
Windows Mobile 6.1	X	X	X	X
Windows Mobile 6.5	X	X	X	X
Windows Phone 7	X	X	X	X
Windows Phone 7.5	X	X	X	X
Windows Phone 8	X	X	X	X
Entourage X				X[1]
Entourage 2004 (DAV)				X[2]
Entourage 2008 (DAV)				X[2]
Entourage 2008 (EWS)	X[4]	X[4]	X[4]	X

[1]WebDav: Contacts, Events, IMAP: Mail

[2]Requires Outlook 2007 Service Pack 3 and the November 2012 Public Update or later.

[3]Requires Outlook 2010 Service Pack 1 and the November 2012 Public Update or later.

[4]EWS only. There is no DAV support for Exchange 2010.

Tools

The following table identifies the version of Microsoft Exchange that can be used together with the Microsoft Exchange Inter-Organization Replication tool (Exscfg.exe; Exssrv.exe). The tool is used to replicate public folder information (including free/busy information) between Exchange organizations. For more information, see Microsoft Exchange Server Inter-Organization Replication. Supported versions are identified by an X character.

Tool	Exchange 2013 CU3 or later	Exchange 2010 SP3	Exchange 2010 SP2	Exchange 2007 SP3	
Inter-Organization Replication tool		X	X	X	

Microsoft .NET Framework

The following table identifies the version of the Microsoft .NET Framework that can be used together with each version of Exchange. Supported versions are identified by an X character.

.NET Framework	Exchange 2013 SP1 or later	Exchange 2013 CU3	Exchange 2010 SP3	Exchange 2010 SP2	Exchange 2007 SP3
.NET Framework 2.0 SP1					X
.NET Framework 3.0					X
.NET Framework 3.5			X[5]		X[3]
.NET Framework 3.5 SP1			X	X	X
.NET Framework 4.0			X[5]	X[4]	

.NET Framework 4.5	X	X	X[5]	X[4]	
.NET Framework 4.5.1	X				

[3]Supported versions of the .NET Framework are included in the .NET Framework 3.5 and in the .NET Framework 3.5 SP1.

[4]Applies only when upgrading the system from the .NET Framework 3.5 and the .NET Framework 3.5 SP1. Uninstalling the .NET Framework 3.5 and the .NET Framework 3.5 SP1 isn't supported.

[5] If you are using Windows Server 2012, the .NET Framework 3.5 must be installed before you can use Exchange 2010 SP3.

Windows Management Framework

The following table identifies the version of the Windows Management Framework, which contains the Windows PowerShell command-line interface, that can be used together with each version of Exchange. Supported versions are identified by an X character.

Windows PowerShell	Exchange 2013 SP1 and later	Exchange 2013 CU3	Exchange 2010 SP3	Exchange 2010 SP2	Exchange 2007 SP3	
Windows PowerShell 1.0					X	
Windows PowerShell 2.0			X	X	X	
Windows Management Framework 3.0	X	X				
Windows Management	X		X[6]			

Framework 4.0						

[6] Requires Exchange Server 2010 SP3 RU5 or later.

Microsoft Management Console

The following table identifies the version of Microsoft Management Console (MMC) that can be used together with each version of Exchange. Supported versions are identified by an X character.

MMC	Exchange 2013 CU3 or later	Exchange 2010 SP3	Exchange 2010 SP2	Exchange 2007 SP3		
MMC 3.0	X		X	X	X	

Windows Installer

The following table identifies the version of Windows Installer that is used together with each version of Exchange. Supported versions are identified by an X character.

Windows Installer	Exchange 2013 CU3 or later	Exchange 2010 SP3	Exchange 2010 SP2	Exchange 2007 SP3
Windows Installer 4.5	X	X	X	X
Windows Installer 5.0	X			

Exchange 2013 deployment permissions reference

Exchange Server 2013 > Planning and deployment > Deployment reference >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-19*

This topic describes the permissions that are required to set up a Microsoft Exchange Server 2013 organization. The universal security groups (USGs) that are associated with management role groups, and other Windows security groups and security principals, are added to the access control

lists (ACLs) of various Active Directory objects. ACLs control what operations can be performed on each object. By understanding what permissions are granted to each role group, security group, or security principal, you can determine what minimum permissions are required to install Exchange 2013.

In some cases, the ACL isn't applied on the usual property, **ntSecurityDescriptor**, but on another property, such as **msExchMailboxSecurityDescriptor**. The directory service can't enforce security that isn't specified in the Windows security descriptor. In most cases, these ACLs are replicated to store ACLs on appropriate objects by the store service. Unfortunately, there is no tool to view these ACLs as anything other than raw binary data.

The columns of each permissions table include the following information:

- **Account** The security principal granted or denied the permissions.
- **ACE type** Access control entry (ACE) type
  - **Allow ACE** An allow ACE allows the user or group associated with the ACE to access an item.
  - **Deny ACE** A deny ACE prevents the user or group associated with the ACE from accessing an item.
- **Inheritance** The type of inheritance used for child objects.
  - **All** indicates that the permissions apply to the object and all sub-objects.
  - **Desc** indicates the permissions apply to the object class listed in the On Property/Applies To row.
  - **None** indicates those permissions only apply the object.
- **Permissions** The permissions granted to the account.
- **On Property/Applies To** In some cases, permissions apply only to a given property, property set, or object class. These limited permissions are specified here.
- **Comments** When applicable, this column explains why the permissions are required or provides other information about the permissions.

The permissions are generally listed in the table by the names that are used on the Active Directory Service Interfaces (ADSI) Edit (AdsiEdit.msc) **Security** property page in the **Advanced** view on the **View/Edit** tab. The ADSI Edit **Security** property page lists a much more condensed view of the permissions. The LDP tool (Ldp.exe) displays the access mask directly as a numeric value. The setup code refers to the permissions by predefined constants.

The following table shows the relationships between these values.

ADSI Edit Summary page	ADSI Edit Advanced view, View/Edit tab	ACL entries applied to a given object	Binary value (access mask in LDP)
Full Control	Full Control	`WRITE_OWNER | WRITE_DAC | READ_CONTROL | DELETE | ACTRL_DS_CONTROL_ACCESS | ACTRL_DS_LIST_OBJECT | ACTRL_DS_DELETE_TREE | ACTRL_DS_WRITE_PROP | ACTRL_DS_READ_PROP`	0x000F01FF

		| ACTRL_DS_SELF | ACTRL_DS_LIST | ACTRL_DS_DELETE_CHILD | ACTRL_DS_CREATE_CHILD	
Read	List Contents + Read All Properties + Read Permissions	ACTRL_DS_LIST | ACTRL_DS_READ_PROP | READ_CONTROL	0x00020014
Write	Write All Properties + All Validated Writes	ACTRL_DS_WRITE_PROP | ACTRL_DS_SELF	0x00000028
	List Contents	ACTRL_DS_LIST	0x00000004
	Read All Properties	ACTRL_DS_READ_PROP	0x00000010
	Write All Properties	ACTRL_DS_WRITE_PROP	0x00000020
	Delete	DELETE	0x00010000
	Delete Subtree	ACTRL_DS_DELETE_TREE	0x00000040
	Read Permissions	READ_CONTROL	0x00020000
	Modify Permissions	WRITE_DAC	0x00040000
	Modify Owner	WRITE_OWNER	0x00080000
	All Validated Writes	ACTRL_DS_SELF	0x00000008
	All Extended Rights	ACTRL_DS_CONTROL_ACCESS	0x00000100
Create All Child Objects	Create All Child Objects	ACTRL_DS_CREATE_CHILD	0x00000001
Delete All Child Objects	Delete All Child Objects	ACTRL_DS_DELETE_CHILD	0x00000002
		ACTRL_DS_LIST_OBJECT	0x00000080

Extended rights are custom rights specified by individual applications. They are specified in the ACL. However, they are meaningless to Active Directory. The specific application enforces any extended rights. Examples of Exchange extended rights are "Create public folder" or "Create named properties in the information store."

For information about permissions that are set during a Microsoft Exchange Server 2010

installation, see Exchange 2010 Deployment Permissions Reference.

Prepare Active Directory Permissions

The permissions tables in this section show the permissions set when you execute the `Setup /PrepareAD` command.

Note:

The permissions described in this section are the default permissions that are configured when you deploy Exchange 2013 using the shared permissions model. If you've deployed Exchange 2013 using the Active Directory split permissions model, the default permission are different. For more information on the changes to the default permissions when using Active Directory split permissions and the shared and split permissions models in general, see Active Directory split permissions in Understanding split permissions. If you don't choose to use Active Directory split permissions when you install Exchange, Exchange will use shared permissions.

Microsoft Exchange Container Permissions

The following table shows the permissions that are set on the Microsoft Exchange container within the configuration partition.

Distinguished name of the object: CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=<domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
Installation Account	Allow ACE	All	Full Control		This is the account that is used to run `/PrepareAD`.
Organization Management	Allow ACE	All	Full Control		
Exchange Trusted Subsystem	Allow ACE	All	Full Control		
Exchange Servers	Allow ACE	All	Read		
Authenticated	Allow ACE	None	Read Property		

Users			List Contents		
Exchange Trusted Subsystem	Allow ACE	All	Modify Permissions	msExchSmtpReceiveConnector	
Public Folder Management	Allow ACE	All	Read List Object		
Delegated Setup	Allow ACE	All	Read List Object		

Microsoft Exchange Autodiscover Container Permissions

The following table shows the permissions set on the Microsoft Exchange Autodiscover container within the configuration partition.

Distinguished name of the object: CN=Microsoft Exchange Autodiscover,CN=Services,CN=Configuration,DC=<domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Exchange Servers	Allow ACE	All	Read	

Microsoft Exchange Organization Container Permissions

The permissions tables in this section show the permissions set on the Microsoft Exchange Organization and sub-containers within the configuration partition.

Distinguished name of the object: CN=<organization>,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=<domain>

Account(s)	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
Enterprise Admins Root Domain Admins Installation	Deny ACE	All	Send As Receive As		Windows administrators aren't allowed to open mailboxes.

Account Organization Management					
Enterprise Admins Schema Admins Root Domain Admins Installation Account Organization Management	Deny ACE	All	Exchange Web Services Impersonation Exchange Web Services Token Serialization		Extended right
Enterprise Admins Schema Admins Root Domain Admins Installation Account	Deny ACE	All	Store Transport Access Store Constrained Delegation Store Read Access Store Read Write Access		
Local System	Allow	All	All Extended Rights		
Authenticated Users	Deny ACE	Desc	Read Property	`msExchAvailabilityUserPassword / msExchAvailabilityAddressSpace`	
Authenticated	Allow	None	Read		

Users					
Organization Management	Allow ACE	All	Read Permissions List Contents Read Property List Object		
Public Folder Management	Allow ACE	All	Read Permissions List Contents Read Property List Object		
NT Authority \Network Service	Allow ACE	All	Read		
Managed Availability Servers	Allow ACE	All	Read Permissions List Contents Read Property List Object		
Exchange Servers	Allow ACE	All	All Extended Rights		
Exchange Servers	Allow ACE	All	Write Property	`groupType`	
Exchange Servers	Allow ACE	All	Write Property	`msExchOwningServer`	
Exchange	Allow ACE	All	Write Property	`msExchMailboxSecurityDescriptor`	

Servers					
Exchange Servers	Allow ACE	All	Write Property	msExchUMServerWritableFlags	
Exchange Servers	Allow ACE	All	Write Property	msExchDatabaseCreated	
Exchange Servers	Allow ACE	All	Write Property	msExchUserCulture	
Exchange Servers	Allow ACE	All	Write Property	msExchMobileMailboxFlags	
Exchange Servers	Allow ACE	All	Write Property	siteFolderGUID	
Exchange Servers	Allow ACE	All	Write Property	siteFolderServer	
Exchange Servers	Allow ACE	All	Write Property	msExchEDBOffline	
Exchange Servers	Allow ACE	All	Write Property	userCertificate	
Exchange Servers	Allow ACE	All	Write Property	msExchUMDtmfMap	
Exchange Servers	Allow ACE	All	Write Property	msExchBlockedSendersHash	
Exchange Servers	Allow ACE	All	Write Property	Personal Information	
Exchange Servers	Allow ACE	All	Write Property	Public Information	
Exchange	Allow ACE	All	Write Property	Exchange Information	

Servers					
Exchange Servers	Allow ACE	All	Write Property	msExchPatchMDB	
Exchange Servers	Allow ACE	All	Write Property	publicDelegates	
Exchange Servers	Allow ACE	All	Write Property	msExchUMSpokenName	
Exchange Servers	Allow ACE	All	Write Property	msExchUMPinChecksum	
Exchange Servers	Allow ACE	All	Write Property	legacyExchangeDN	
Exchange Servers	Allow ACE	All	Write Property	msExchSafeSendersHash	
Exchange Servers	Allow ACE	All	Write Property	thumbnailPhoto	
Organization Management	Allow ACE	All	Create top level public folder		
Public Folder Management	Allow ACE	All	Create top level public folder		
Organization Management	Allow ACE	All	View information store status		
Public Folder Management	Allow ACE	All	View information store status		
Organization Management	Allow ACE	All	Administer information		

			store		
Public Folder Management	Allow ACE	All	Administer information store		
Organization Management	Allow ACE	All	Create named properties in the information store		
Public Folder Management	Allow ACE	All	Create named properties in the information store		
Organization Management	Allow ACE	All	Modify public folder ACL		
Public Folder Management	Allow ACE	All	Modify public folder ACL		
Organization Management	Allow ACE	All	Modify public folder quotas		
Public Folder Management	Allow ACE	All	Modify public folder quotas		
Organization Management	Allow ACE	All	Modify public folder admin ACL		
Public Folder Management	Allow ACE	All	Modify public folder admin ACL		
Organization Management	Allow ACE	All	Modify public folder expiry		

Public Folder Management	Allow ACE	All	Modify public folder expiry		
Organization Management	Allow ACE	All	Modify public folder replica list		
Public Folder Management	Allow ACE	All	Modify public folder replica list		
Organization Management	Allow ACE	All	Modify public folder deleted item retention		
Public Folder Management	Allow ACE	All	Modify public folder deleted item retention		
Organization Management	Allow ACE	All	Create public folder		
Public Folder Management	Allow ACE	All	Create public folder		
Public Folder Management	Allow ACE	All	Mail Enable Public Folder		
Everyone NT Authority \Anonymous Logon	Allow ACE	All	Create named properties in the information store		
Everyone NT Authority \Anonymous Logon	Allow ACE	All	Create public folder		

Everyone NT Authority\Anonymous Logon	Allow ACE	Desc	Read Permissions List Contents Read Property List Object	/msExchPrivateMDB	
Everyone NT Authority\Anonymous Logon	Allow ACE	Desc	Read Permissions List Contents Read Property List Object	/msExchPublicMDB	
Exchange Servers	Allow ACE	Desc	Read Permissions List Contents Read Property List Object	/siteAddressing	

Distinguished name of the object: CN=All Address Lists,CN=Address Lists Container,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Authenticated Users	Allow ACE	All	List Contents	
Organization Management	Allow ACE	All	Write Property	msExchLastAppliedRecipientFilter msExchRecipientFilterFlags
Public Folder Management	Allow ACE	All	Write Property	msExchLastAppliedRecipientFilter msExchRecipientFilterFlags

Distinguished name of the object: CN=Offline Address Lists,CN=Address Lists Container, CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Authenticated Users	Allow ACE	All	Download Offline Address Book	

Distinguished name of the object: CN=Addressing,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Authenticated users	Allow ACE	All	Read	

Distinguished name of the object: CN=Recipient Policies,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Organization Management	Allow ACE	All	Write Property	`msExchLastAppliedRecipientFilter` `msExchRecipientFilterFlags`
Public Folder Management	Allow ACE	All	Write Property	`msExchLastAppliedRecipientFilter` `msExchRecipientFilterFlags`

Configuration Partition Container Permissions

The permissions tables in this section show the permissions set by the `Setup /PrepareAD` command on various containers within the configuration partition.

Distinguished name of the object: CN=Sites,CN=Configuration,DC=<domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Organization Management Exchange Trusted Subsystem	Allow ACE	All	Write Property	`msExchVersion / site`
Organization Management Exchange Trusted	Allow ACE	All	Write Property	`msExchVersion / site-link`

Subsystem				
Organization Management Exchange Trusted Subsystem	Allow ACE	All	Write Property	msExchPartnerId / site
Organization Management Exchange Trusted Subsystem	Allow ACE	All	Write Property	msExchMinorPartnerId / site
Organization Management Exchange Trusted Subsystem	Allow ACE	All	Write Property	msExchResponsibleforSites / site
Organization Management Exchange Trusted Subsystem	Allow ACE		Write Property	msExchTransportSiteFlags / site
Organization Management Exchange Trusted Subsystem	Allow ACE	All	Write Property	msExchCost / site-link
Organization Management Exchange Trusted Subsystem Local System Exchange Servers	Allow ACE	Desc	Read Permissions List Contents Read Property List Object	/ msExchEdgeSyncEHFConnector

Organization Management Exchange Trusted Subsystem Local System Exchange Servers	Allow ACE	Desc	Read Permissions List Contents Read Property List Object	/msExchEdgeSyncMservConnector
Organization Management Exchange Trusted Subsystem	Allow ACE	Children	Create Child Delete Child Delete Tree	msExchEdgeSyncServiceConfig / site
Organization Management Exchange Trusted Subsystem Local System Exchange Servers	Allow ACE	Desc	Read Permissions List Contents Read Property List Object	/msExchEdgeSyncServiceConfig
Organization Management Exchange Trusted Subsystem	Allow ACE	Children	Create Child Delete Child Delete Tree	msExchEdgeSyncMservConnector / msExchEdgeSyncServiceConfig
Organization Management Exchange Trusted Subsystem	Allow ACE	Children	Create Child Delete Child Delete Tree	msExchEdgeSyncEHFConnector / msExchEdgeSyncServiceConfig

Distinguished name of the object: CN=Deleted Objects,CN=Configuration,DC=<domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
Exchange	Allow ACE	All	List Contents		

Servers					
Organization Administration	Allow ACE	All	Read List Object		
Installation Account	Allow ACE	All	Read Permission Write Permission List Contents Read Property List Object		This is the account that is used to run `/PrepareAD`.
Exchange Trusted Subsystem	Allow ACE	All	Read List Object		
Network Service	Allow ACE	All	List Contents		

Exchange Administrative Group Permissions

The `Setup /PrepareAD` command also configures the following permissions on the administrative groups within the organization.

Distinguished name of the object: CN=<admin group>,CN=Administrative Groups,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
Organization Management	Allow ACE	Desc	Access Recipient Update Service	`msExchExchangeServer`	Allows Exchange Recipient Administrators to stamp recipients with

					proxy address information.
Local System	Allow ACE	Desc	Access Recipient Update Service	`msExchExchangeServer`	Allows the servers to stamp recipients with proxy address information.
Public Folder Management	Allow ACE	Desc	Access Recipient Update Service	`msExchExchangeServer`	Allows Exchange Public Folder Administrators to stamp recipients with proxy address information.

Distinguished name of the object: CN=Advanced Security Settings,CN=<admin group>,CN=Administrative Groups,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Authenticated Users	Allow ACE	None	List Contents	

Distinguished name of the object: CN=Encryption,CN=Advanced Security Settings,CN=<admin group>,CN=Administrative Groups,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Authenticated Users	Allow ACE	None	Read Property	

Distinguished name of the object: CN=Arrays,CN=<admin group>,CN=Administrative Groups,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/

				Applies to
Authenticated Users	Allow ACE	None	List Contents	

Distinguished name of the object: CN=Database Availability Groups,CN=<admin group>,CN=Administrative Groups,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Authenticated Users	Allow ACE	None	List Contents	

Distinguished name of the object: CN=Databases,CN=<admin group>,CN=Administrative Groups,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Authenticated Users	Allow ACE	None	List Contents	

Distinguished name of the object: CN=Servers,CN=<admin group>,CN=Administrative Groups,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
Exchange Servers	Deny ACE	All	Receive As		Exchange Servers aren't allowed to open mailboxes.
Authenticated Users	Allow ACE	None	List Contents		

Microsoft Exchange Security Groups Container Permissions

The permissions tables in this section show the permissions set on the Microsoft Exchange Security Groups container within the root domain partition.

Distinguished name of the object: OU=Microsoft Exchange Security Groups,DC=<root

domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Organization Management	Allow ACE	All	Full Control	
Exchange Trusted Subsystem	Allow ACE	All	Create Child	/ Group
Exchange Trusted Subsystem	Allow ACE	Desc	Delete	/ group
Exchange Trusted Subsystem	Allow ACE	Desc	Write Property	Member / group

Distinguished name of the object: CN=Organization Management,OU=Microsoft Exchange Security Groups,DC=<root domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Organization Management	Allow ACE	All	Full Control	

Distinguished name of the object: CN=Public Folder Management,OU=Microsoft Exchange Security Groups,DC=<root domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Organization Management	Allow ACE	All	Full Control	

Distinguished name of the object: CN=ExchangeLegacyInterop,OU=Microsoft Exchange Security Groups,DC=<root domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Organization Management	Allow ACE	All	Full Control	

Distinguished name of the object: CN=Exchange Servers,OU=Microsoft Exchange Security

Groups,DC=<root domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Organization Management	Allow ACE	All	Full Control	
Root Domain Administrators	Allow ACE	All	Read Members Write Members	
Child Domain Administrators	Allow ACE	All	Read Members Write Members	

Prepare Domain

The following tables show the permissions set when you execute the `Setup /PrepareDomain` command.

Note:

The permissions described in this section are the default permissions that are configured when you deploy Exchange 2013 using the shared permissions model. If you've deployed Exchange 2013 using the Active Directory split permissions model, the default permission are different. For more information on the changes to the default permissions when using Active Directory split permissions and the shared and split permissions models in general, see Active Directory split permissions in Understanding split permissions. If you don't choose to use Active Directory split permissions when you install Exchange, Exchange will use shared permissions.

Distinguished name of the object: DC=<domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
Authenticated Users	Allow ACE	All	Read Property	`Exchange Information`	
NT AUTHORITY \NETWORK	Allow ACE	All	Read Property	`Exchange Personal Information`	Grants Transport service read permissions.
Exchange Servers	Allow ACE	All	Write Property	`groupType`	

Exchange Servers	Allow ACE	All	Write Property	msExchMailboxSecurityDescriptor	
Exchange Servers	Allow ACE	All	Write Property	msExchUMServerWritableFlags	
Exchange Servers	Allow ACE	All	Read Property	Exchange Personal Information	
Exchange Servers	Allow ACE	All	Read Property	Exchange Information	
Exchange Servers	Allow ACE	All	Write Property	msExchUserCultulre	
Exchange Servers	Allow ACE	All	Read Property	memberOf	
Exchange Servers	Allow ACE	All	Read Property	garbageCollPeriod	
Exchange Servers	Allow ACE	All	Read Property	userAccountControl	
Exchange Servers	Allow ACE	All	Read Property	canonicalName	
Exchange Servers	Allow ACE	All	Replication Synchronization		Extended right
Exchange Servers	Allow ACE	All	Create Child Delete Chile List Children	msExchActiveSyncDevices / User	
Exchange Servers	Allow ACE	All	Create Child Delete Child	msExchActiveSyncDevices / inetOrgPerson	

			List Children		
Exchange Servers	Allow ACE	All	Write Property	msExchSafeSendersHash	
Exchange Servers	Allow ACE	All	Write Property	msExchPublicDelegates	
Exchange Servers	Allow ACE	All	Write Property	msExchMobileMailboxFlags	
Exchange Servers	Allow ACE	All	Write Property	msExchSafeRecipientsHash	
Exchange Servers	Allow ACE	All	Write Property	userCertificate	
Exchange Servers	Allow ACE	All	Write Property	msExchUMDtmfMap	
Exchange Servers	Allow ACE	All	Write Property	msExchBlockedSendersHash	
Exchange Servers	Allow ACE	All	Write Property	msExchUMSpokenName	
Exchange Servers	Allow ACE	All	Write Property	msExchUMPinChecksum	
Exchange Servers	Allow ACE	All	Write Property	thumbnailPhoto	
Organization Management	Allow ACE	All	Read List Object		
Organization Management	Allow ACE	All	Write Property	Exchange Information	
Organization	Allow ACE	All	Write Property	garbageCollPeriod	

Management					
Organization Management	Allow ACE	All	Write Property	legacyExchangeDN	
Organization Management	Allow ACE	All	Write Property	msExchPublicDelegates	
Organization Management	Allow ACE	All	Write Property	textEncodedORAddress	
Organization Management	Allow ACE	All	Write Property	proxyAddresses	
Organization Management	Allow ACE	All	Write Property	mail	
Organization Management	Allow ACE	All	Write Property	displayNamePrintable	
Organization Management	Allow ACE	All	Write Property	showInAddressBook	
Organization Management	Allow ACE	All	Write Property	Exchange Personal Information	
Organization Management	Allow ACE	All	Full Control	/ msExchDynamicDistributionList	
Organization Management	Allow ACE	All	Write Property	adminDisplayName	
Organization Management	Allow ACE	All	Write Property	displayName	
Exchange Trusted Subsystem	Allow ACE	All	Read List Object		

Exchange Trusted Subsystem	Allow ACE	All	Write Property	displayName	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	Public Information	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	msExchPublicDelegates	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	adminDisplayName	
Exchange Trusted Subsystem	Allow ACE	All	Full Control	/ msExchDynamicDistributionList	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	Exchange Information	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	Exchange Personal Information	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	garbageCollPeriod	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	textEncodedORAddress	
Exchange	Allow ACE	All	Write Property	showInAddressBook	

Trusted Subsystem					
Exchange Trusted Subsystem	Allow ACE	All	Write Property	legacyExchangeDN	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	Personal Information	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	proxyAddresses	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	displayNamePrintable	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	mail	
Exchange Windows Permissions	Allow ACE	All	Write Property	pwdLastSet	
Exchange Windows Permissions	Allow ACE	All	WriteDACL	/ user	
Exchange Windows Permissions	Allow ACE	All	WriteDACL	/ inetOrgPerson	
Exchange Windows	Allow ACE	All	Delete Tree	/ user	

Permissions					
Exchange Windows Permissions	Allow ACE	All	Delete Tree	/ inetOrgPerson	
Exchange Windows Permissions	Allow ACE	All	Write Property	sAMAccountName	
Exchange Windows Permissions	Allow ACE	All	Create Child Delete	/ contact	
Exchange Windows Permissions	Allow ACE	All	Create Child Delete	/ inetOrgPerson	
Exchange Windows Permissions	Allow ACE	All	Create Child Delete	/ user	
Exchange Windows Permissions	Allow ACE	All	Create Child Delete	/ organizationUnit	
Exchange Windows Permissions	Allow ACE	All	Create Child Delete	/ group	
Exchange Windows Permissions	Allow ACE	All	Create Child Delete Child	/ computer	
Exchange Windows Permissions	Allow ACE	All	Write Property	Member	

Exchange Windows Permissions	Allow ACE	All	Write Property	wwwHomePage	
Exchange Windows Permissions	Allow ACE	All	Write Property	countryCode	
Exchange Windows Permissions	Allow ACE	All	Write Property	userAccountControl	
Exchange Windows Permissions	Allow ACE	All	Write Property	managedBy	
Exchange Windows Permissions	Allow ACE	All	Reset Password on Next Logon		Extended right
Exchange Windows Permissions	Allow ACE	All	Change Password	/ user	Extended right
Delegated Setup	Allow ACE	All	Read Property	User Account Restrictions	

Distinguished name of the object: CN=AdminSDHolder,CN=System,DC=<domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
Authenticated Users	Allow ACE	All	Read Property	Exchange Information	
NT AUTHORITY \NETWORK	Allow ACE	All	Read Property	Exchange Personal Information	Grants Transport service read permissions.

Exchange Servers	Allow ACE	All	Write Property	groupType	
Exchange Servers	Allow ACE	All	Write Property	msExchMailboxSecurityDescriptor	
Exchange Servers	Allow ACE	All	Write Property	msExchUMServerWritableFlags	
Exchange Servers	Allow ACE	All	Read Property	Exchange Personal Information	
Exchange Servers	Allow ACE	All	Read Property	Exchange Information	
Exchange Servers	Allow ACE	All	Write Property	msExchUserCultulre	
Exchange Servers	Allow ACE	All	Read Property	memberOf	
Exchange Servers	Allow ACE	All	Read Property	garbageCollPeriod	
Exchange Servers	Allow ACE	All	Read Property	userAccountControl	
Exchange Servers	Allow ACE	All	Read Property	canonicalName	
Exchange Servers	Allow ACE	All	Replication Synchronization		Extended right
Exchange Servers	Allow ACE	All	Create Child Delete Chile List Children	msExchActiveSyncDevices / User	

Exchange Servers	Allow ACE	All	Create Child Delete Child List Children	msExchActiveSyncDevices / inetOrgPerson	
Exchange Servers	Allow ACE	All	Write Property	msExchSafeSendersHash	
Exchange Servers	Allow ACE	All	Write Property	msExchPublicDelegates	
Exchange Servers	Allow ACE	All	Write Property	msExchMobileMailboxFlags	
Exchange Servers	Allow ACE	All	Write Property	msExchSafeRecipientsHash	
Exchange Servers	Allow ACE	All	Write Property	userCertificate	
Exchange Servers	Allow ACE	All	Write Property	msExchUMDtmfMap	
Exchange Servers	Allow ACE	All	Write Property	msExchBlockedSendersHash	
Exchange Servers	Allow ACE	All	Write Property	msExchUMSpokenName	
Exchange Servers	Allow ACE	All	Write Property	msExchUMPinChecksum	
Exchange Servers	Allow ACE	All	Write Property	thumbnailPhoto	
Organization Management	Allow ACE	All	Read List Object		
Organization	Allow ACE	All	Write Property	Exchange Information	

Management					
Organization Management	Allow ACE	All	Write Property	garbageCollPeriod	
Organization Management	Allow ACE	All	Write Property	legacyExchangeDN	
Organization Management	Allow ACE	All	Write Property	msExchPublicDelegates	
Organization Management	Allow ACE	All	Write Property	textEncodedORAddress	
Organization Management	Allow ACE	All	Write Property	proxyAddresses	
Organization Management	Allow ACE	All	Write Property	mail	
Organization Management	Allow ACE	All	Write Property	displayNamePrintable	
Organization Management	Allow ACE	All	Write Property	showInAddressBook	
Organization Management	Allow ACE	All	Write Property	Exchange Personal Information	
Organization Management	Allow ACE	All	Full Control	/ msExchDynamicDistributionList	
Organization Management	Allow ACE	All	Write Property	adminDisplayName	
Organization Management	Allow ACE	All	Write Property	displayName	
Exchange	Allow ACE	All	Read		

Trusted Subsystem			List Object		
Exchange Trusted Subsystem	Allow ACE	All	Write Property	displayName	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	Public Information	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	msExchPublicDelegates	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	adminDisplayName	
Exchange Trusted Subsystem	Allow ACE	All	Full Control	/ msExchDynamicDistributionList	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	Exchange Information	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	Exchange Personal Information	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	garbageCollPeriod	
Exchange Trusted	Allow ACE	All	Write Property	textEncodedORAddress	

Subsystem					
Exchange Trusted Subsystem	Allow ACE	All	Write Property	showInAddressBook	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	legacyExchangeDN	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	Personal Information	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	proxyAddresses	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	displayNamePrintable	
Exchange Trusted Subsystem	Allow ACE	All	Write Property	mail	
Exchange Windows Permissions	Allow ACE	All	Write Property	pwdLastSet	
Exchange Windows Permissions	Allow ACE	All	Write Property	sAMAccountName	
Exchange Windows Permissions	Allow ACE	All	Write Property	Member	

Exchange Windows Permissions	Allow ACE	All	Write Property	wwwHomePage	
Exchange Windows Permissions	Allow ACE	All	Write Property	countryCode	
Exchange Windows Permissions	Allow ACE	All	Write Property	userAccountControl	
Exchange Windows Permissions	Allow ACE	All	Write Property	managedBy	
Delegated Setup	Allow ACE	All	Read Property	User Account Restrictions	

Distinguished name of the object: CN=Deleted Objects,DC=<domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Exchange Servers	Allow ACE	All	List Contents	

Distinguished name of the object: CN=Microsoft Exchange System Objects,DC=<domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
NT AUTHORITY \NETWORK	Allow ACE	All	Read Property List Contents Read Permissions	
Authenticated Users	Allow ACE	All	Read Permissions	
Authenticated Users	Allow ACE	All	Read Property	garbageCollPeriod

Authenticated Users	Allow ACE	All	Read Property	adminDisplayName
Authenticated Users	Allow ACE	All	Read Property	modifyTimeStamp
Exchange Servers	Deny ACE	All	Delete Tree	
Exchange Servers	Allow ACE	All	Read Permissions List Contents Read PropertyDelete Tree	
Exchange Servers	Allow ACE	All	Create Child	/ msExchSystemMailbox
Exchange Servers	Allow ACE	All	Create Child Delete Child	/ publicFolder
Exchange Servers	Allow ACE	All	Create Child	/ user
Exchange Servers	Allow ACE	All	Delete Child	/ msExchSystemMailbox
Exchange Servers	Allow ACE	All	Delete Child	/ user
Exchange Servers	Allow ACE	Desc	Write Property	/ publicFolder
Exchange Servers	Allow ACE	Desc	Write Property	/ msExchSystemMailbox
Exchange Servers	Allow ACE	Desc	Write Property	/ user
Exchange Servers	Allow ACE	Desc	Change Password Reset Password on Next Logon	/ user
Organization Management	Allow ACE	All	Read Permissions List Contents	

			Read Property	
Organization Management	Allow ACE	Desc	Write Property	/ msExchSystemMailbox
Organization Management	Allow ACE	All	Create Child Delete Child	/ msExchSystemMailbox
Organization Management	Allow ACE	Desc	Write Property	/ user
Organization Management	Allow ACE	All	Create Child Delete Child	/ user
Organization Management	Allow ACE	Desc	Read Property Write Property	mail / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	displayNamePrintable / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	displayName / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	textEncodedORAddress / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	proxyAddresses / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	cn / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	showInAddressBook / publicFolder
Organization	Allow ACE	Desc	Read Property	Exchange Information / publicFolder

Management			Write Property	
Organization Management	Allow ACE	Desc	Read Property Write Property	legacyExchangeDN / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	Exchange Personal Information / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	msDSPhoneticDisplayName / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	msExchPFContacts / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	garbageCollPeriod / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	name / publicFolder
Organization Management	Allow ACE	Desc	Read Property Write Property	msExchPublicDelegates / publicFolder
Public Folder Management	Allow ACE	All	Read Permissions List Contents Read Property	
Public Folder Management	Allow ACE	Desc	Read Property Write Property	mail / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	displayNamePrintable / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	displayName / publicFolder

Public Folder Management	Allow ACE	Desc	Read Property Write Property	textEncodedORAddress / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	proxyAddresses / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	cn / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	showInAddressBook / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	Exchange Information / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	legacyExchangeDN / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	Exchange Personal Information / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	msDSPhoneticDisplayName / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	msExchPFContacts / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	garbageCollPeriod / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	name / publicFolder
Public Folder Management	Allow ACE	Desc	Read Property Write Property	msExchPublicDelegates / publicFolder

Exchange Trusted Subsystem	Allow ACE	All	Read Permissions List Contents Read Property	
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	mail / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	displayNamePrintable / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	displayName / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	textEncodedORAddress / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	proxyAddresses / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	cn / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	showInAddressBook / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	Exchange Information / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	legacyExchangeDN / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	Exchange Personal Information / publicFolder
Exchange Trusted	Allow ACE	Desc	Read Property	msDSPhoneticDisplayName / publicFolder

Subsystem			Write Property	
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	msExchPFContacts / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	garbageCollPeriod / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	name / publicFolder
Exchange Trusted Subsystem	Allow ACE	Desc	Read Property Write Property	msExchPublicDelegates / publicFolder

Distinguished name of the object: CN=Exchange Install Domain Servers,CN=Microsoft Exchange System Objects,DC=<domain>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Organization Management	Allow ACE	All	Full Control	

Server Role Installation

During installation of the Client Access and Mailbox server roles, Setup adds the Organization Management USG to the administrator security group on the local computer so that members of the management role group named Organization Management can manage the server.

The following permissions table shows the permissions set when you install the Client Access or Mailbox server roles.

Distinguished name of the object: CN=<server>,CN=Servers,CN=<admin group>,CN=Administrative Groups,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
MACHINE$	Allow ACE	All	Read Permissions List Contents		

			Read Property List Object		
MACHINE$	Allow ACE	None	Write Property	msExchServerSite msExchEdgeSyncCredential	
Exchange Servers	Allow ACE	All	Store Transport Access Store Constrained Delegation Store Read Only Access Store Read and Write Access		Extended rights
NT AUTHORITY\NETWORK	Allow ACE	All	Exchange Web Services Token Serialization		Extended right Only granted on Mailbox server role objects.
NT AUTHORITY\NETWORK	Allow ACE	All	Read Permissions List Contents Read Property List Object		
Local System	Allow ACE	All	Read Permissions List Contents Read Property		

			List Object		
Delegated Setup	Allow ACE	All	Full Control		
Delegated Setup	Deny ACE	All	Create Child Delete Child	/msExchPublicMDB	
Authenticated Users	Allow ACE	All	Read Property		
Delegated Setup	Deny ACE	All	Receive As Send As		Extended right

Database Availability Groups

The permissions tables in this section show the permissions set with regards to the database availability groups and its members.

Distinguished name of the object: CN=<DAGName>,CN=Database Availability Groups,CN=<admin group>,CN=Administrative Groups,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to
Authenticated Users	Allow ACE	None	Read Property	

Edge Transport

If you install an Edge Transport server and establish an Edge Subscription with the Exchange organization, the permissions in the following permissions table are set when the Edge Transport server is instantiated into the organization.

Distinguished name of the object: CN=<server>,CN=Servers,CN=<admin group>,CN=Administrative Groups,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
Exchange Servers	Allow ACE	All	Write Property		

Authenticated Users	Allow ACE	None	Read Properties		ACE is defined in schema for `msExchExchangeServer` class objects `defaultSecurityDescriptor`.

Mailbox Server Installation

During installation of the first Mailbox server, the following containers are created, if they do not already exist. The following permissions table shows the permissions that are applied.

Distinguished name of the object: CN=Availability Configuration,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
Exchange Servers	Allow ACE	Desc	Read Property	`msExchAvailabilityUserPassword` / `msExchAvailabilityAddressSpaceObjects`	Extended right

Distinguished name of the object: CN=Default <Server>,CN=SMTP Receive Connectors,CN=Protocols,CN=<Server>,CN=Servers,CN=<admin group>,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
ExchangeLegacyInterop	Deny ACE	All	Accept Forest Headers		
ExchangeLegacyInterop	Deny ACE	All	Accept Organization Headers		
Exchange Servers	Allow ACE	All	Accept Any Sender		
ExchangeLegacyInterop	Allow ACE	All	Accept Any Sender		

S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Any Sender		This is the well-known security identifier (SID) for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Any Sender		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Accept Any Sender		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Accept EXCH50		
ExchangeLegacyInterop	Allow ACE	All	Accept EXCH50		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept EXCH50		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-	Allow ACE	All	Accept EXCH50		This is the well-known SID for

1139599005-3936102811-1022490595-22					Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Accept EXCH50		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Submit Messages to any Recipient		
ExchangeLegacyInterop	Allow ACE	All	Submit Messages to any Recipient		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Submit Messages to any Recipient		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Submit Messages to any Recipient		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-	Allow ACE	All	Submit Messages to		This is the well-known SID for

1139599005-3936102811-1022490595-23			any Recipient		externally secured servers.
Exchange Servers	Allow ACE	All	Accept XShadow		
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept XShadow		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Accept Routing Headers		
ExchangeLegacyInterop	Allow ACE	All	Accept Routing Headers		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Routing Headers		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Routing Headers		This is the well-known SID for Edge Transport servers.
S-1-9-	Allow ACE	All	Accept Routing		This is the well-

1419165041-1139599005-3936102811-1022490595-23			Headers		known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Accept XSessionParams		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept XSessionParams		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept XSessionParams		This is the well-known SID for Mailbox servers.
Exchange Servers	Allow ACE	All	Accept Forest Headers		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Forest Headers		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-	Allow ACE	All	Accept Forest Headers		This is the well-known SID for

1139599005-3936102811-1022490595-22					Edge Transport servers.
Exchange Servers	Allow ACE	All	Accept xAttr		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept xAttr		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept xAttr		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Accept XProxyFrom		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Forest XProxyFrom		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-	Allow ACE	All	Accept Forest XProxyFrom		This is the well-known SID for Edge Transport servers.

1022490595-22					
Exchange Servers	Allow ACE	All	Accept XSysProbe		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Forest XSysProbe		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Forest XSysProbe		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Send XMessageContext Extended Properties		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Send XMessageContext Extended Properties		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-	Allow ACE	All	Send XMessageContext Extended Properties		This is the well-known SID for Edge Transport servers.

1022490595-22					
Exchange Servers	Allow ACE	All	Send XMessageContext Fast Index		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Send XMessageContext Fast Index		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Send XMessageContext Fast Index		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Send XMessageContext AD Recipient Cache		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Send XMessageContext AD Recipient Cache		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-	Allow ACE	All	Send XMessageCont		This is the well-known SID for

1139599005-3936102811-1022490595-22			ext AD Recipient Cache		Edge Transport servers.
Exchange Servers	Allow ACE	All	Accept Authentication Flag		
ExchangeLegacyInterop	Allow ACE	All	Accept Authentication Flag		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Authentication Flag		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Authentication Flag		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Accept Authentication Flag		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Bypass Anti-Spam		

ExchangeLegacyInterop	Allow ACE	All	Bypass Anti-Spam		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Bypass Anti-Spam		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Bypass Anti-Spam		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Bypass Anti-Spam		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Bypass Message Size Limit		
ExchangeLegacyInterop	Allow ACE	All	Bypass Message Size Limit		
S-1-9-1419165041-1139599005-3936102811-	Allow ACE	All	Bypass Message Size Limit		This is the well-known SID for Mailbox servers.

1022490595-21					
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Bypass Message Size Limit		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Bypass Message Size Limit		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Accept Organization Headers		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Organization Headers		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Organization Headers		This is the well-known SID for Edge Transport servers.
Exchange	Allow ACE	All	Submit		

Servers			Messages to Server		
ExchangeLegacyInterop	Allow ACE	All	Submit Messages to Server		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Submit Messages to Server		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Submit Messages to Server		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Submit Messages to Server		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Accept Authoritative Domain Sender		
ExchangeLegacyInterop	Allow ACE	All	Accept Authoritative Domain Sender		

S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Authoritative Domain Sender		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Authoritative Domain Sender		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Accept Authoritative Domain Sender		This is the well-known SID for externally secured servers.
Authenticated Users	Allow ACE	All	Submit Messages to any Recipient		
Authenticated Users	Allow ACE	All	Accept Routing Headers		
Authenticated Users	Allow ACE	All	Bypass Anti-Spam		
Authenticated Users	Allow ACE	All	Submit Messages to Server		

Distinguished name of the object: CN=Client <Server>,CN=SMTP Receive

Connectors,CN=Protocols,CN=<Server>,CN=Servers,CN=<admin group>,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
Authenticated Users	Allow ACE	All	Submit Messages to any Recipient		
Authenticated Users	Allow ACE	All	Accept Routing Headers		
Authenticated Users	Allow ACE	All	Bypass Anti-Spam		
Authenticated Users	Allow ACE	All	Submit Messages to Server		
Exchange Servers	Allow ACE	All	Accept XSessionParams		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept XSessionParams		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept XSessionParams		This is the well-known SID for Mailbox servers.

Exchange Servers	Allow ACE	All	Accept Any Sender		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Any Sender		This is the well-known security identifier (SID) for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Any Sender		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Accept Any Sender		This is the well-known SID for externally secured servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Exch50		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-	Allow ACE	All	Accept Exch50		This is the well-known SID for Edge Transport servers.

1022490595-22					
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Accept Exch50		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Accept Exch50		
Exchange Servers	Allow ACE	All	Submit Messages to any Recipient		
ExchangeLegacyInterop	Allow ACE	All	Submit Messages to any Recipient		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Submit Messages to any Recipient		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Submit Messages to any Recipient		This is the well-known SID for Edge Transport servers.
S-1-9-	Allow ACE	All	Submit		This is the well-

1419165041-1139599005-3936102811-1022490595-23			Messages to any Recipient		known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Accept XShadow		
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept XShadow		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Accept Routing Headers		
ExchangeLegacyInterop	Allow ACE	All	Accept Routing Headers		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Routing Headers		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Routing Headers		This is the well-known SID for Edge Transport servers.

S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Accept Routing Headers		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Accept Forest Headers		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Forest Headers		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Forest Headers		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Accept xAttr		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept xAttr		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-	Allow ACE	All	Accept xAttr		This is the well-known SID for

1139599005-3936102811-1022490595-22					Edge Transport servers.
Exchange Servers	Allow ACE	All	Accept XProxyFrom		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Forest XProxyFrom		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Forest XProxyFrom		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Accept Authentication Flag		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Authentication Flag		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-	Allow ACE	All	Accept Authentication Flag		This is the well-known SID for Edge Transport

3936102811-1022490595-22					servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Accept Authentication Flag		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Accept XSysProbe		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Forest XSysProbe		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Forest XSysProbe		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Accept Authentication Flag		This is the well-known SID for externally secured servers.
Exchange	Allow ACE	All	Bypass Anti-		

Servers			Spam		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Bypass Anti-Spam		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Bypass Anti-Spam		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Bypass Anti-Spam		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Send XMessageContext Extended Properties		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Send XMessageContext Extended Properties		This is the well-known SID for Mailbox servers.
S-1-9-	Allow ACE	All	Send		This is the well-

1419165041-1139599005-3936102811-1022490595-22			XMessageContext Extended Properties		known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Send XMessageContext Fast Index		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Send XMessageContext Fast Index		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Send XMessageContext Fast Index		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Bypass Message Size Limit		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Bypass Message Size Limit		This is the well-known SID for Mailbox servers.
S-1-9-	Allow ACE	All	Bypass		This is the well-

1419165041-1139599005-3936102811-1022490595-22			Message Size Limit		known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Bypass Message Size Limit		This is the well-known SID for externally secured servers.
Exchange Servers	Allow ACE	All	Accept Organization Headers		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Organization Headers		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Organization Headers		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Send XMessageContext AD Recipient		

			Cache		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Send XMessageContext AD Recipient Cache		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Send XMessageContext AD Recipient Cache		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Submit Messages to Server		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Submit Messages to Server		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Submit Messages to Server		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-	Allow ACE	All	Submit Messages to		This is the well-known SID for

1139599005-3936102811-1022490595-23			Server		externally secured servers.
Exchange Servers	Allow ACE	All	Accept Authoritative Domain Sender		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Accept Authoritative Domain Sender		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Accept Authoritative Domain Sender		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Accept Authoritative Domain Sender		This is the well-known SID for externally secured servers.

SMTP Send Connector Creation

The following table shows the permissions set when you create Send connectors.

Distinguished name of the object: CN=<Connector

Name>,CN=Connections,CN=<routing group>,CN=Routing Groups, CN=<admin group>,CN=<organization>

Account	ACE type	Inheritance	Permissions	On property/ Applies to	Comments
NT AUTHORITY \ANONYMOUS LOGON	Allow ACE	All	Send Routing Headers		
Exchange Servers	Allow ACE	All	Send Organization Headers		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Send Organization Headers		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Send Organization Headers		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Send Forest Headers		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-	Allow ACE	All	Send Forest Headers		This is the well-known SID for Mailbox servers.

21					
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Send Forest Headers		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Send XShadow		
S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Send XShadow		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Send XShadow		This is the well-known SID for Edge Transport servers.
Exchange Servers	Allow ACE	All	Send Routing Headers		
S-1-9-1419165041-1139599005-3936102811-1022490595-10	Allow ACE	All	Send Routing Headers		This is the well-known SID for partner servers.

S-1-9-1419165041-1139599005-3936102811-1022490595-21	Allow ACE	All	Send Routing Headers		This is the well-known SID for Mailbox servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Send Routing Headers		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Send Routing Headers		This is the well-known SID for externally secured servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-24	Allow ACE	All	Send Routing Headers		This is the well-known SID for Legacy Exchange Servers.
Exchange Servers	Allow ACE	All	Send Exch50		
S-1-9-1419165041-1139599005-3936102811-	Allow ACE	All	Send Exch50		This is the well-known SID for Mailbox servers.

1022490595-21					
S-1-9-1419165041-1139599005-3936102811-1022490595-22	Allow ACE	All	Send Exch50		This is the well-known SID for Edge Transport servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-23	Allow ACE	All	Send Exch50		This is the well-known SID for externally secured servers.
S-1-9-1419165041-1139599005-3936102811-1022490595-24	Allow ACE	All	Send Exch50		This is the well-known SID for Legacy Exchange Servers.

Deployment security checklist

Exchange Server 2013 > Planning and deployment > Deployment reference >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-01*

Microsoft Exchange Server 2013 features are designed to help improve the security of your messaging environment. Generally, for Exchange 2013, the following conditions are true:

- Accounts that are used by Exchange 2013 have the minimum rights that are required to perform the given task sets.
- By default, services are started only when they are required.

- Access control list (ACL) rights for Exchange objects are minimized.
- Administrative permissions are set according to the scope of change on the object that a given modification requires.
- By default, all internal default message paths are encrypted.

This topic lists steps that we recommend you take to harden the messaging environment before you install Microsoft Exchange. We recommend that you refer to this checklist every time that you install a new Exchange server role.

Before installing Exchange 2013, perform the following procedures.

Procedure	Done?
Run Microsoft Update.	
Run the Microsoft Windows Malicious Software Removal Tool. The Malicious Software Removal Tool is included with Microsoft Update. More information about the tool can be found at Malicious Software Removal Tool.	
Run the Microsoft Baseline Security Analyzer.	

Exchange 2013 sizing and capacity planning

Exchange Server 2013 > Planning and deployment > Deployment reference >

Topic Last Modified: *2013-05-16*

Sizing and capacity planning for Exchange Server 2013 is an important part of your Exchange 2013 deployment. Configuring a system for optimum performance is an iterative process. Take the time to understand all the variables that affect your system, including user profile, architecture, and hardware. With this knowledge, you can establish baseline metrics for your systems and make adjustments to improve system performance. For more information and guidance about sizing and capacity planning for your Exchange organization, see the Exchange Team blog article Ask the Perf Guy: Sizing Exchange 2013 Deployments.

Exchange 2013 storage configuration options

Applies to: *Exchange Server 2013, Exchange Server, Exchange Online*

Topic Last Modified: *2014-07-10*

Understanding storage options and requirements for the Mailbox server role in Microsoft Exchange Server 2013 is an important part of your Mailbox server storage design solution.

Contents

Storage architectures

The following table describes supported storage architectures and provides best practice guidance for each type of storage architecture where appropriate.

Supported storage architectures

Storage architecture	Description	Best practice
Direct-attached storage (DAS)	DAS is a digital storage system directly attached to a server or workstation, without a storage network in between. For example, DAS transports include Serial Attached Small Computer System Interface (SCSI) and Serial Attached Advanced Technology Attachment (ATA).	Not available.
Storage area network (SAN): Internet Small Computer	SAN is an architecture to attach remote computer storage	Don't share physical disks backing up Exchange data with

System Interface (iSCSI)	devices (such as disk arrays and tape libraries) to servers in such a way that the devices appear as locally attached to the operating system (for example, block storage). iSCSI SANs encapsulate SCSI commands within IP packets and use standard networking infrastructure as the storage transport (for example, Ethernet).	other applications. Use dedicated storage networks. Use multiple network paths for stand-alone configurations.
SAN: Fibre Channel	Fibre Channel SANs encapsulate SCSI commands within Fibre Channel packets and generally utilize specialized Fibre Channel networks as the storage transport.	Don't share physical disks backing up Exchange data with other applications. Use multiple Fibre Channel network paths for stand-alone configurations. Follow storage vendor's best practices for tuning Fibre Channel host bus adapters (HBAs), for example, Queue Depth and Queue Target.

A network-attached storage (NAS) unit is a self-contained computer connected to a network, with the sole purpose of supplying file-based data storage services to other devices on the network. The operating system and other software on the NAS unit provide the functionality of data storage, file systems, and access to files, and the management of these functionalities (for example, file storage).

All storage used by Exchange for storage of Exchange data must be block-level storage because Exchange 2013 doesn't support the use of NAS volumes, other than in the SMB 3.0 scenario outlined in the topic Exchange 2013 virtualization. Also, in a virtualized environment, NAS storage that's presented to the guest as block-level storage via the hypervisor isn't supported.

Return to top

Physical disk types

The following table provides a list of supported physical disk types and provides best practice guidance for each physical disk type where appropriate.

Supported physical disk types

Physical disk type	Description	Supported or best practice
Serial ATA (SATA)	SATA is a serial interface for ATA and integrated device electronics (IDE) disks. SATA disks are available in a variety of form factors, speeds, and capacities. In general, choose SATA disks for Exchange 2013 mailbox storage when you have the following design requirements: • High capacity • Moderate performance • Moderate power utilization	Supported: 512-byte sector disks for Windows Server 2008 and Windows Server 2008 R2. In addition, 512e disks are supported for Windows Server 2008 R2 with the following: • The hotfix described in Microsoft Knowledge Base article 982018, An update that improves the compatibility of Windows 7 and Windows Server 2008 R2 with Advanced Format Disks is available. • Windows Server 2008 R2 with Service Pack 1 (SP1) and Exchange Server 2010 SP1. Support requires that all copies of a database reside on the same physical disk type. For example, it is not a supported configuration to host one copy of a given database on a 512-byte sector disk and another copy of that same database on a 512e disk. Also be aware that 4-kilobyte (KB) sector disks are not supported for any version of

		Microsoft Exchange and 512e disks are not supported for any version of Exchange prior to Exchange Server 2010 SP1. Best practice: Consider enterprise class SATA disks, which generally have better heat, vibration, and reliability characteristics.
Serial Attached SCSI	Serial Attached SCSI is a serial interface for SCSI disks. Serial Attached SCSI disks are available in a variety of form factors, speeds, and capacities. In general, choose Serial Attached SCSI disks for Exchange 2013 mailbox storage when you have the following design requirements: • Moderate capacity • High performance • Moderate power utilization	Supported: 512-byte sector disks for Windows Server 2008 and Windows Server 2008 R2. In addition, 512e disks are supported for Windows Server 2008 R2 with the following: • The hotfix described in Microsoft Knowledge Base article 982018, An update that improves the compatibility of Windows 7 and Windows Server 2008 R2 with Advanced Format Disks is available. • Windows Server 2008 R2 with Service Pack 1 (SP1) and Exchange Server 2010 SP1. Support requires that all copies of a database reside on the same physical disk type. For example, it is not a supported configuration to host one copy of a given database on a 512-byte sector disk and another

		copy of that same database on a 512e disk. Also be aware that 4-kilobyte (KB) sector disks are not supported for any version of Microsoft Exchange and 512e disks are not supported for any version of Exchange prior to Exchange Server 2010 SP1. Best practice: Physical disk-write caching must be disabled when used without a UPS.
Fibre Channel	Fibre Channel is an electrical interface used to connect disks to Fibre Channel-based SANs. Fibre Channel disks are available in a variety of speeds and capacities. In general, choose Fibre Channel disks for Exchange 2013 mailbox storage when you have the following design requirements: • Moderate capacity • High performance • SAN connectivity	Supported: 512-byte sector disks for Windows Server 2008 and Windows Server 2008 R2. In addition, 512e disks are supported for Windows Server 2008 R2 with the following: • The hotfix described in Microsoft Knowledge Base article 982018, An update that improves the compatibility of Windows 7 and Windows Server 2008 R2 with Advanced Format Disks is available. • Windows Server 2008 R2 with Service Pack 1 (SP1) and Exchange Server 2010 SP1. Support requires that all copies of a database reside on the same physical disk type. For example, it is not a supported configuration to host one copy

		of a given database on a 512-byte sector disk and another copy of that same database on a 512e disk. Also be aware that 4-kilobyte (KB) sector disks are not supported for any version of Microsoft Exchange and 512e disks are not supported for any version of Exchange prior to Exchange Server 2010 SP1. Best practice: Physical disk-write caching must be disabled when used without a UPS.
Solid-state drive (SSD) (flash disk)	An SSD is a data storage device that uses solid-state memory to store persistent data. An SSD emulates a hard disk drive interface. SSD disks are available in a variety of speeds (different I/O performance capabilities) and capacities. In general, choose SSD disks for Exchange 2013 mailbox storage when you have the following design requirements: • Low capacity • Extremely high performance	Supported: 512-byte sector disks for Windows Server 2008 and Windows Server 2008 R2. In addition, 512e disks are supported for Windows Server 2008 R2 with the following: • The hotfix described in Microsoft Knowledge Base article 982018, An update that improves the compatibility of Windows 7 and Windows Server 2008 R2 with Advanced Format Disks is available. • Windows Server 2008 R2 with Service Pack 1 (SP1) and Exchange Server 2010 SP1. Support requires that all copies of a database reside on the same physical disk type. For example,

		it is not a supported configuration to host one copy of a given database on a 512-byte sector disk and another copy of that same database on a 512e disk. Also be aware that 4-kilobyte (KB) sector disks are not supported for any version of Microsoft Exchange and 512e disks are not supported for any version of Exchange prior to Exchange Server 2010 SP1. Best practice: Physical disk-write caching must be disabled when used without a UPS. In general, Exchange 2013 Mailbox servers don't require the performance characteristics of SSD storage.

Factors to consider when choosing disk types

There are several trade-offs when choosing disk types for Exchange 2013 storage. The correct disk is one that balances performance (both sequential and random) with capacity, reliability, power utilization, and capital cost. The following table of supported physical disk types provides information to help you when considering these factors.

Factors in disk type choice

Disk speed (RPM)	Disk form factor	Interface or transport	Capacity	Random I/O performance	Sequential I/O performance	Power utilization
5,400	2.5-inch	SATA	Average	Poor	Poor	Excellent
5,400	3.5-inch	SATA	Excellent	Poor	Poor	Above average

7,200	2.5-inch	SATA	Average	Average	Average	Excellent
7,200	2.5-inch	Serial Attached SCSI	Average	Average	Above average	Excellent
7,200	3.5-inch	SATA	Excellent	Average	Above average	Above average
7,200	3.5-inch	Serial Attached SCSI	Excellent	Average	Above average	Above average
7,200	3.5-inch	Fibre Channel	Excellent	Average	Above average	Average
10,000	2.5-inch	Serial Attached SCSI	Below average	Excellent	Above average	Above average
10,000	3.5-inch	SATA	Average	Average	Above average	Above average
10,000	3.5-inch	Serial Attached SCSI	Average	Above average	Above average	Below average
10,000	3.5-inch	Fibre Channel	Average	Above average	Above average	Below average
15,000	2.5-inch	Serial Attached SCSI	Poor	Excellent	Excellent	Average
15,000	3.5-inch	Serial Attached SCSI	Average	Excellent	Excellent	Below average

15,000	3.5-inch	Fibre Channel	Average	Excellent	Excellent	Poor
SSD: enterprise class	Not applicable	SATA, Serial Attached SCSI, Fibre Channel	Poor	Excellent	Excellent	Excellent

Return to top

Best practices for supported storage configurations

This section provides best practice information about supported disk and array controller configurations.

Redundant Array of Independent Disks (RAID) is often used to both improve the performance characteristics of individual disks (by striping data across several disks) as well as to provide protection from individual disk failures. With the advancements in Exchange 2013 high availability, RAID is not a required component for Exchange 2013 storage design. However, RAID is still an essential component of Exchange 2013 storage design for standalone servers as well as solutions that require storage fault tolerance.

Operating System, System, or Pagefile Volume

The recommended configuration for an operating system, system or pagefile volume is to utilize RAID technology to protect this data type. The recommended RAID configuration is either RAID-1 or RAID-1/0, however all RAID types are supported.

Separated Mailbox Database and Log Volumes

If you are deploying a standalone Mailbox server role architecture, RAID technology is required for the mailbox database and log volumes. The recommended RAID configuration for mailbox volumes is RAID-1/0 (especially if you are using 5.4K or 7.2K disks); however all RAID types are supported. For log volumes, RAID-1 or RAID-1/0 is the recommended RAID configuration.

When using RAID-5 or RAID-6 configurations for the operating system, pagefile, or Exchange data volumes, note the following:

- RAID-5 configurations, including variations such as RAID-50 and RAID-51, should have no more than 7 disks per array group and array controller high-priority scrubbing and surface scanning enabled.
- RAID-6 configurations should have array controller high-priority scrubbing and surface scanning enabled.

While JBOD is supported in high availability architectures that have 3 or more highly available database copies, because the log and mailbox database volumes are separated, JBOD is not

recommended.

Mailbox Database and Log Volume Co-Location

Mailbox database and log volume co-location is not recommended in standalone architectures. In high availability architectures, there are two possibilities for this scenario:
1. Single database per volume
2. Multiple databases per volume

Single Database Per Volume

From an Exchange perspective, JBOD means having both the database and its associated logs stored on a single disk. To deploy on JBOD, you must deploy a minimum of three highly available database copies. Utilizing a single disk is a single point of failure, because when the disk fails, the database copy residing on that disk is lost. Having a minimum of three database copies ensures fault tolerance by having two additional copies in the event that one copy (or one disk) fails. However, placement of three highly available database copies, as well as the use of lagged database copies, can affect storage design. The following table shows guidelines for RAID or JBOD considerations.

RAID or JBOD Considerations

Datacenter servers	Two highly available copies (total)	Three highly available copies (total)	Two or more highly available copies per datacenter	One lagged copy	Two or more lagged copies per datacenter
Primary datacenter servers	RAID	RAID or JBOD (2 copies)	RAID or JBOD	RAID	RAID or JBOD
Secondary datacenter servers	RAID	RAID (1 copy)	RAID or JBOD	RAID	RAID or JBOD

To deploy on JBOD with the primary datacenter servers, you need three or more highly available database copies within the DAG. If mixing lagged copies on the same server hosting highly available database copies (for example, not using dedicated lagged database copy servers), you need at least two lagged database copies.

For the secondary datacenter servers to use JBOD, you should have at least two highly available database copies in the secondary datacenter. The loss of a copy in the secondary datacenter won't result in requiring a reseed across the WAN or having a single point of failure in the event the secondary datacenter is activated. If mixing lagged database copies on the same server hosting highly available database copies (for example, not using dedicated lagged database copy servers), you need at least two lagged database copies.

For dedicated lagged database copy servers, you should have at least two lagged database copies within a datacenter to use JBOD. Otherwise, the loss of disk results in the loss of the lagged database copy, as well as the loss of the protection mechanism.

Multiple Databases Per Volume

Multiple databases per volume is a new JBOD scenario available in Exchange 2013 that allows for active and passive copies (including lagged copies) to be mixed on a single disk, enabling better disk utilization. However, to deploy lagged copies in this manner, automatic lagged copy log file play down must be enabled. The following table shows guidelines for JBOD considerations for multiple databases per volume.

JBOD Considerations

Datacenter Servers	3 or more copies (total)	Two or more copies per datacenter
Primary datacenter servers	JBOD	JBOD
Secondary datacenter servers	N/A	JBOD

The following table provides guidance about storage array configurations for Exchange 2013.

Supported RAID types for the Exchange 2013 Mailbox server role

RAID type	Description	Supported or best practice
Disk array RAID stripe size (KB)	The stripe size is the per disk unit of data distribution within a RAID set. Stripe size is also referred to as *block size*.	Best practice: 256 KB or greater. Follow storage vendor best practices.
Storage array cache settings	The cache settings are provided by a battery-backed caching array controller.	Best practice: 75 percent write cache and 25 percent read cache (battery-backed cache). Follow storage vendor best practices. This guidance also applies to JBOD configurations.
Physical disk write caching	The settings for the cache are on each individual disk.	Supported: Physical disk write caching must be disabled when used without a UPS.

The following table provides guidance about database and log file choices.

Database and log file choices for the Exchange 2013 Mailbox server role

Database and log file options	Description	Stand-alone: supported or best practice	High availability: supported or best practice
File placement: database per log isolation	Database per log isolation refers to placing the database file and logs from the same mailbox database onto different volumes backed by different physical disks.	Best practice: For recoverability, move database (.edb) file and logs from the same database to different volumes backed by different physical disks.	Supported: Isolation of logs and databases isn't required.
File placement: database files per volume	Database files per volume refers to how you distribute database files within or across disk volumes.	Best practice: Based on your backup methodology.	Supported: When using JBOD, create a single volume with separate directories for database(s) and for log files.
File placement: log streams per volume	Log streams per volume refers to how you distribute database log files within or across disk volumes.	Best practice: Based on your backup methodology.	Supported: When using JBOD, create a single volume with separate directories for database(s) and for log files. Best practice: When using JBOD, leverage multiple databases per volume.
Database size	Database size refers to the disk database (.edb) file size.	Supported: Approximately 16 terabytes. Best practice:	Supported: Approximately 16 terabytes. Best practice:

		• 200 gigabytes (GB) or less. • Provision for 120 percent of calculated maximum database size.	• 2 terabytes or less. • Provision for 120 percent of calculated maximum database size.
Log truncation method	Log truncation method is the process for truncating and deleting old database log files. There are two mechanisms: • Circular logging, in which Exchange deletes the logs. • Log truncation, which occurs after a successful full or incremental Volume Shadow Copy Service (VSS) backup.	Best practice: • Use backups for log truncation (for example, circular logging disabled). • Provision for three days of log generation capacity.	Best practice: • Enable circular logging for deployments that use Exchange native data protection features. • Provision for three days beyond replay lag setting of log generation capacity.

The following table provides guidance about Windows disk types.

Windows disk types for the Exchange 2013 Mailbox server role

Windows disk type	Description	Stand-alone: supported or best practice	High availability: supported or best practice
Basic disk	A disk initialized for basic storage is called a basic disk. A basic disk contains basic volumes, such as primary partitions, extended partitions, and logical	Supported. Best practice: Use basic disks.	Supported. Best practice: Use basic disks.

	drives.		
Dynamic disk	A disk initialized for dynamic storage is called a dynamic disk. A dynamic disk contains dynamic volumes, such as simple volumes, spanned volumes, striped volumes, mirrored volumes, and RAID-5 volumes.	Supported.	Supported.

The following table provides guidance on volume configurations.

Volume configurations for the Exchange 2013 Mailbox server role

Volume configuration	Description	Stand-alone: supported or best practice	High availability: supported or best practice
GUID partition table (GPT)	GPT is a disk architecture that expands on the older master boot record (MBR) partitioning scheme. The maximum NTFS formatted partition size is 256 terabytes.	Supported. Best practice: Use GPT partitions.	Supported. Best practice: Use GPT partitions.
MBR	An MBR, or partition sector, is the 512-byte boot sector that is the first sector (LBA Sector 0) of a partitioned data storage device such as a hard disk. The maximum NTFS formatted partition size is	Supported.	Supported.

	2 terabytes.		
Partition alignment	Partition alignment refers to aligning partitions on sector boundaries for optimal performance.	Supported: The Windows Server 2008 R2 and Windows Server 2012 default is 1 megabyte (MB).	Supported: The Windows Server 2008 R2 and Windows Server 2012 default is 1 MB.
Volume path	Volume path refers to how a volume is accessed.	Supported: Drive letter or mount point. Best practice: Mount point host volume must be RAID enabled.	Supported: Drive letter or mount point. Best practice: Mount point host volume must be RAID-enabled.
File system	File system is a method for storing and organizing computer files and the data they contain to make it easy to find and access the files.	Supported: NTFS and ReFS.	Supported: NTFS and ReFS.
NTFS defragmentation	NTFS defragmentation is a process that reduces the amount of fragmentation in Windows file systems. It does this by physically organizing the contents of the disk to store the pieces of each file close together and contiguously.	Supported. Best practice: Not required and not recommended. On Windows Server 2012, we also recommend disabling the automatic disk optimization and defragmentation feature.	Supported. Best practice: Not required and not recommended. On Windows Server 2012, we also recommend disabling the automatic disk optimization and defragmentation feature.
NTFS allocation unit	NTFS allocation unit size	Supported: All	Supported: All

size	represents the smallest amount of disk space that can be allocated to hold a file.	allocation unit sizes. Best practice: 64 KB for both .edb and log file volumes.	allocation unit sizes. Best practice: 64 KB for both .edb and log file volumes.
NTFS compression	NTFS compression is the process of reducing the actual size of a file stored on the hard disk.	Supported: Not supported for Exchange database or log files.	Supported: Not supported for Exchange database or log files.
NTFS Encrypting File System (EFS)	EFS enables users to encrypt individual files, folders, or entire data drives. Because EFS provides strong encryption through industry-standard algorithms and public key cryptography, encrypted files are confidential even if an attacker bypasses system security.	Supported: Not supported for Exchange database or log files.	Not supported for Exchange database or log files.
Windows BitLocker (volume encryption)	Windows BitLocker is a data protection feature in Windows Server 2008. BitLocker protects against data theft or exposure on computers that are lost or stolen, and it offers more secure data deletion when computers are decommissioned.	Supported: All Exchange database and log files.	Supported: All Exchange database and log files. Windows failover clusters require Windows Server 2008 R2 or Windows Server 2008 R2 SP1 and the following hotfix: You cannot enable BitLocker on a disk

			volume in Windows Server 2008 R2 if the computer is a failover cluster node. Exchange volumes with Bitlocker enabled are not supported on Windows failover clusters running earlier versions of Windows. For more information about Windows 7 BitLocker encryption, see BitLocker Drive Encryption in Windows 7: Frequently Asked Questions.
Server Message Block (SMB) 3.0	The Server Message Block (SMB) protocol is a network file sharing protocol (on top of TCP/IP or other network protocols) that allows applications on a computer to access files and resources on a remote server. It also allows applications to communicate with any server program that is set	Limited Support. Supported scenario is a hardware virtualized deployment where the disks are hosted on VHDs on an SMB 3.0 share. These VHDs are presented to the host via a hypervisor. For more information, see Exchange 2013 virtualization.	Limited Support. Supported scenario is a hardware virtualized deployment where the disks are hosted on VHDs on an SMB 3.0 share. These VHDs are presented to the host via a hypervisor. For more information, see Exchange 2013 virtualization.

	up to receive an SMB client request. Windows Server 2012 introduces the new 3.0 version of the SMB protocol with the following features: • SMB Transparent failover • SMB Scaleout • SMB Multichannel • SMB Direct • SMB Encryption • VSS for SMB file shares • SMB Directory Leasing • SMB PowerShell		
Storage Spaces	Storage Spaces is a new storage solution that delivers virtualization capabilities for Windows Server 2012. Storage Spaces allow you to organize physical disks into storage pools, which can be easily expanded by simply adding disks. These disks can be connected either through USB, SATA or SAS. It also utilizes virtual disks (spaces), which behave just like physical disks, with associated powerful capabilities such as thin	Supported. Same restrictions as for physical disk types outlined in this topic.	Supported. Same restrictions as for physical disk types outlined in this topic.

	provisioning, as well as resiliency to failures of underlying physical media. For more information on Storage Spaces, see Storage Spaces Overview.		
Resilient File System (ReFS)	ReFS is a newly engineered file system for Windows Server 2012 that is built on the foundations of NTFS. ReFS maintains high degree of compatibility with NTFS while providing enhanced data verification and auto-correction techniques as well as an integrated end-to-end resiliency to corruptions especially when used in conjunction with the storage spaces feature. For more information on ReFS, see Resilient File System Overview.	Supported for volumes containing Exchange database files, log files and content indexing files, provided that the following hotfix is installed: Exchange Server 2013 databases become fragmented in Windows Server 2012. Not supported for volumes containing Exchange binaries. Best practice: Data integrity features must be disabled for the Exchange database (.edb) files or the volume that hosts these files. Integrity features can be enabled for volumes containing the content	Supported for volumes containing Exchange database files, log files and content indexing files, provided that the following hotfix is installed: Exchange Server 2013 databases become fragmented in Windows Server 2012. Not supported for volumes containing Exchange binaries. Best practice: Data integrity features must be disabled for the Exchange database (.edb) files or the volume that hosts these files. Integrity features can be enabled for volumes containing the content

		index catalog, provided that the volume does not contain any databases or log files.	index catalog, provided that the volume does not contain any databases or log files.
Data De-Duplication	Data deduplication is a new technique to optimize storage utilization for Windows Server 2012. It is a method of finding and removing duplication within data without compromising its fidelity or integrity. The goal is to store more data in less space by segmenting files into small variable-sized chunks, identifying duplicate chunks, and maintaining a single copy of each chunk. Redundant copies of the chunk are replaced by a reference to the single copy, the chunks are organized into container files, and the containers are compressed for further space optimization.	Not Supported for Exchange database files. Note: Can be used for Exchange database files that are completely offline (used as backups or archives).	Not Supported for Exchange database files. Note: Can be used for Exchange database files that are completely offline (used as backups or archives).

Return to top

IPv6 support in Exchange 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-11*

Internet Protocol version 6 (IPv6) is the most recent version of the Internet Protocol (IP). IPv6 is intended to correct many of the shortcomings of IPv4, which was the previous version of the IP.

In Microsoft Exchange Server 2013, IPv6 is supported only when IPv4 is also installed and enabled. If Exchange 2013 is deployed in this configuration, and the network supports IPv4 and IPv6, all Exchange servers can send data to and receive data from devices, servers, and clients that use IPv6 addresses.

This topic discusses IPv6 addressing in Exchange 2013. For additional background information about IPv6, see IPv6.

Contents

IPv6 support in Exchange 2013 components

The following table describes the components in Exchange 2013 that are affected by IPv6.

Exchange 2013 features and IPv6

Feature	IPv6 supported	Comments
IP Allow list and IP Block list in the Connection Filtering agent	Yes	
IP Allow List providers and IP Block List providers in the Connection Filtering agent.	No	Currently, there is no widely accepted industry standard protocol for looking up IPv6 addresses. Most IP Block List providers don't support IPv6 addresses. If you allow anonymous connections from

		unknown IPv6 addresses on a Receive connector, you increase the risk that spammers will bypass IP Block List providers and successfully deliver spam into your organization.
Sender reputation in the Protocol Analysis agent	No	The Protocol Analysis agent doesn't compute the sender reputation level (SRL) for messages that originate from IPv6 senders. For more information about sender reputation, see Sender reputation and the Protocol Analysis agent.
Sender ID	Yes	For more information, see Sender ID.
Receive connectors	Yes	IPv6 addresses are accepted for the following components: • Local IP address bindings • Remote IP addresses • IP address ranges We strongly recommend against configuring Receive connectors to accept anonymous connections from unknown IPv6 addresses. If your organization must receive mail from senders who use IPv6 addresses, create a dedicated Receive connector that restricts the remote IP addresses to the specific IPv6 addresses that those

		senders use. For more information, see Receive connectors.
Send connectors	Yes	IPv6 addresses are accepted for the following components: • Smart host IP addresses • The *SourceIPAddress* parameter for Send connectors configured on Edge Transport servers **Note:** If you want to specify an IPv6 address for the *SourceIPAddress* parameter, make sure that the appropriate DNS AAAA and mail exchange (MX) records are configured correctly. This helps ensure message delivery if a remote messaging server tries any kind of reverse lookup test on the specified IPv6 address. For more information, see Send connectors.
Incoming message rate limits	Partial	Incoming message rate limits that you can set on a Receive connector, such as the *MaxInboundConnectionPercentagePerSource* parameter, the *MaxInboundConnectionPerSource* parameter, and the *TarpitInterval* parameter, only apply to a global IPv6 address. Link local IPv6 addresses and site local IPv6 addresses aren't affected by any specified incoming message rate

		limits.
Unified Messaging	Yes	For more information, see IPv6 support in Unified Messaging.
Database availability group (DAG) member	Yes	Static IPv6 addresses are supported by Windows Server and the Cluster service. However, using static IPv6 addresses goes against best practices. Exchange 2013 doesn't support the configuration of static IPv6 addresses during setup. Failover clusters support Intra-site Automatic Tunnel Addressing Protocol (ISATAP). They support only IPv6 addresses that allow for dynamic registration in DNS. Link local addresses can't be used in a cluster. For more information about DAG network requirements, see the "Network requirements" section in Planning for high availability and site resilience.

Return to top

Enable or disable protocols in the operating system

Exchange 2013 servers fully support IPv6 networks. Therefore, even if you aren't using IPv6, you don't need to disable IPv6 on your Exchange servers.

IPv6 support in Exchange 2013 requires IPv4 to be installed and enabled on all Exchange 2013 servers. Uninstalling IPv4 from your Exchange 2013 servers isn't supported.

To learn more about IPv6 support in Microsoft Windows, see IPv6 for Microsoft Windows:

Frequently Asked Questions.

Return to top

IPv6 address basics

An IPv6 address is 128-bits long. The address is described by using colon-hexadecimal notation. Colon-hexadecimal notation describes the 128-bit address by using eight 16-bit, 4-digit hexadecimal numbers separated by the colon character (:). An example of an IPv6 address in colon-hexadecimal notation is 2001:0DB8:0000:0000:02AA:00FF:C0A8:640A.

You can express an IPv6 address by using the following methods:

- **Suppress leading zeros** You can omit the leading zeros in any of the eight 4-digit hexadecimal numbers in an IPv6 address.
- **Double-colon compression** You can use two colons (::) to represent contiguous 16-bit hexadecimal digits that contain all zeros. These all-zero digits may exist at the beginning, middle, or end of the IPv6 address. You can only use double-colon compression one time in an IPv6 address.
- **Trailing dotted-decimal notation** You may express the last 32 bits at the end of an IPv6 address in dotted-decimal notation by separating the 8-bit digits with a period (.). Trailing dotted-decimal notation is frequently used with IPv4-compatible addresses.

The following table provides examples of the IPv6 address notation and the equivalent IPv6 address syntax.

IPv6 address notation and syntax

IPv6 address notation	IPv6 address syntax
Full IPv6 address	2001:0DB8:0000:0000:02AA:00FF:C0A8:640A
IPv6 address that uses suppressed leading zeros	2001:DB8:0:0:2AA:FF:C0A8:640A
IPv6 address that uses double-colon compression	2001:DB8::2AA:FF:C0A8:640A
IPv6 address that uses trailing dotted-decimal notation	2001:DB8::2AA:FF:192.168.100.10

IPv6 addresses are categorized into the following types:

- **Unicast address** A packet is delivered to one interface.
- **Multicast address** A packet is delivered to multiple interfaces.
- **Anycast address** A packet is delivered to the nearest of multiple interfaces. The distance between interfaces is defined by the routing cost.

IPv6 unicast addresses have the following possible scopes:

- **Link local** The scope of the IPv6 address is the local subnet. IPv6 link local addresses are comparable to IPv4 link local addresses used in Automatic Private IP Addressing (APIPA).
- **Site local** The scope of the IPv6 address is the local organization. Site local addresses were deprecated by RFC 3879 and replaced by unique local addresses as defined in RFC 4193. IPv6 site local addresses and IPv6 unique local addresses are comparable to IPv4 private IP addresses.
- **Global** The scope of the IPv6 address is the whole world. IPv6 global addresses are comparable to IPv4 public IP addresses.

The following table provides a comparison of IPv4 elements and IPv6 elements.

IPv4 vs. IPv6 elements

Item	IPv4	IPv6
Private IP address	10.0.0.0/8 172.16.0.0/12 192.168.0.0/16	FD00::/8
Link local address	169.254.0.0/16	FE80::/64
Loopback address	127.0.0.1	::1
Unspecified address	0.0.0.0	::
Address resolution	Address Resolution Protocol (ARP)	Neighbor Discovery (ND)
Domain Name System (DNS) host name resolution	Address record (A record)	AAAA record or A6 record

For more information about IPv6 addressing, see IPv6 Address Types.

Supported IPv6 Address Input Formats

The following types of IPv6 address input formats are supported in Exchange 2013:

- A single IPv6 address
- An IPv6 address range
- An IPv6 address together with a subnet mask
- An IPv6 address together with a subnet mask that uses Classless Interdomain Routing (CIDR) notation

The following table provides examples of the acceptable IPv6 address input formats in Exchange 2013.

IPv6 address examples

Type	Example of an IPv6 address
Single address	2001:DB8::2AA:FF:C0A8:640A
Address range	2001:DB8::2AA:FF:C0A8:640A-2001:DB8::2AA:FF:C0A8:6414
Address together with subnet mask	2001:DB8::2AA:FF:C0A8:640A(FFFF:FFFF:FFFF:FFFF::)
Address together with subnet mask that uses CIDR notation	2001:DB8::2AA:FF:C0A8:640A/64

In Exchange 2013, the following input formats are supported:

- Suppression of leading zeros
- Double-colon compression
- Trailing dotted-decimal notation

Return to top

Exchange 2013 language support

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-08*

Microsoft Exchange Server 2013 has enhanced language support for both servers and clients. This topic lists the languages that are available for both servers and clients in Exchange 2013.

Supported server languages for Exchange 2013

The following server languages are supported and available for Exchange 2013:

- Chinese (Simplified)
- Chinese (Traditional)
- English
- French
- German
- Italian
- Japanese

- Korean
- Portuguese
- Russian
- Spanish

Supported client languages for Exchange 2013

The following client languages are supported and available for Exchange 2013:

- Amharic
- Arabic
- Basque (Basque)
- Bengali (India)
- Bulgarian
- Catalan
- Chinese (Simplified)
- Chinese (Traditional)
- Croatian
- Czech
- Danish
- Dutch
- English
- Estonian
- Filipino (Philippines)
- Finnish
- French
- Galician
- German
- Greek
- Gujarati
- Hebrew
- Hindi
- Hungarian
- Icelandic
- Indonesian
- Italian
- Japanese
- Kannada
- Kazakh
- Kiswahili
- Korean
- Latvian
- Lithuanian

- Malay (Brunei Darussalam)
- Malay (Malaysia)
- Malayalam
- Marathi
- Norwegian (Bokmål)
- Norwegian (Nynorsk)
- Oriya
- Persian
- Polish
- Portuguese (Brazil)
- Portuguese (Portugal)
- Romanian
- Russian
- Serbian (Cyrillic, Serbia)
- Serbian (Latin)
- Slovak
- Slovenian
- Spanish
- Swedish
- Tamil
- Telugu
- Thai
- Turkish
- Ukrainian
- Urdu
- Vietnamese
- Welsh

Exchange 2013 readiness checks

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-10*

The topics in this section provide details about the readiness checks Microsoft Exchange Server 2013 is performing when Exchange is installed. Readiness checks ensure that your Active Directory forest and Exchange servers are ready for Exchange 2013. Each readiness check topic describes the actions that you can take to resolve issues that are found when the readiness checks are run. You should only perform the steps outlined in a readiness check topic if that readiness check was displayed during setup.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

AD LDS directory exists in default location

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-12*

Microsoft Exchange Server 2013 Setup can't continue because the attempt to install Active Directory Lightweight Directory Services (AD LDS) failed.

An older installation of AD LDS exists in the default location. Setup can't perform a new AD LDS install in an existing AD LDS directory structure.

To resolve this issue, remove the existing AD LDS directory and then run Setup again.

For more information about AD LDS directory management, see Administering AD LDS Directory Partitions.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Duplicate Microsoft Exchange System Objects container exists in Active Directory

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-02-18*

Microsoft Exchange Server 2013 Setup can't continue because it found a duplicate Microsoft Exchange System Objects container in Active Directory Domain Naming context. When Setup finds a duplicate Microsoft Exchange System Objects container, you must delete the duplicate container before Setup can continue. When a duplicate Microsoft Exchange System Objects container exists, you can't solve the problem by running **DomainPrep** again. You must identify and delete the duplicate Microsoft Exchange System Objects container.

To resolve this issue, do the following:

1. Log on to the domain controller with administrative credentials.
2. In **Administrative Tools**, click **Active Directory Users and Computers**.
3. In the **Active Directory Users and Computers** management console pane, click **View** from the toolbar menu and then select **Advanced Features**.
4. Locate the duplicate Microsoft Exchange System Objects container.
5. Verify the duplicate Microsoft Exchange System Objects container doesn't contain valid Active Directory objects.
6. Right-click the duplicate Microsoft Exchange System Objects container, and then click **Delete**.
7. Confirm the deletion by clicking **Yes** in the Active Directory dialog box.

Note:
If you want the change to be replicated immediately, you must manually initiate replication between domain controllers.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Client Access server role is already installed

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-13*

Microsoft Exchange Server 2013 Setup has detected that you're attempting to install the Client Access server role; however, the role is already installed on the computer.

If you want to reinstall the Client Access server role, you must first uninstall Exchange and then install the Client Access server role.

You may also receive this message if a previous installation of Exchange didn't complete

successfully. If this happens, uninstall Exchange and then reinstall the Client Access server role. If the installation continues to fail, do the following:

- Make sure that your organization and the computer you're installing Exchange on meet the Exchange 2013 system requirements.
- Make sure that you've installed all the prerequisites required by Exchange, as described in Exchange 2013 prerequisites.
- Read the Release notes for Exchange 2013.
- View the Exchange Setup log located in <*installation drive*>:\ExchangeSetupLogs \ExchangeSetup.log, and look for errors or warnings that may have occurred during Setup.
- Search the Exchange Server forums. In your search query, specify the title of this topic.
- Search the Internet using your favorite search engine. Be sure to include the title of this topic in your search.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Active Directory does not exist or cannot be contacted

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-08*

Microsoft Exchange Server 2013 Setup can't continue because it can't contact a valid Active Directory directory service site. Setup requires that the server you're installing Exchange on is able to locate the configuration naming context in Active Directory.

To resolve this issue, verify that the user account running Exchange Setup is an Active Directory user and then run Setup again. If this doesn't resolve the issue, follow the guidance about using the dcdiag.exe and repadmin.exe support tools from the topics below to further diagnose the problem.

For more information about Active Directory troubleshooting and configuration for Exchange, see the following topics:

- Prepare Active Directory and domains
- Troubleshooting Active Directory Domain Services
- Configuring a Computer for Troubleshooting
- Troubleshooting Active Directory Replication Problems
- Monitoring and Troubleshooting Active Directory Replication Using Repadmin

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

The local computer isn't joined to an Active Directory domain

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-03-22*

Microsoft Exchange Server 2013 Setup can't continue because it detected that the local computer isn't a member of an Active Directory domain. You must join the local computer to an Active Directory domain before you can install Exchange Server 2013. You may also see this message if you log into a local user account on the computer instead of a domain user account with sufficient administrative rights to install Exchange 2013.

For more information, see Exchange 2013 system requirements.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Installation of the first Exchange server in the organization can't be delegated

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-03-27*

Microsoft Exchange Server 2013 Setup can't continue because the logged-on user doesn't have the account permissions that are required to install the first Exchange 2013 server in the organization.

Although Exchange 2013 Setup allows using delegation to install successive server roles, Setup requires that the user who is logged on is a member of the Enterprise Admins Windows security group when the first Exchange 2013 server in the organization is installed. This is required because

Exchange 2013 Setup creates and configures objects in the Exchange Organization container in Active Directory during installation.

Note:
If you haven't prepared the Active Directory schema for Exchange 2013, the logged-on user must also be a member of the Schema Admins Windows security group. Alternately, another user who's a member of the Schema Admins Windows group can prepare the Active Directory schema before Exchange 2013 is installed.

To resolve this issue, add the logged-on user as a member of the Enterprise Admins security group. Or, log on to an account that's a member of the Enterprise Admins security group. Then run Exchange 2013 Setup again.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Installation of the first Exchange server in the organization can't be delegated

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-03-27*

Microsoft Exchange Server 2013 Setup can't continue because the logged-on user doesn't have the account permissions that are required to install the first Exchange 2013 server in the organization.

Although Exchange 2013 Setup allows using delegation to install successive server roles, Setup requires that the user who is logged on is a member of the Enterprise Admins Windows security group when the first Exchange 2013 server in the organization is installed. This is required because Exchange 2013 Setup creates and configures objects in the Exchange Organization container in Active Directory during installation.

Note:
If you haven't prepared the Active Directory schema for Exchange 2013, the logged-on user must also be a member of the Schema Admins Windows security group. Alternately, another user who's a member of the Schema Admins Windows group can prepare the Active Directory schema before Exchange 2013 is installed.

To resolve this issue, add the logged-on user as a member of the Enterprise Admins security group. Or, log on to an account that's a member of the Enterprise Admins security group. Then run

Exchange 2013 Setup again.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Installation of the first Exchange server in the organization can't be delegated

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-03-27*

Microsoft Exchange Server 2013 Setup can't continue because the logged-on user doesn't have the account permissions that are required to install the first Exchange 2013 server in the organization.

Although Exchange 2013 Setup allows using delegation to install successive server roles, Setup requires that the user who is logged on is a member of the Enterprise Admins Windows security group when the first Exchange 2013 server in the organization is installed. This is required because Exchange 2013 Setup creates and configures objects in the Exchange Organization container in Active Directory during installation.

Note:

If you haven't prepared the Active Directory schema for Exchange 2013, the logged-on user must also be a member of the Schema Admins Windows security group. Alternately, another user who's a member of the Schema Admins Windows group can prepare the Active Directory schema before Exchange 2013 is installed.

To resolve this issue, add the logged-on user as a member of the Enterprise Admins security group. Or, log on to an account that's a member of the Enterprise Admins security group. Then run Exchange 2013 Setup again.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Installation of the first Exchange server in the organization can't be delegated

Applies to: *Exchange Server*

Topic Last Modified: *2013-03-27*

Microsoft Exchange Server 2013 Setup can’t continue because the logged-on user doesn’t have the account permissions that are required to install the first Exchange 2013 server in the organization.

Although Exchange 2013 Setup allows using delegation to install successive server roles, Setup requires that the user who is logged on is a member of the Enterprise Admins Windows security group when the first Exchange 2013 server in the organization is installed. This is required because Exchange 2013 Setup creates and configures objects in the Exchange Organization container in Active Directory during installation.

Note:
If you haven’t prepared the Active Directory schema for Exchange 2013, the logged-on user must also be a member of the Schema Admins Windows security group. Alternately, another user who’s a member of the Schema Admins Windows group can prepare the Active Directory schema before Exchange 2013 is installed.

To resolve this issue, add the logged-on user as a member of the Enterprise Admins security group. Or, log on to an account that’s a member of the Enterprise Admins security group. Then run Exchange 2013 Setup again.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you’re looking for? Please take a minute to send us feedback about the information you were hoping to find.

Installation of the first Exchange server in the organization can't be delegated

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-03-27*

Microsoft Exchange Server 2013 Setup can't continue because the logged-on user doesn't have the account permissions that are required to install the first Exchange 2013 server in the organization.

Although Exchange 2013 Setup allows using delegation to install successive server roles, Setup requires that the user who is logged on is a member of the Enterprise Admins Windows security group when the first Exchange 2013 server in the organization is installed. This is required because Exchange 2013 Setup creates and configures objects in the Exchange Organization container in Active Directory during installation.

Note:
If you haven't prepared the Active Directory schema for Exchange 2013, the logged-on user must also be a member of the Schema Admins Windows security group. Alternately, another user who's a member of the Schema Admins Windows group can prepare the Active Directory schema before Exchange 2013 is installed.

To resolve this issue, add the logged-on user as a member of the Enterprise Admins security group. Or, log on to an account that's a member of the Enterprise Admins security group. Then run Exchange 2013 Setup again.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Active Directory functional level isn't Windows Server 2003 or later

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-04-10*

Microsoft Exchange Server 2013 Setup can't continue because it detected that the forest functional level of the current Active Directory forest isn't Windows Server 2003 native or later. Before you can install Exchange 2013, you must raise the forest functional level to Windows Server 2003 or later.

For information about how to raise the forest functional level of the current Active Directory forest to Windows Server 2003 or later, see Raise the Forest Functional Level.

For more information about Active Directory functional levels, see the following topics:

- What are Active Directory Functional Levels?
- How Active Directory Functional Levels Work

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server,

Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Cannot write to the Exchange organization container

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-08*

Microsoft Exchange Server 2013 Setup can't continue because the logged-on user doesn't have the account permissions that are required to write to the organization container in the Active Directory directory service.

Setup requires that the user who is logged on when Exchange 2013 is installed has permission to create and modify objects in Active Directory. If you're running Exchange 2013 Setup in your organization for the first time, the account you use must be a member of the Schema Admins and Enterprise Admins groups. These permissions are required because Active Directory is prepared for Exchange 2013 the first time Setup is run. After Active Directory is prepared, the account you use to install additional Exchange 2013 servers must be a member of the Organization Management management role group.

To resolve this issue, grant the logged-on user the appropriate permissions, or log on with an account that has those permissions and run Exchange 2013 Setup again.

Important:
Cross-forest installation of Exchange 2013 isn't supported. Use an account that is a member of the Active Directory forest where you're installing Exchange 2013.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Global updates required

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-12*

Microsoft Exchange Server 2013 Setup can't continue because the logged-on user doesn't have the account permissions that are required to make global updates to Active Directory.

Setup requires that the user who is logged on when Exchange 2013 is installed has permission to create and modify objects in Active Directory. If you're running Exchange 2013 Setup in your organization for the first time, the account you use must be a member of the Schema Admins and Enterprise Admins groups. These permissions are required because Active Directory is prepared for Exchange 2013 the first time Setup is run. After Active Directory is prepared, the account you use to install additional Exchange 2013 servers must be a member of the Organization Management management role group.

To resolve this issue, grant the logged-on user the appropriate permissions, or log on with an account that has those permissions and run Exchange 2013 Setup again.

Important:
Cross-forest installation of Exchange 2013 isn't supported. Use an account that is a member of the Active Directory forest where you're installing Exchange 2013.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

The Host record for the local computer cannot be found in the DNS database

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-04-10*

Microsoft Exchange Server 2013 Setup can't continue because the Host (A) record for this computer can't be found in the Domain Name System (DNS) database.

Exchange 2013 Setup requires that the local computer have a valid HOST (A) record registered with the authoritative DNS database for the domain.

Exchange depends on DNS Host (A) records for the IP Address of its next internal or external destination server.

To resolve this issue:

- Verify that the local TCP/IP configuration points to the correct DNS server. For more information, see Configure TCP/IP settings.
- Use Nslookup.exe to verify that the Host (A) record exists on the DNS server. For more information, see To verify A resource records exist in DNS.

For information about DNS name resolution, troubleshooting, and Host (A) records, see the following:

- Troubleshooting DNS
- Managing resource records

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Installation on domain controllers is not supported with Active Directory split permissions

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-12*

Microsoft Exchange Server 2013 Setup has detected that you're attempting to run Setup on an Active Directory domain controller and one of the following is true:

- The Exchange organization is already configured for Active Directory split permissions.
- You selected the Active Directory split permissions option in Exchange 2013 Setup.

The installation of Exchange 2013 on domain controllers isn't supported when the Exchange organization is configured for Active Directory split permissions.

If you want to install Exchange 2013 on a domain controller, you must configure the Exchange organization for Role Based Access Control (RBAC) split permissions or shared permissions.

Important:
We don't recommend installing Exchange 2013 on Active Directory domain controllers. For more information, see Installing Exchange on a domain controller is not recommended.

If you want to continue using Active Directory split permissions, you must install Exchange 2013 on a member server.

For more information about split and shared permissions in Exchange 2013, see the following

topics:

Understanding split permissions

Configure Exchange 2013 for split permissions

Configure Exchange 2013 for shared permissions

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

The current account isn't logged into an Active Directory domain

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-04-10*

Microsoft Exchange Server 2013 Setup can't continue because it detected that the current account isn't logged on to an Active Directory domain. You must log in using an Active Directory account that has the permissions required to install Exchange Server 2013.

Setup requires that the user who is logged on when Exchange 2013 is installed has permission to create and modify objects in Active Directory. If you're running Exchange 2013 Setup in your organization for the first time, the account you use must be a member of the Schema Admins and Enterprise Admins groups. These permissions are required because Active Directory is prepared for Exchange 2013 the first time Setup is run. After Active Directory is prepared, the account you use to install additional Exchange 2013 servers must be a member of the Organization Management management role group.

To resolve this issue, grant the logged-on user the appropriate permissions, or log on with an account that has those permissions and run Exchange 2013 Setup again.

Important:
Cross-forest installation of Exchange 2013 isn't supported. Use an account that is a member of the Active Directory forest where you're installing Exchange 2013.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

The computer needs to be restarted before Setup can continue

Applies to: *Exchange Server*

Topic Last Modified: *2014-07-17*

Microsoft Exchange Server 2013 Setup can't continue because it detected that the local computer needs to be restarted to complete the installation of other programs or updates.

Why is this happening?

When programs or updates are installed, they sometimes need to make changes to files or other resources on the computer that can only be made during Windows startup. To tell Windows that it needs to make changes during startup, a program or update makes a change to one or both of the following Windows registry keys:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Updates**UpdateExeVolatile**
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager **PendingFileRenameOperations**

This table shows all of the values that you might see in these keys.

Value of UpdateExeVolatile	Value of PendingFileRenameOperations	Results
Greater than 3	Not applicable	An invalid value was returned.
3	Not applicable	An update that's been installed and another update that's been removed are waiting for the computer to be restarted. If it's a security update, the computer might be at risk until it's restarted.
2	Not applicable	An update that's been installed is waiting for the computer to be restarted. If it's a security

		update, the computer might be at risk until it's restarted.
1	Not applicable	An update removal is waiting for the computer to be restarted.
0 or missing	Exists and contains value	The installation of an update didn't finish. The system needs to be restarted to finish the installation. If it's a security update, the computer might be at risk until it's restarted.
0 or missing	Empty	The computer doesn’t need to be restarted.

Exchange Setup won't continue and it'll show this error if the **UpdateExeVolatile** registry key exists and has as value other than 0, or if the **PendingFileRenameOperations** registry key isn't empty. If you see this error, it usually just means that the computer hasn't been restarted since the last time a program or update was installed. However, sometimes the installation of a program or update doesn't complete properly and the values in these keys aren't reset. The values in these keys are considered to be *orphaned*. When this happens, Setup will keep telling you that the computer needs to be restarted even if you already have.

How do I fix it?

There are a couple things you can do to fix this error.

Restart the computer

Restart the computer, and then run Exchange 2013 Setup again. The most common reason for why you'll see this error is because the computer hasn't been restarted since the last time a program or update was installed. Restarting the computer will often fix this error and let Setup continue. If it does, you’re all set and you can get back to installing Exchange.

Remove the orphaned values from the registry

If restarting the computer doesn't fix this error, another program or update probably didn't finish installing properly. When this happens, you'll need to use these steps to remove the orphaned values from the **UpdateExeVolatile** and **PendingFileRenameOperations** registry keys before Setup can continue.

Before you go any further, a note about making changes to the Windows registry (or just "registry"). Making incorrect changes to the registry could cause serious problems, and could even force you to reinstall Windows. Before you make any changes to the registry, make sure you have a working backup of your computer.

1. Press the Windows key + 'R' to open the **Run** window.
2. In **Open**, type `regedit.exe`, and press Enter to open Registry Editor.
3. Navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Updates**.
4. If the **UpdateExeVolatile** key exists and contains anything other than 0, do the following:
   a. In the right navigation pane, double-click the **UpdateExeVolatile** key.
   b. Replace the value in the key with **0**, and then click **OK**.
5. Navigate to **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager**.
6. If the **PendingFileRenameOperations** key contains any value, right-click **PendingFileRenameOperations** in the navigation pane, and then click **Delete**.
7. Close Registry Editor.

After you've finished these steps, this error should no longer appear when you run Exchange Setup.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

The logged-on user is not a member of the Schema Admins group

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-12*

Microsoft Exchange Server 2013 Setup can't continue because the user account performing the Active Directory schema update process isn't a member of the Schema Admins and Enterprise Admins groups.

Setup requires that the user who is logged on when Exchange 2013 is installed has permission to create and modify objects in Active Directory. If you're running Exchange 2013 Setup in your organization for the first time, the account you use must be a member of the Schema Admins and Enterprise Admins groups. These permissions are required because Active Directory is prepared for Exchange 2013 the first time Setup is run. After Active Directory is prepared, the account you use to install additional Exchange 2013 servers must be a member of the Organization Management management role group.

To resolve this issue, grant the logged-on user the appropriate permissions, or log on with an account that has those permissions and run Exchange 2013 Setup again.

Important:
Cross-forest installation of Exchange 2013 isn't supported. Use an account that is a member of the Active Directory forest where you're installing Exchange 2013.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

UCMA 4.0, Core Runtime not installed

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-12-10*

Microsoft Exchange Server 2013 Setup can't continue because the local computer requires a software update. You'll need to install this update before Exchange 2013 Setup can continue.

Exchange 2013 Setup requires that the Unified Communications Managed API 4.0 Runtime update be installed on the computer before installation can continue.

Download and install the 64-bit update from the following URL, and then click **retry** on the **Readiness Checks** page.

Note:
If this update requires a reboot to complete installation, you'll need to exit Exchange 2013 Setup, reboot, and then start Setup again.

Unified Communications Managed API 4.0 Runtime

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Cannot remove mailbox database

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-08*

Microsoft Exchange Server 2013 Setup can't continue because it can't remove a user mailbox database from the local server without incurring potential data loss.

Exchange 2013 Setup determines whether all mailbox databases have been removed from the server before the Mailbox server role is removed. However, user mailboxes might still remain on the server.

To resolve this issue, move any mailboxes on the server to another Exchange server or, if the mailboxes and the data contained within them are no longer required, disable the mailboxes. Then run Exchange 2013 Setup again.

- For more information about how to identify a mailbox in the database, see Get-Mailbox.
- For more information about how to move a mailbox, see Mailbox moves in Exchange 2013.
- For more information about how to disable a mailbox, see Disable-Mailbox.
- For more information about how to remove a mailbox database, see Manage mailbox databases in Exchange 2013.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Installing Exchange on a domain controller is not recommended

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-03-22*

Microsoft Exchange Server 2013 Setup has detected that the computer you're attempting to install Exchange 2013 on is an Active Directory domain controller. Installing Exchange 2013 on a domain controller isn't recommended.

If you install Exchange 2013 on a domain controller, be aware of the following issues:

- Configuring Exchange 2013 for Active Directory split permissions isn't supported.
- The Exchange Trusted Subsystem universal security group (USG) is added to the Domain Admins group when Exchange is installed on a domain controller. When this occurs, all Exchange servers in the domain are granted domain administrator rights in that domain.
- Exchange Server and Active Directory are both resource-intensive applications. There are

performance implications to be considered when both are running on the same computer.
- You must make sure that the domain controller Exchange 2013 is installed on is a global catalog server.
- Exchange services may not start correctly when the domain controller is also a global catalog server.
- System shutdown will take considerably longer if Exchange services aren't stopped before shutting down or restarting the server.
- Demoting a domain controller to a member server isn't supported.
- Running Exchange 2013 on a clustered node that is also an Active Directory domain controller isn't supported.

We recommend that you install Exchange 2013 on a member server.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

KB2619234 update not installed

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-08*

Microsoft Exchange Server 2013 Setup can't continue because the local computer requires a software update. You'll need to install this update before Exchange 2013 Setup can continue.

Exchange 2013 Setup requires an update to Windows that allows Outlook Anywhere (formerly known as RPC over HTTP) to work correctly.

Download and install the 64-bit update from the following URL, and then click **retry** on the **Readiness Checks** page.

Note:
If this update requires a reboot to complete installation, you'll need to exit Exchange 2013 Setup, reboot, and then start Setup again.

Microsoft Knowledge Base article KB2619234, A hotfix is available to enable the Association Cookie/GUID that is used by RPC over HTTP to also be used at the RPC layer in Windows 7 and in Windows Server 2008 R2)

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Installation of the first Exchange server in the organization can't be delegated

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-03-27*

Microsoft Exchange Server 2013 Setup can't continue because the logged-on user doesn't have the account permissions that are required to install the first Exchange 2013 server in the organization.

Although Exchange 2013 Setup allows using delegation to install successive server roles, Setup requires that the user who is logged on is a member of the Enterprise Admins Windows security group when the first Exchange 2013 server in the organization is installed. This is required because Exchange 2013 Setup creates and configures objects in the Exchange Organization container in Active Directory during installation.

Note:
If you haven't prepared the Active Directory schema for Exchange 2013, the logged-on user must also be a member of the Schema Admins Windows security group. Alternately, another user who's a member of the Schema Admins Windows group can prepare the Active Directory schema before Exchange 2013 is installed.

To resolve this issue, add the logged-on user as a member of the Enterprise Admins security group. Or, log on to an account that's a member of the Enterprise Admins security group. Then run Exchange 2013 Setup again.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

No Exchange 2010 or Exchange 2007 servers detected

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-08*

Microsoft Exchange Server 2013 Setup displayed this warning because no Exchange Server 2010 or Exchange Server 2007 server roles exist in the organization.

Caution:
If you continue with Exchange Server 2013 installation, you won't be able to add Exchange 2010 or Exchange 2007 servers to the organization at a future date.

Before deploying Exchange 2013, consider the following factors that may require you to deploy Exchange 2010 or Exchange 2007 servers prior to deploying Exchange 2013:

- **Third-party or in-house developed applications** Applications developed for earlier versions of Exchange may not be compatible with Exchange 2013. You may need to maintain Exchange 2010 or Exchange 2007 servers to support these applications.
- **Coexistence or migration requirements** If you plan on migrating mailboxes into your organization, some solutions may require the use of Exchange 2010 or Exchange 2007 servers.

If you decide that you need to deploy Exchange 2010 or Exchange 2007 servers, you must do so before you deploy Exchange 2013. Active Directory must be prepared for each Exchange version in the following order:

1. Exchange 2007
2. Exchange 2010
3. Exchange 2013

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

The computer needs to be restarted before Setup can continue

Applies to: *Exchange Server*

Topic Last Modified: *2014-07-25*

Microsoft Exchange Server 2013 Setup can't continue because it detected that the local computer needs to be restarted to complete the installation of other programs or updates.

Why is this happening?

When programs or updates are installed, they sometimes need to make changes to files or other resources on the computer that can only be made during Windows startup. To tell Windows that it needs to make changes during startup, a program or update makes a change to one or both of the following Windows registry keys:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Updates**UpdateExeVolatile**
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager **PendingFileRenameOperations**

This table shows all of the values that you might see in these keys.

Value of UpdateExeVolatile	Value of PendingFileRenameOperations	Results
Greater than 3	Not applicable	An invalid value was returned.
3	Not applicable	An update that's been installed and another update that's been removed are waiting for the computer to be restarted. If it's a security update, the computer might be at risk until it's restarted.
2	Not applicable	An update that's been installed is waiting for the computer to be restarted. If it's a security update, the computer might be at risk until it's restarted.
1	Not applicable	An update removal is waiting for the computer to be restarted.
0 or missing	Exists and contains value	The installation of an update didn't finish. The system needs to be restarted to finish the installation. If it's a security

		update, the computer might be at risk until it's restarted.
0 or missing	Empty	The computer doesn't need to be restarted.

Exchange Setup won't continue and it'll show this error if the **UpdateExeVolatile** registry key exists and has as value other than 0, or if the **PendingFileRenameOperations** registry key isn't empty. If you see this error, it usually just means that the computer hasn't been restarted since the last time a program or update was installed. However, sometimes the installation of a program or update doesn't complete properly and the values in these keys aren't reset. The values in these keys are considered to be *orphaned*. When this happens, Setup will keep telling you that the computer needs to be restarted even if you already have.

How do I fix it?

There are a couple things you can do to fix this error.

Restart the computer

Restart the computer, and then run Exchange 2013 Setup again. The most common reason for why you'll see this error is because the computer hasn't been restarted since the last time a program or update was installed. Restarting the computer will often fix this error and let Setup continue. If it does, you're all set and you can get back to installing Exchange.

Remove the orphaned values from the registry

If restarting the computer doesn't fix this error, another program or update probably didn't finish installing properly. When this happens, you'll need to use these steps to remove the orphaned values from the **UpdateExeVolatile** and **PendingFileRenameOperations** registry keys before Setup can continue.

Before you go any further, a note about making changes to the Windows registry (or just "registry"). Making incorrect changes to the registry could cause serious problems, and could even force you to reinstall Windows. Before you make any changes to the registry, make sure you have a working backup of your computer.

1. Press the Windows key + 'R' to open the **Run** window.
2. In **Open**, type `regedit.exe`, and press Enter to open Registry Editor.
3. Navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Updates**.
4. If the **UpdateExeVolatile** key exists and contains anything other than 0, do the following:
   a. In the right navigation pane, double-click the **UpdateExeVolatile** key.
   b. Replace the value in the key with **0**, and then click **OK**.
5. Navigate to **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager**.
6. If the **PendingFileRenameOperations** key contains any value, right-click **PendingFileRenameOperations** in the navigation pane, and then click **Delete**.

7. Close Registry Editor.

After you've finished these steps, this error should no longer appear when you run Exchange Setup.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Exchange 2007 servers must be upgraded to Service Pack 3, Update Rollup 10

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-02-18*

Microsoft Exchange Server 2013 Setup can't continue because it detected one or more Exchange Server 2007 servers haven't been upgraded to Exchange Server 2007 Service Pack 3 (SP3) Roll Up 10 (RU10). Before you can install Exchange 2013, all Exchange 2007 servers in your organization must be upgraded to Exchange 2007 SP3 RU10. This requirement includes Exchange 2007 Edge Transport servers. For more information, see Upgrade from Exchange 2007 to Exchange 2013.

Important:
After you upgrade your Exchange 2007 Edge Transport servers to Exchange 2007 SP3 RU10, you must re-create the Edge subscription between your Exchange organization and each Edge Transport server to update their server version in Active Directory. For more information about re-creating Edge subscriptions in Exchange 2007, see Subscribing the Edge Transport Server to the Exchange Organization.

Exchange 2010 servers must be upgraded to Service Pack 3

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-01-28*

Microsoft Exchange Server 2013 Setup can't continue because it detected one or more Microsoft Exchange Server 2010 servers haven't been upgraded to Service Pack 3 (SP3) for Exchange Server 2010. Before you can install Exchange 2013, all Exchange 2010 servers in your organization must be upgraded to Exchange 2010 SP3. This requirement includes Exchange 2010 Edge Transport servers. For more information, see Upgrade from Exchange 2010 to Exchange 2013.

Important:

After you upgrade your Exchange 2010 Edge Transport servers to Exchange 2010 SP3, you must re-create the Edge subscription between your Exchange organization and each Edge Transport server to update their server version in Active Directory. For more information about re-creating Edge subscriptions in Exchange 2010, see Managing Edge Subscriptions.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Office 2010 Filter Pack not installed

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-12*

Microsoft Exchange Server 2013 Setup can't continue because the local computer requires a software update. You'll need to install this update before Exchange 2013 Setup can continue.

Exchange 2013 Setup requires that the Microsoft Office 2010 Filter Pack update be installed on the computer before installation can continue.

Download and install the 64-bit update from the following URL, and then click **retry** on the **Readiness Checks** page.

Note:

If this update requires a reboot to complete installation, you'll need to exit Exchange 2013 Setup, reboot, and then start Setup again.

Microsoft Office 2010 Filter Packs

Note:

You must also install the Microsoft Office 2010 Filter Pack Service Pack 1 64-bit update. For more information, see Office 2010 Filter Pack SP1 not installed.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server,

Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Office 2010 Filter Pack SP1 not installed

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-11-08*

Microsoft Exchange Server 2013 Setup can't continue because the local computer requires a software update. You'll need to install this update before Exchange 2013 Setup can continue.

Exchange 2013 Setup requires that the Microsoft Office 2010 Filter Pack Service Pack 1 update be installed on the computer before installation can continue.

Download and install the 64-bit update from the following URL, and then click **retry** on the **Readiness Checks** page.

Note:
If this update requires a reboot to complete installation, you'll need to exit Exchange 2013 Setup, reboot, and then start Setup again.

Service Pack 1 for Microsoft Office Filter Pack 2010 (KB2460041) 64-bit Edition

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Can't install Exchange 2013 in a forest containing Exchange 2000 or Exchange 2003 servers.

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-04-17*

Microsoft Exchange Server 2013 can't continue because one or more computers running Exchange 2000 Server or Exchange Server 2003 were found in the Active Directory forest. Before you can install Exchange 2013, all Exchange 2000 and Exchange 2003 servers must be removed from the forest. Mailboxes, public folders, and all other Exchange objects or components must be upgraded to either Exchange Server 2007 or Exchange Server 2010. You can't upgrade from Exchange 2000 or Exchange 2003 directly to Exchange 2013.

The path you need to follow to install Exchange 2013 in your organization depends on the version of Exchange you currently have installed in your forest. See the following table for more information.

If you have the following installed in your organization	**You must take this path to upgrade to Exchange 2013**
Exchange 2000	1. Install Exchange 2007 into your Exchange 2000 organization. 2. Configure Exchange 2007 and Exchange 2000 coexistence. 3. Migrate Exchange 2000 mailboxes, public folders, and other components to Exchange 2007. 4. Decommission and remove all Exchange 2000 servers. 5. Install Exchange 2013 into your Exchange 2007 organization. 6. Configure Exchange 2013 and Exchange 2007 coexistence. 7. Migrate Exchange 2007 mailboxes, public folders, and other components to Exchange 2013. 8. Decommission and remove all Exchange 2007 servers. For more information, see Upgrading to Exchange 2007 and Upgrade from Exchange 2007 to Exchange 2013.
Exchange 2003	1. Install Exchange 2010 into your Exchange 2003 organization. 2. Configure Exchange 2010 and Exchange 2003

	coexistence. 3. Migrate Exchange 2003 mailboxes, public folders, and other components to Exchange 2010. 4. Decommission and remove all Exchange 2003 servers. 5. Install Exchange 2013 into your Exchange 2010 organization. 6. Configure Exchange 2013 and Exchange 2010 coexistence. 7. Migrate Exchange 2010 mailboxes, public folders, and other components to Exchange 2013. 8. Decommission and remove all Exchange 2010 servers. For more information, see Exchange 2003 - Planning Roadmap for Upgrade and Coexistence and Upgrade from Exchange 2010 to Exchange 2013.

When upgrading to Exchange 2010 or Exchange 2013, you can use the Exchange Deployment Assistant to help you complete your deployment. For more information, see the following links:

- Exchange 2010 Deployment Assistant
- Exchange 2013 Deployment Assistant

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

An incompatible operating system was found

Applies to: *Exchange Server*

Topic Last Modified: *2014-01-30*

Microsoft Exchange Server 2013 Setup can't continue because it detected an incompatible operating system. You must install a compatible operating system on this computer before you install Exchange 2013. The following table shows the operating systems that are compatible with Exchange 2013.

Important:
Exchange 2013 doesn't support the Server Core installation option of Windows Server 2008 R2, Windows Server 2012, or Windows Server 2012 R2.

Supported operating systems for Exchange 2013

Component	Requirement
Mailbox, Client Access, and Edge Transport server roles	One of the following: • Windows Server 2012 R2 Standard or Datacenter[1] • Windows Server 2012 Standard or Datacenter • Windows Server 2008 R2 Standard with Service Pack 1 (SP1) • Windows Server 2008 R2 Enterprise with Service Pack 1 (SP1) • Windows Server 2008 R2 Datacenter RTM or later
Management tools	One of the following: • Windows Server 2012 R2 Standard or Datacenter[1] • Windows Server 2012 Standard or Datacenter • Windows Server 2008 R2 Standard with SP1 • Windows Server 2008 R2 Enterprise with SP1 • Windows Server 2008 R2 Datacenter RTM or later • 64-bit edition of Windows 8.1[2] • 64-bit edition of Windows 8 • 64-bit edition of Windows 7 with Service Pack 1

[1] Windows Server 2012 R2 is supported only with Exchange 2013 SP1 or later.

[2] Windows 8.1 is supported only with Exchange 2013 SP1 or later.

For more information, see Exchange 2013 system requirements.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

Exchange 2010 servers must be upgraded to Service Pack 3

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2013-01-28*

Microsoft Exchange Server 2013 Setup can't continue because it detected one or more Microsoft Exchange Server 2010 servers haven't been upgraded to Service Pack 3 (SP3) for Exchange Server 2010. Before you can install Exchange 2013, all Exchange 2010 servers in your organization must be upgraded to Exchange 2010 SP3. This requirement includes Exchange 2010 Edge Transport servers. For more information, see Upgrade from Exchange 2010 to Exchange 2013.

Important:
After you upgrade your Exchange 2010 Edge Transport servers to Exchange 2010 SP3, you must re-create the Edge subscription between your Exchange organization and each Edge Transport server to update their server version in Active Directory. For more information about re-creating Edge subscriptions in Exchange 2010, see Managing Edge Subscriptions.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

ExecutionPolicy GPO is defined

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2014-01-31*

Microsoft Exchange Server 2013 Setup can’t continue because it detected that the **ExecutionPolicy** Group Policy Object (GPO) defines one or both of the following policies:

- **MachinePolicy**
- **UserPolicy**

It doesn't matter how the policies have been defined. It only matters that they have been defined.

When you run Exchange 2013 Setup, Exchange stops and disables the Windows Management Instrumentation (WMI) service. When either of these policies are defined, the WMI service needs to be enabled to run a Windows PowerShell script. If the policies are defined and the WMI service is stopped, Setup will fail and the server will be left in an inconsistent state.

To allow Setup to continue, you need to temporarily remove any definition of **MachinePolicy** or **UserPolicy** in the **ExecutionPolicy** GPO.

For information on how to remove any definitions of **MachinePolicy** or **UserPolicy** in the **ExecutionPolicy** GPO, see Knowledge Base article KB981474.

Note:
Even though this Knowledge Base article was written for Exchange 2010, it also applies to Exchange 2013 cumulative updates and service packs.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you’re looking for? Please take a minute to send us feedback about the information you were hoping to find.

Primary DNS Suffix is missing

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2014-01-15*

Microsoft Exchange Server 2013 Setup can't continue because the primary domain name system (DNS) suffix for the computer you're installing Exchange on hasn't been configured.

To resolve this issue, add a primary DNS suffix on the computer using the steps below and then run Setup again.

Important:
Changing the computer name or primary DNS suffix after you install Exchange 2013 isn't supported.

1. Log on to the computer where you want to install the Edge Transport role as a user that's a member of the local Administrators group.
2. Open the **Control Panel** and then double-click **System**.
3. In the **Computer name, domain, and workgroup settings** section, click **Change settings**.
4. In the **System Properties** window, make sure the **Computer Name** tab is selected and then click **Change...**.
5. In **Computer Name/Domain Changes**, click **More...**.
6. In **Primary DNS suffix of this computer**, enter the DNS domain name for the Edge Transport server. For example, contoso.com.
7. Click **OK** to close each window.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Did you find what you're looking for? Please take a minute to send us feedback about the information you were hoping to find.

The Simple Mail Transport Protocol is currently installed_SMTPSvcInstalled

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because the Microsoft Windows Server™ 2003 Simple Mail Transfer Protocol (SMTP) service is installed on this computer.

Microsoft Exchange setup requires that the SMTP service not be installed on servers that are used for Exchange 2007.

To resolve this issue, uninstall the SMTP service and rerun Microsoft Exchange setup.

To uninstall the SMTP service by using Add or Remove a Windows Component in Control Panel

1. On the **Start** menu, click **Control Panel**.
2. Double-click **Add or Remove Programs**.
3. Click **Add/Remove Windows Components**.

4. In the **Components** list, select the **Application Server** check box and then click **Details**.
5. Select **Internet Information Services Manager** and then click **Details**.
6. Select **SMTP Service** and then click to clear the check box.
7. Click **OK** two times to return to the **Components** list and then click **Next**.
8. Click **Finish** when the SMTP service is uninstalled.

SMTP Addressing Format Not Supported_SMTPAddressLiteral

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 and Exchange Server 2010 setup cannot continue because the specified recipient policy uses an unsupported Simple Mail Transfer Protocol (SMTP) address format.

Exchange 2007 and Exchange 2010 setup requires that all SMTP addresses used for e-mail address policies not contain IP address literals, for example: *user@[10.10.1.1]*.

To resolve this issue, change the value of the SMTP address in the recipient policy so that it does not contain an IP address literal. Replace brackets ([]) and numbers (10.10.1.1) of the IP address literal with the Domain Name System (DNS) naming format, for example: *user@contoso.com*, and then rerun Exchange setup.

For more information about managing recipient policies in Exchange Server 2007, see "Managing E-Mail Address Policies" (http://go.microsoft.com/fwlink/?LinkId=86653).

For more information about managing recipient policies in Exchange Server 2010, see "Managing E-Mail Address Policies" (http://go.microsoft.com/fwlink/?LinkId=179519).

The World Wide Web Publishing

Service is disabled or missing_ShouldReRunSetupForW3SVC

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because its attempt to install the Mailbox Server or Client Access role found that the World Wide Web Publishing Service is either disabled or not installed on this computer.

Exchange 2007 setup requires the computer that you are installing Microsoft Exchange to have the World Wide Web Publishing Service installed and set to something other than disabled.

To resolve this issue, verify that the World Wide Web Publishing service is installed and not disabled on the local computer, and then rerun Microsoft Exchange setup.

To verify that the World Wide Web Publishing Service is installed and not disabled

1. Right-click **My Computer** on the desktop, and then click **Manage**.
2. Expand the **Services and Applications** node, and then click the **Services** node.
3. In the right pane, locate the **World Wide Web Publishing Service**.

If the **World Wide Web Publishing Service** is not displayed in the list of services installed, follow the steps in the procedure below to install it.

4. If the **World Wide Web Publishing Service** is displayed but has a status other than **Started**, continue with the steps below to start it.
5. Right-click **World Wide Web Publishing Service**, and then click **Properties**.
6. Verify the **Startup Type** is **Automatic** and the **Service status** is set to **Started**.
7. Click **Apply**, and then click **OK**.

To install the World Wide Web Publishing Service

1. On the **Start** menu, select **Settings**, **Control Panel**, and then click **Add or Remove Programs**
2. Click **Add/Remove Windows Components**.
3. In the **Components** list, select the **Application Server** check box, and then click **Details**.
4. Select **Internet Information Services Manager**, and then click **Details**.
5. Select **World Wide Web Service**, and then select the check box.
6. Click **OK** two times to return to the **Components** list, and then click **Next**.

7. Click **Finish** when the IIS service is installed.

This Server is the Source for a Send Connector_ServerIsSourceForSendConnector

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because its attempt to remove the Hub Transport server role failed. The local computer is the source for one or more Send connectors in the Exchange organization.

A Send connector represents a logical gateway through which outbound messages are sent.

Exchange 2007 setup requires that all Send connectors for an Exchange organization be moved or deleted from the Hub Transport server computer before the server role can be deleted.

To resolve this issue, move or delete all Send connectors from the local computer and then rerun setup.

For more information about modifying or removing Send connectors, see the following topics in the Exchange Server 2007 product documentation:

- "How to Remove a Send Connector" (http://go.microsoft.com/fwlink/?LinkId=86655).
- "How to Modify the Configuration of a Send Connector" (http://go.microsoft.com/fwlink/?LinkId=86656).

The local computer is responsible for expanding group

membership_ServerIsGroupExpansionServer

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because its attempt to uninstall a Hub Transport role responsible for expanding group membership failed.

Exchange 2007 setup requires that distribution list expansion be removed from the current Bridgehead server before the Hub Transport role can be uninstalled.

The expansion of distribution lists enables identification of individual recipients who belong to the distribution list to be identified, or the identification of additional distribution lists for expansion. An expanded distribution list can return the path for any required delivery status notification (DSN). DSNs notify the Microsoft Exchange administrator or e-mail sender of the status of a particular e-mail message. Additionally, distribution list expansion identifies whether Out of Office messages or automatically generated replies should be sent to the sender of the original message.

To resolve this issue, move distribution group expansion to another server and rerun Microsoft Exchange setup.

To change the expansion server for a distribution group or dynamic distribution group

1. Open the Exchange Management Console.
2. In the console tree, expand **Recipient Configuration**.
3. In the console tree, click **Distribution Group**.
4. Create a filter to view all distribution groups and dynamic distribution groups that use the current Bridgehead server as an expansion server.
   a. In the action pane, click **Create Filter**.
   b. In the property drop-down list, click **Expansion Server**.
   c. In the operator drop-down list, click **Equals**.
   d. At the value box, click the **Browse** button to select the Bridgehead server that currently acting as the expansion server.

Note:
The following step is optional.

1. Click **Add Expression** to specify additional filter criteria. Only messages that meet all filter criteria will be displayed.
2. Click **Apply Filter**. The results that meet the filter criteria are displayed.
1. In the results pane, click the distribution group you want to change the expansion server for, and then click **Properties** in the action pane.
2. On **Properties**, click the **Advanced** tab.
3. In the Expansion server drop-down list, select a specific server from the list or select **Any server in the organization**.
4. Repeat steps 5 through 7 for all distribution groups or for dynamic distribution groups that are using the Bridgehead server as their expansion server.

The local computer is responsible for expanding group membership_ServerIsDynamicGroupExpansionServer

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because its attempt to uninstall a Hub Transport role responsible for expanding group membership failed.

Exchange 2007 setup requires that distribution list expansion be removed from the current Bridgehead server before the Hub Transport role can be uninstalled.

The expansion of distribution lists enables identification of individual recipients who belong to the distribution list to be identified, or the identification of additional distribution lists for expansion. An expanded distribution list can return the path for any required delivery status notification (DSN). DSNs notify the Microsoft Exchange administrator or e-mail sender of the status of a particular e-mail message. Additionally, distribution list expansion identifies whether Out of Office messages or automatically generated replies should be sent to the sender of the original message.

To resolve this issue, move distribution group expansion to another server and rerun Microsoft Exchange setup.

To change the expansion server for a distribution group or dynamic distribution group

1. Open the Exchange Management Console.
2. In the console tree, expand **Recipient Configuration**.
3. In the console tree, click **Distribution Group**.
4. Create a filter to view all distribution groups and dynamic distribution groups that use the current Bridgehead server as an expansion server.
   a. In the action pane, click **Create Filter**.
   b. In the property drop-down list, click **Expansion Server**.
   c. In the operator drop-down list, click **Equals**.
   d. At the value box, click the **Browse** button to select the Bridgehead server that currently acting as the expansion server.

Note:
The following step is optional.

1. Click **Add Expression** to specify additional filter criteria. Only messages that meet all filter criteria will be displayed.
2. Click **Apply Filter**. The results that meet the filter criteria are displayed.
1. In the results pane, click the distribution group you want to change the expansion server for, and then click **Properties** in the action pane.
2. On **Properties**, click the **Advanced** tab.
3. In the Expansion server drop-down list, select a specific server from the list or select **Any server in the organization**.
4. Repeat steps 5 through 7 for all distribution groups or for dynamic distribution groups that are using the Bridgehead server as their expansion server.

The fully-qualified domain name of the computer matches a recipient policy_ServerFQDNMatchesSMTPPolicy

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the

community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because the Fully Qualified Domain Name (FQDN) of the local computer matches the Simple Mail Transfer Protocol (SMTP) address of a recipient policy.

Microsoft Exchange setup requires that the FQDN of the servers in an Exchange organization not match any SMTP addresses of recipient policies in the same Exchange organization.

If the FQDN of a computer matches the SMTP address of a recipient policy, this match can cause mail to fail over SMTP and stall in the MTA queues.

To resolve this issue, rename the local computer or remove or rename the recipient policy and rerun Microsoft Exchange setup.

To rename the local computer

1. Open **System** in **Control Panel**.
2. On the **Computer Name** tab, click **Change**.
3. Under **Computer name**, type a new name for the computer, and then click **OK**. You will be prompted to provide a user name and user password to rename the computer in the domain.
4. Click **OK** to close the **System Properties** dialog box. You will be prompted to restart your computer to apply your changes.

Important:
If the computer that you want to rename is a domain controller, see "Rename a domain controller" (http://go.microsoft.com/fwlink/?LinkId=66828).

To modify the recipient policy SMTP address

1. Start Exchange System Manager.
2. Click **Organization**, click **Recipients**, and then click **Recipient Policies**.
3. Double-click the policy that you want to change.
4. Click the **E-Mail Addresses** tab, and then change the appropriate SMTP address

For more information about Recipient Policy naming issues, see Microsoft Knowledge Base article 288175, "XCON: Recipient Policy Cannot Match the FQDN of Any Server in the Organization, 5.4.8 NDRs" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=288175).

Older database files present_SecondSGFilesExist

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because it detected existing Microsoft Exchange database files in the target installation path.

Exchange 2007 setup requires that the target installation path be empty of Microsoft Exchange database files.

To resolve this issue, remove all existing files from target installation paths and then rerun setup.

To delete existing Exchange Server database files from the target installation path

1. In My Computer or Windows Explorer, locate the target install path.

Note:
By default, the database files are located in: <systemDrive>:\Program Files\Microsoft\Exchange Server\Mailbox\First Storage Group.

2. Right-click the files to be removed, and then select **Delete**.
3. At the **Confirm File Delete** dialog, click **Yes**.
4. Repeat steps 2 and 3 for all files in the target install paths.

The schema master is not running Windows Server 2003 Service Pack 1 or later_SchemaFSMONotWin2003SPn

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because the domain controller assigned the Active Directory directory service schema master role, also known as flexible single master operations (FSMO), is not running Microsoft Windows Server 2003 Service Pack 1 (SP1) or a later version.

Exchange 2007 setup requires that the domain controller that serves as the schema FSMO run Windows Server 2003 SP1 or a later version.

The FSMO controls all updates and modifications to the Active Directory schema.

To resolve this issue, do one or more of the following:

- Upgrade the FSMO domain controller to Windows Server 2003 SP1 or a later version and rerun Microsoft Exchange setup.
- If there is a FSMO domain controller running Microsoft Windows Server 2003 Service Pack 1 (SP1) or a later version in the Exchange organization, run Exchange 2007 setup with the /domaincontroller parameter pointing to that FSMO domain controller:

[*/DomainController*, or */dc <FQDN of domain controller>*]

Use the */DomainController* parameter to specify the domain controller to use to read from and write to Active Directory during setup. You can use NetBIOS or the fully qualified domain name (FQDN) format.

To obtain the latest service pack for Windows Server 2003, see the "Windows Server TechCenter" (http://go.microsoft.com/fwlink/?LinkId=45315).

For more information about Exchange Server 2007 Setup parameters, see "How to Install Exchange 2007 in Unattended Mode" (http://go.microsoft.com/fwlink/?LinkId=86476) in the Exchange Server 2007 product documentation.

Cannot find a recipient update service_RUSMissing

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 or Exchange Server 2010 setup cannot continue because the

Recipient Update Service (RUS) responsible for a domain in the existing Exchange organization cannot be found.

Microsoft Exchange setup requires that each domain in the existing Exchange organization have an instance of the Recipient Update Service.

If an instance of the Recipient Update Service is missing for a domain, new user objects created in the domain will not receive e-mail addresses issued to them.

To resolve this issue, verify that an instance of the Recipient Update Service exists for each domain and create an instance of the Recipient Update Service for the domains that do not have one and then rerun Microsoft Exchange setup.

To create a Recipient Update Service instance for a domain
1. Open Exchange System Manager. 2. Expand **Recipients**. 3. Right-click the **Recipient Update Services** node, click **New**, and then click **Recipient Update Service**. 4. In the New Object window, click **Browse** to locate the name of the domain. 5. Select the name of the domain and then click **OK**. 6. In the New Object window, click **Next**, and then **Finish**.

For more information about the Recipient Update Service, see the following Microsoft Knowledge Base articles:

- "How the Recipient Update Service applies recipient policies" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=328738).
- "How the Recipient Update Service Populates Address Lists" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=253828).
- "How to check the progress of the Exchange Recipient Update Service" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=246127).
- "Tasks performed by the Exchange Recipient Update Service" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=253770).

Active Directory domain is mixed mode_RootDomainModeMixed

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because an existing Active Directory domain is not set to Microsoft Windows®°2000 Server native mode or better.

Exchange 2007 setup will create Universal Security Groups that can only exist in Windows 2000 Server native mode, or better, domains.

To resolve this issue, follow these steps to raise the domain functional level to at least the Windows 2000 Server native level, and then rerun Exchange 2007 setup.

To raise the domain functional level

1. Open Active Directory Domains and Trusts.
2. In the console tree, right-click the domain for which you want to raise functionality, and then click **Raise Domain Functional Level**.
3. In **Select an available domain functional level**, use one of the following procedures:
   - To raise the domain functional level to Windows 2000 Server native, click **Windows 2000 native**, and then click **Raise**.
   - To raise domain functional level to Windows Server® 2003, click **Windows Server 2003**, and then click **Raise.**

Caution:
If you have or will have any domain controllers running Windows NT® 4.0 and earlier, do not raise the domain functional level to Windows 2000 Server native. After the domain functional level is set to Windows 2000 Server native, it cannot be changed back to Windows 2000 Server mixed. If you have or will have any domain controllers running Windows NT 4.0 and earlier or Windows 2000 Server, do not raise the domain functional level to Windows Server 2003. After the domain functional level is set to Windows Server 2003, it cannot be changed back to Windows 2000 Server mixed or Windows 2000 Server native.

The primary DNS server cannot be contacted_PrimaryDNSTestFailed

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because communication with the primary Domain Name System (DNS) server cannot be established.

Exchange 2007 setup requires that the local computer communicate with the authoritative DNS database for the domain.

Microsoft Exchange depends on DNS to resolve the IP Address of its next internal or external destination server.

Communication with the primary DNS server can fail for the following reasons:

- The local TCP/IP configuration does not point to the correct DNS server.
- The DNS server is down or unreachable because of a network failure or other reasons.

To resolve this issue:

- Verify that the local TCP/IP configuration points to the correct DNS server.

To verify the local TCP/IP configuration

1. Review the local TCP/IP configuration:

For more information, see "Configure TCP/IP to use DNS" (http://go.microsoft.com/fwlink/?LinkId=68094).

- Verify that the DNS server is running and can be contacted.

To verify that the DNS server is running and can be contacted

1. Verify that the DNS server is running by doing one or more of the following checks:
   - Look at the DNS server status from the DNS Administration program on the DNS server.
   - Restart the DNS server.

For more information, see "Start, stop, pause, or restart a DNS server" (http://go.microsoft.com/fwlink/?LinkId=62999).

   - Verify the DNS server responsiveness by using the **nslookup** command.

For more information, see the instructions in "Verify DNS server responsiveness using the nslookup command" (http://go.microsoft.com/fwlink/?LinkId=63000).

Insufficient permissions to run / PrepareDomain_PrepareDomainNotAdmin

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because the attempt to run the **/PrepareDomain** process failed. The logged on user has insufficient permissions to perform the **/PrepareDomain** process.

Exchange 2007 setup requires that the user who is logged on when running the **/PrepareDomain** process be a member of the Domain Admins group for the domain to be prepared, and a member of the Enterprise Admins group.

To resolve this issue, grant the logged-on user Domain Admins group permissions for the domain being prepared and enroll them in the Enterprise Admins groups, or log on with an account that has those permissions and rerun Exchange 2007 setup.

For more information about Active Directory permissions that are needed with Microsoft Exchange, see "Working with Active Directory Permissions in Exchange Server" (http://go.microsoft.com/fwlink/?LinkId=47592).

Active Directory domain is mixed mode_PrepareDomainModeMixed

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because an existing Active Directory

domain is not set to Microsoft Windows®°2000 Server native mode or better.

Exchange 2007 setup will create Universal Security Groups that can only exist in Windows 2000 Server native mode, or better, domains.

To resolve this issue, follow these steps to raise the domain functional level to at least the Windows 2000 Server native level, and then rerun Exchange 2007 setup.

To raise the domain functional level

1. Open Active Directory Domains and Trusts.
2. In the console tree, right-click the domain for which you want to raise functionality, and then click **Raise Domain Functional Level**.
3. In **Select an available domain functional level**, use one of the following procedures:
   - To raise the domain functional level to Windows 2000 Server native, click **Windows 2000 native**, and then click **Raise**.
   - To raise domain functional level to Windows Server® 2003, click **Windows Server 2003**, and then click **Raise.**

Caution:
If you have or will have any domain controllers running Windows NT® 4.0 and earlier, do not raise the domain functional level to Windows 2000 Server native. After the domain functional level is set to Windows 2000 Server native, it cannot be changed back to Windows 2000 Server mixed. If you have or will have any domain controllers running Windows NT 4.0 and earlier or Windows 2000 Server, do not raise the domain functional level to Windows Server 2003. After the domain functional level is set to Windows Server 2003, it cannot be changed back to Windows 2000 Server mixed or Windows 2000 Server native.

The operating system is in debug mode_OSCheckedBuild

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

The Microsoft® Exchange Server Analyzer Tool queries the **Win32_OperatingSystem** Microsoft

Windows® Management Instrumentation (WMI) class to determine whether a value is set for the **Debug** property. If the value for this key on an Exchange Server computer is set to **True**, an error is displayed.

Windows debug mode is toggled by adding the **/debug** parameter in the Boot.ini file. When **/debug** is specified in the Boot.ini file of a Windows Server computer, the kernel debugger is loaded during startup and kept in memory at all times. This enables a support professional to dial in to the system being debugged and break in to the debugger, even when the system is not suspended at a Kernel STOP screen. Unlike the **/crashdebug** switch, the **/debug** switch uses the COM port whether you are debugging or not. This switch is used when debugging problems that are regularly reproducible. It is likely that the **/debug** parameter was set as a means to troubleshoot a problem, and was unintentionally left set.

Because of the additional processes that are being run, performance suffers greatly. Therefore, running Exchange Server on a computer where Windows is running in debug mode is not recommended.

To eliminate this error, edit the Boot.ini file and remove the **/debug** parameter.

⊟ To correct this error

1. In Windows Explorer, navigate to the System Partition. This is the partition that holds the hardware specific files such as Boot.ini and NTLDR.
2. If you cannot see the Boot.ini file, it could be because the **Folder Options** are set to **Hide protected operating system files**. If this is the case, in the Windows Explorer window, click **Tools**, click **Folder Options**, and then click **View**. Clear the **Hide protected operating system files (Recommended)** check box. When prompted, click **Yes**.
3. When the Boot.ini file is visible in Windows Explorer, right-click the file, click **Open With**, and then select **Notepad** to open the file.
4. In the **[Operating Systems]** section, remove the **/debug** parameter.
5. Save and close the file, and then restart the Exchange Server computer for the change to take effect.

For more information about the parameters that can be used in the Boot.ini file, see the Microsoft Knowledge Base article 833721, "Available switch options for the Windows XP and Windows Server 2003 Boot.ini files" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=833721).

The logged-on user is not a member of the local Administrators group_NotLocalAdmin

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because the logged-on user is not a member of the local computer's administrators group.

Exchange 2007 setup requires that the logged-on user who installs Microsoft Exchange has full access to the local computer.

To resolve this issue, log on to the computer by using an account that has local administrator access and then rerun Microsoft Exchange setup.

Not in schema master site/ domain_NotInSchemaMasterSite

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because the computer that is running setup is not in the same Active Directory® directory service site or domain as the server that is assigned the domain schema master role, also known as flexible single master operations or FSMO.

Exchange 2007 setup requires the domain controller that serves as the domain schema master to be in the same site and domain as the local computer that is running Exchange setup.

The domain schema master controls all updates and modifications to the Active Directory schema.

To resolve this issue, run Exchange Server 2007 setup using the **/prepareschema** and **/prepareAD** switches from the same site and domain as the domain schema master.

For more information about the **/prepareschema** and **/prepareAD** setup switches, see the Exchange 2007 product documentation topic "How to Prepare Active Directory and Domains" (http://go.microsoft.com/fwlink/?linkid=78453)

You can use the Schema Master tool to identify the role. However, the Schmmgmt.dll DLL must be registered in order to make the Schema tool available as an MMC snap-in.

To view the current schema master

1. At a command prompt, type **regsvr32 schmmgmt.dll**

Note:

RegSvr32 has been successfully registered when the following dialog box is displayed: DllRegisterServer in schmmgmt.dll succeeded.

2. To open a new management console, click **Start**, click **Run**, and then type **mmc**.
3. On the Console menu, click **Add/Remove Snap-in**.
4. Click **Add** to open the **Add Standalone Snap-in** dialog box.
5. Select **Active Directory Schema**, and then click **Add**.
6. "Active Directory Schema" is displayed in the Add/Remove snap-in. Click **Close**, and then click **OK** to return to the console.
7. Select **Active Directory Schema** so that the **Classes** and **Attributes** sections are displayed on the right side.
8. Right-click **Active Directory Schema** and then click **Operations Master**.
9. The current schema master is displayed

After you identify the current schema master, determine which subnet the schema master is located in. Then, use one of the following methods to install Exchange:

- Modify the subnet on the Exchange server to move it into the site in which the schema master is located. Then, install Exchange.
- Temporarily force a site membership change on the Exchange server, and then install Exchange. After Exchange is installed, return the Exchange server to its original site.

To force site membership

1. On the server on which you want to install Exchange, start Registry Editor. To do this, click **Start**, click **Run**, type **regedit**, and then click **OK**.
2. Locate the following registry subkey:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters

3. Create the following new **String** value:

Value name: **SiteName**

Value type: **REG_SZ**

Value data: **<site_that_contains_the_schema_master>**

4. Exit Registry Editor, and then restart the Netlogon service. This action forces the Exchange server to participate in the site that you specified.
5. Install Exchange.
6. Remove the registry entry that you added in step 3.

7. Restart the Netlogon service. This action returns Exchange to the original site.

Not in schema master site/ domain_NotInSchemaMasterDomain

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because the computer that is running setup is not in the same Active Directory® directory service site or domain as the server that is assigned the domain schema master role, also known as flexible single master operations or FSMO.

Exchange 2007 setup requires the domain controller that serves as the domain schema master to be in the same site and domain as the local computer that is running Exchange setup.

The domain schema master controls all updates and modifications to the Active Directory schema.

To resolve this issue, run Exchange Server 2007 setup using the **/prepareschema** and **/prepareAD** switches from the same site and domain as the domain schema master.

For more information about the **/prepareschema** and **/prepareAD** setup switches, see the Exchange 2007 product documentation topic "How to Prepare Active Directory and Domains" (http://go.microsoft.com/fwlink/?linkid=78453)

You can use the Schema Master tool to identify the role. However, the Schmmgmt.dll DLL must be registered in order to make the Schema tool available as an MMC snap-in.

To view the current schema master

1. At a command prompt, type **regsvr32 schmmgmt.dll**

Note:
RegSvr32 has been successfully registered when the following dialog box is displayed: DllRegisterServer in schmmgmt.dll succeeded.

2. To open a new management console, click **Start**, click **Run**, and then type **mmc**.

3. On the Console menu, click **Add/Remove Snap-in**.

4. Click **Add** to open the **Add Standalone Snap-in** dialog box.

5. Select **Active Directory Schema**, and then click **Add**.

6. "Active Directory Schema" is displayed in the Add/Remove snap-in. Click **Close**, and then click **OK** to return to the console.
7. Select **Active Directory Schema** so that the **Classes** and **Attributes** sections are displayed on the right side.
8. Right-click **Active Directory Schema** and then click **Operations Master**.
9. The current schema master is displayed

After you identify the current schema master, determine which subnet the schema master is located in. Then, use one of the following methods to install Exchange:

- Modify the subnet on the Exchange server to move it into the site in which the schema master is located. Then, install Exchange.
- Temporarily force a site membership change on the Exchange server, and then install Exchange. After Exchange is installed, return the Exchange server to its original site.

To force site membership

1. On the server on which you want to install Exchange, start Registry Editor. To do this, click **Start**, click **Run**, type **regedit**, and then click **OK**.
2. Locate the following registry subkey:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters

3. Create the following new **String** value:

Value name: **SiteName**

Value type: **REG_SZ**

Value data: **<site_that_contains_the_schema_master>**

4. Exit Registry Editor, and then restart the Netlogon service. This action forces the Exchange server to participate in the site that you specified.
5. Install Exchange.
6. Remove the registry entry that you added in step 3.
7. Restart the Netlogon service. This action returns Exchange to the original site.

Cannot install Exchange 2007 roles after you prepare Active Directory for Exchange 2010_NoE12ServerWarning

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the

community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

When you run Microsoft Exchange Server 2010 **Setup /PrepareAD**, the Microsoft Exchange Server Analyzer Tool queries the existing Active Directory topology to determine whether any Microsoft Exchange Server 2007 server roles exist. If Exchange 2007 server roles are not detected, you receive the following warning message:

> Setup is going to prepare the organization for Exchange Server 2010 via 'Setup /PrepareAD' and no Exchange Server 2007 roles have been detected in this topology. After this operation, it will be impossible to install any Exchange Server 2007 roles.

Before deploying Exchange Server 2010, consider the following factors that may require you to deploy an Exchange 2007 server with all server roles installed prior to deploying Exchange 2007:

- **Third-party or in-house developed applications** Applications developed for Exchange 2003 may not be compatible with Exchange 2010, and thus need to be upgraded or replaced. You can maintain these applications and the associated user population on Exchange 2003; move to Exchange 2007; or replace the software with a compatible version for Exchange 2010.
- **Coexistence or migration requirements** If you plan on migrating mailboxes into your organization, you can either deploy Exchange 2007 and use the Microsoft Transporter Suite, or you can use a third-party coexistence or migration solution. To download the Microsoft Transporter Suite, go to Microsoft Transporter Suite at the Microsoft Download Center.

In addition, when evaluating the options for your organization, make sure you have considered the following questions:

- Do you have a strategy in place to move the dependent applications to Exchange 2010 before Exchange 2003 reaches end of support? For more information, visit the Microsoft Support Lifecycle Web page (http://go.microsoft.com/fwlink/?LinkID=55839).
- Does your strategy require WebDAV and Web Services coexistence (Exchange 2007)?
- Have you considered third-party products that support Exchange or other Microsoft technologies that will allow you to meet your coexistence or migration requirements?
- What is your hardware lifecycle approach (continue to use as many existing 32-bit servers as possible versus buying new 64-bit servers)?
- What is your plan for migration (migrate all servers as fast as possible versus migrating in a phased strategy)? Similarly, what is your timeline for the coexistence of different versions of Exchange?

If you decide that you need to deploy an Exchange 2007 server prior to deploying Exchange 2010, the deployment of a single Exchange 2007 with all server roles is sufficient to enable the deployment of future Exchange 2007 servers in the organization. To deploy the Exchange 2007 server into your Exchange 2003 organization, follow these steps:

1. Run Exchange 2007 **Setup /PrepareSchema**.
2. Run Exchange 2007 **Setup /PrepareAD**.

3. Run Exchange 2007 **Setup /PrepareDomain** on all domains that contain recipients, Exchange 2003 servers, or global catalogs that could be used by an Exchange server.
4. Install an Exchange 2007 server with all four server roles (Hub Transport, Client Access, Mailbox, and Unified Messaging).

The Distributed Transaction Coordinator Service must be started before setup can continue_MSDTCStopped

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because its attempt to install the Client Access Server or Unified Messaging server roles failed because the Distributed Transaction Coordinator service is not started on the target computer.

Exchange 2007 setup requires the computer that you are installing Microsoft Exchange to have the Distributed Transaction Coordinator service status set to **Started**.

The Distributed Transaction Coordinator service provides services designed to ensure successful and complete transactions, even with system failures, process failures, and communication failures.

Each computer participating in a distributed transaction manages its own resources and data and also acts in concert with other computers in the transaction. Above all, a distributed transaction must commit or abort its work entirely on all participating computers. The Distributed Transaction Coordinator performs the transaction coordination role for the components involved and acts as a transaction manager for each computer that manages transactions.

Both the Client Access Server and Unified Messaging server roles have dependencies on the Distributed Transaction Coordinator service.

To resolve this issue, verify that the Distributed Transaction Coordinator service status is set to

Started on the local computer, and then rerun Microsoft Exchange setup.

To set the status of the Distributed Transaction Coordinator service to 'Started'

1. Right-click **My Computer**, and then click **Manage**.
2. Expand the **Services and Applications** node, and then click the **Services** node.
3. In the right pane, locate the **Distributed Transaction Coordinator**.
4. Right-click **Distributed Transaction Coordinator**, and then click **Properties**.
5. Set the **Startup Type** to **Automatic** and the **Service status** to **Started**.
6. Click **Apply**, and then click **OK**.

Messages currently exist in one or more queues_MessagesInQueue

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup displays this warning because its attempt to uninstall a transport role could cause data loss from a transport queue.

Exchange 2007 setup checks to make sure that the transport queues are empty before it removes the roles associated with managing those queues.

If you remove the transport roles before delivery of messages still in the transport queues, those messages may be held indefinitely.

To resolve this issue, inspect the referenced queues to make sure that they are empty of messages before you continue with setup.

To view the contents of a queue

1. Open the Exchange Management Console.
2. In the console tree, click **Toolbox**.
3. In the result pane, click **Exchange Queue Viewer**.
4. In the action pane, click **Open Tool**.
5. In the Queue Viewer, click the **Queues** tab. A list of all queues on the server to which you are connected is displayed.

6. Right-click the queue you want and select **Properties** to view the properties of the queue.

To view messages in a queue

1. Follow steps 1 through 4.
2. In the Queue Viewer, click the **Messages** tab. A list of all messages on the server to which you are connected is displayed. To adjust the view to a single queue, click the **Queues** tab, double-click the queue name, and then click the Server\Queue tab that appears.
3. To view detailed information about a message, select a message, and then click **Properties** in the action pane.

Storage group drive specification is missing_MailboxLogDriveDoesNotExist

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 Disaster Recovery setup cannot continue because Disaster Recovery setup cannot access the storage group database drive specification that was used in the previous installation of this server.

Microsoft Exchange Disaster Recovery setup requires that the same storage group database drive specifications previously used for this server be available during the restore.

To resolve this situation, configure the drives to match the original logical drive configuration and rerun Microsoft Exchange Disaster Recovery setup.

Assign or change a drive letter

1. Open **Computer Management (Local)**.
2. In the console tree, click **Computer Management (Local)**, click **Storage**, and then click **Disk Management**.
3. Right-click a partition, logical drive, or volume, and then click **Change Drive Letter and Paths**.
4. Use one of the following procedures:
   - To assign a drive letter, click **Add**, click the drive letter you want to use, and then click **OK**.
   - To modify a drive letter, click it, click **Change**, click the drive letter you want to use, and then click **OK**.

For more information about how to assign drive letters, see "Assign, change, or remove a drive letter" (http://go.microsoft.com/fwlink/?LinkId=66764).

Mailbox database drive specification is missing_MailboxEDBDriveDoesNotExist

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 Disaster Recovery setup cannot continue because Disaster Recovery setup cannot access the mailbox database drive specification that was used in the previous installation of this server.

Microsoft Exchange Disaster Recovery setup requires that the same mailbox database drive specifications previously used for this server be available during the restore.

To resolve this situation, configure the drives to match the original logical drive configuration and rerun Microsoft Exchange Disaster Recovery setup.

Assign or change a drive letter

1. Open **Computer Management (Local)**.
2. In the console tree, click **Computer Management (Local)**, click **Storage**, and then click **Disk Management**.
3. Right-click a partition, logical drive, or volume, and then click **Change Drive Letter and Paths**.
4. Use one of the following procedures:
   - To assign a drive letter, click **Add**, click the drive letter you want to use, and then click **OK**.
   - To modify a drive letter, click it, click **Change**, click the drive letter you want to use, and then click **OK**.

For more information about how to assign drive letters, see "Assign, change, or remove a drive letter" (http://go.microsoft.com/fwlink/?LinkId=66764).

The Windows Process Activation Service - Process Model component is required_LonghornWASProcessModelInstalled

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Exchange Server 2010 setup cannot continue installation on the Windows Server 2008-or Windows Server 2008 R2-based computer because the Windows Process Activation Service - Process Model feature is not installed on the server.

The Windows Process Activation Service generalizes the Internet Information Services (IIS) process model, removing the dependency on HTTP. All the features of IIS that were previously available only to HTTP applications are now available to applications hosting Windows Communication Foundation (WCF) services, by using non-HTTP protocols. IIS 7.0 also uses Windows Process Activation Service for message-based activation over HTTP.

The Process Model hosts Web and WCF services. Introduced in IIS 6.0, the Process Model is a new architecture that features rapid failure protection, health monitoring, and recycling.

To resolve this issue, install the Windows Process Activation Service - Process Model feature on this server and then rerun Exchange 2010 setup.

Install the Windows Process Activation Service - Process Model feature by using the Server Manager tool

1. Click **Start**, click **Administrative Tools** and then **Server Manager**.
2. In the left navigation pane, right-click **Features**, and then click **Add Features**.
3. On the **Select Features** pane, scroll down to **Windows Process Activation Service**.
4. Select the check boxes for **Process Model**.
5. Click **Next** from the **Select Features** pane, and then click **Install** at the **Confirm Installations Selections** pane.

6. Click **Close** to leave the Add Role Services wizard.

IIS 7 component not installed_LonghornIIS7WindowsAuthNotInstalled

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2010 Setup or Microsoft Exchange Server 2007 Setup cannot install the Client Access Server (CAS) role or the Mailbox server role on a Microsoft Windows Server 2008-based computer or on a Windows Server 2008 R2-based computer because the required Internet Information Server (IIS) 7 components are not installed.

Exchange 2010 Setup and Exchange 2007 Setup require that the Windows Server 2008-based computer or the Windows Server 2008 R2-based computer on which you are installing the CAS role already has the following IIS 7 components installed.

Required IIS 7 Components for the CAS server role
Dynamic Content Compression
Static Content Compression
Basic Authentication
Windows Authentication
IIS 7 Digest Authentication
ASP.NET
Client Certificate Mapping

Directory Browsing
HTTP Errors
HTTP Logging
HTTP Redirection
Tracing
ISAPI Filters
Request Monitor
Static Content

Exchange 2010 Setup and Exchange 2007 Setup require that the Windows Server 2008-based computer or the Windows Server 2008 R2-based computer on which you are installing the Mailbox server role already has the following IIS 7 components installed.

Required IIS 7 Components for the Mailbox server role
Basic Authentication
Windows Authentication

To address this issue, follow the appropriate steps to install the required IIS 7 components on the destination computer, and then run Microsoft Exchange Setup again.

Install the IIS 7 Components for the CAS server role by using the Windows Server 2008 Server Manager

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS),** and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS**.
4. In the **Security** area, click to select the following check boxes:
   - **Basic Authentication**
   - **Digest Authentication**
   - **Windows Authentication**
5. In the **Performance** area, click to select the following check boxes:
   - **Static Compression**
   - **Dynamic Compression**
6. In the **Select Role Services** pane, click **Next**, and then click **Install** in the **Confirm Installations Selections** pane.

7. Click **Close** to exit the Add Role Services wizard.

Install the IIS 7 Components for the Mailbox server role by using the Windows Server 2008 Server Manager

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS),** and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS**.
4. In the **Security** area, click to select the following check boxes:
   - **Basic Authentication**
   - **Windows Authentication**
5. In the **Select Role Services** pane, click **Next**, and then click **Install** in the **Confirm Installations Selections** pane.
6. Click **Close** to exit the Add Role Services wizard.

IIS 7 .NET Extensibility component is required_LonghornIIS7NetExt

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2010 Setup cannot continue because the required Internet Information Server (IIS) 7 .NET Extensibility component is not installed on the target server.

Before Exchange 2010 can use Windows PowerShell Remoting, the IIS 7 .NET Extensibility component must be installed on the target server.

To address this error, use Server Manager to install the IIS 7 .NET Extensibility component on this server and then rerun Exchange 2010 setup.

Install the IIS 7 .NET Extensibility component in Windows Server 2008 or in Windows Server 2008 R2 by using the Server Manager tool

1. Click **Start**, click **Administrative Tools** and then **Server Manager**.
2. In the left navigation pane, expand **Roles**, and then right-click **Web Server (IIS)** and select **Add**

Role Services.

3. On the **Select Role Services** pane, scroll down to **Application Development**.
4. Select the check box under **.NET Extensibility**.
5. Click **Next** from the **Select Role Services** pane, and then click **Install** at the **Confirm Installations Selections** pane.
6. Click **Close** to exit the Add Role Services wizard.

Uninstall Unified Messaging Language Packs_AdditionalUMLangPackExists

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because the Unified Messaging server role upgrade failed.

Exchange setup requires that all Unified Messaging server role language packs, except the US English language pack, be uninstalled before the Unified Messaging server role upgrade can continue.

Unified Messaging (UM) language packs that are included with Exchange 2007 contain pre-recorded prompts, Text-to-Speech (TTS) conversion support for a given language.

Exchange 2007 UM language packs enable callers and Outlook Voice Access users to interact with the Unified Messaging system in multiple languages.

The existing non-US English language packs need to be uninstalled so that new language packs can be installed.

To resolve this issue, uninstall all Unified Messaging server role language packs, except the US English language pack, and then rerun Exchange 2007 setup.

For more information about uninstalling Unified Messaging server role language packs, see How to Remove a Unified Messaging Language Pack from a Unified Messaging Server in the Exchange 2007 product documentation.

Insufficient permissions to prepare Active Directory_ADUpdateRequired

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because the attempted domain preparation failed.

Exchange setup requires that the Active Directory directory service be modified for Exchange Server 2007 before domains in the Active Directory can be prepared.

The account being used to run the **setup /PrepareAD** command has insufficient permissions to execute this command even though it appears to belong to the Enterprise Admins group. The account may have expired.

To resolve this issue, verify that the logged-on user account is valid and belongs to the Enterprise Admins group, or log on with an account that has those permissions and rerun **setup /PrepareAD**.

For more information about how to perform the PrepareAD process, see "How to Prepare Active Directory and Domains" (http://go.microsoft.com/fwlink/?LinkId=78453).

For more information about Active Directory permissions that are needed with Microsoft Exchange, see "Working with Active Directory Permissions in Exchange Server" (http://go.microsoft.com/fwlink/?LinkId=47592).

Hub Transport role not detected in local site_BridgeheadRoleNotPresentInSite

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup displays this warning because an existing Hub Transport server role could not be detected in the local Active Directory® directory service site.

You have chosen to install the Mailbox Server role before an instance of the Hub Transport role is installed in the Active Directory site.

Exchange 2007 Hub Transport Services are deployed inside your organization's Active Directory. Hub Transport Services handle all mail flow inside the organization, applies organizational mail flow routing rules, and are responsible for delivering messages to a recipient's mailbox.

Users will be able to log on to their mailboxes, but mail cannot be sent or received from this mailbox server until a Hub Transport role is installed.

Insufficient permissions to remove all server roles_CannotUninstallDelegatedServer

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because the attempt to uninstall all server roles failed.

Exchange 2007 setup requires that the user who is logged on when uninstalling all server roles be a member of the Exchange Organization Administrators groups, or the Enterprise Admins group.

To resolve this issue, grant the logged-on user Exchange Organization Administrator permissions or enroll them in the Enterprise Admins group, or log on with an account that has those permissions and rerun Exchange 2007 setup.

For more information about Active Directory permissions needed with Microsoft Exchange, see "Working with Active Directory Permissions in Exchange Server" (http://go.microsoft.com/fwlink/?LinkId=47592).

Client Access role not detected in local site_ClientAccessRoleNotPresentInSite

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup displayed this warning because an existing Client Access role could not be detected in the local Active Directory® directory service site.

You have chosen to install the Mailbox Server role before an instance of the Client Access role is installed in the Active Directory site.

The Client Access server role accepts connections to your Exchange 2007 server from a variety of different clients. Software clients such as Microsoft Outlook Express and Eudora use POP3 or IMAP4 connections to communicate with the Exchange server. Hardware clients, such as mobile devices, use ActiveSync, POP3, or IMAP4 to communicate with the Exchange server.

If users access their Inbox by using any client other than Microsoft Office Outlook®, you must install the Client Access server role in your Exchange organization.

Users will be able to logon to their mailboxes with Outlook but will not be able to use Outlook Web Access or mobile devices against the same mailbox until a Client Access role is present in the local Active Directory site.

Only the Mailbox role can be installed on a cluster server_ClusSvcInstalledRoleBlock

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because its attempt to install a role other than the Mailbox Server role on this clustered server failed.

Microsoft Exchange does not support Exchange 2007 roles other than the Mailbox Server role on servers that have the Cluster service installed.

To address this issue, see "High Availability for Exchange 2007" in the Exchange Server 2007 Operations Guide (http://go.microsoft.com/fwlink/?LinkId=68190).

Setup cannot install Exchange to a read-only domain controller_ComputerRODC

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 Setup cannot continue the installation because the target computer is a Read-Only Domain Controller (RODC).

Read-Only Domain Controllers are a feature of Windows Server 2008 Active Directory. The RODC is a type of domain controller install option that can be installed in remote sites that may have lower levels of physical security.

Exchange Setup requires write access to the target install computer.

To resolve this issue, select a target install computer that is not designated as an RODC, and rerun Setup.

Domain Controller Override is set in the Registry_ConfigDCHostNameMismatch

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because its attempt to use the specified domain controller failed. A domain controller has been statically mapped in the registry.

Exchange 2007 setup requires that the domain controller specified in the setup command match the domain controller that has been statically mapped using a registry override.

To resolve this issue, run setup again, specifying the statically mapped domain controller for the **/DomainController:** <*FQDN of the statically mapped domain controller*> parameter.

For more information about DSAccess and directory services detection, see Microsoft Knowledge Base article 250570, "Directory Service Server Detection and DSAccess Usage" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=250570).

Insufficient permissions to prepare Active Directory_DomainPrepWithoutADUpdate

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because the attempted domain preparation failed.

Exchange setup requires that the Active Directory directory service be modified for Exchange Server 2007 before domains in the Active Directory can be prepared.

The account being used to run the **setup /PrepareAD** command has insufficient permissions to execute this command even though it appears to belong to the Enterprise Admins group. The account may have expired.

To resolve this issue, verify that the logged-on user account is valid and belongs to the Enterprise Admins group, or log on with an account that has those permissions and rerun **setup /PrepareAD**.

For more information about how to perform the PrepareAD process, see "How to Prepare Active Directory and Domains" (http://go.microsoft.com/fwlink/?LinkId=78453).

For more information about Active Directory permissions that are needed with Microsoft Exchange, see "Working with Active Directory Permissions in Exchange Server" (http://go.microsoft.com/fwlink/?LinkId=47592).

The local computer is already running

Exchange Server_ExchangeAlreadyInstalled

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because the local computer has previous Microsoft Exchange components installed.

Exchange 2007 setup requires that the local computer not have existing Microsoft Exchange components installed.

To resolve this issue, remove any Microsoft Exchange 2000 Server or Microsoft Exchange Server 2003 components, and then rerun Microsoft Exchange setup.

To remove Microsoft Exchange components

1. Click **Start**, point to **Settings**, and then click **Control Panel**.
2. Double-click **Add or Remove Programs**.
3. In the **Currently installed programs** list, click **Microsoft Exchange**, and then click **Change/ Remove**.
4. In Microsoft Exchange Installation Wizard, click **Next**.
5. In the Action list on the Component Selection page, click the down arrow next to each component that has been installed, and then click **Remove**.

Note:
Installed components have a check mark in the Action list. When you click **Remove**, the check mark is replaced by the word **Remove**.

6. Click **Next** two times.
7. Click **Finish**.

Older database files present_FirstSGFilesExist

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because it detected existing Microsoft Exchange database files in the target installation path.

Exchange 2007 setup requires that the target installation path be empty of Microsoft Exchange database files.

To resolve this issue, remove all existing files from target installation paths and then rerun setup.

To delete existing Exchange Server database files from the target installation path

1. In My Computer or Windows Explorer, locate the target install path.

Note:
By default, the database files are located in: <systemDrive>:\Program Files\Microsoft\Exchange Server\Mailbox\First Storage Group.

2. Right-click the files to be removed, and then select **Delete**.
3. At the **Confirm File Delete** dialog, click **Yes**.
4. Repeat steps 2 and 3 for all files in the target install paths.

One or more Active Directory Connector servers were found_ADCFound

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 and Exchange Server 2010 setup cannot continue because one or more Active Directory Connectors (ADC) have been found in the current Microsoft Exchange environment.

ADC replicates objects from Exchange Server version 5.5 to the Active Directory directory service in a mixed mode Microsoft Exchange environment and is not supported by Exchange 2007 or Exchange 2010.

Exchange 2007 or Exchange 2010 setup requires that all ADC components be removed.

To resolve this issue, remove all ADC components, and rerun Exchange 2007 or Exchange 2010 setup.

To remove Active Directory Connector components
1. To disable the ADC service on the server that is running the ADC service, right-click **My Computer** on the desktop, and then click **Manage**. 2. Expand the **Services and Applications** node, and then click the **Services** node. 3. In the right pane, right-click **Microsoft Active Directory Connector** and then click **Properties**. 4. Change the **Startup Type** to **Disabled**. The next time that the computer starts, the ADC service will not start. 5. Click **Apply**, and then click **OK**. 6. To uninstall the ADC service, use the Active Directory Installation Wizard on the Microsoft Exchange 2000 Server or Microsoft Exchange Server 2003 CD. Open the \ADC\I386 folder and double-click the Setup.exe program. Follow the prompts to **Remove All** ADC service components.
Important:
You must complete step 6 and **Remove All** ADC components to resolve this issue. It is insufficient to disable the ADC service.

For more information about ADC, see the following Microsoft Knowledge Base articles:

- 325300, "Support WebCast: Introduction to the Active Directory Connector" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=325300).
- 325221, "Support WebCast: Microsoft Advanced Active Directory Connector" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=325221).
- 312632, "How To Install and Configure the Active Directory Connector in Exchange 2000 Server" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=312632).

The schema master is not running Windows Server 2003 Service Pack 1 or later_DomainControllerIsOutOfSite

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because the domain controller assigned the Active Directory directory service schema master role, also known as flexible single master operations (FSMO), is not running Microsoft Windows Server 2003 Service Pack 1 (SP1) or a later version.

Exchange 2007 setup requires that the domain controller that serves as the schema FSMO run Windows Server 2003 SP1 or a later version.

The FSMO controls all updates and modifications to the Active Directory schema.

To resolve this issue, do one or more of the following:

- Upgrade the FSMO domain controller to Windows Server 2003 SP1 or a later version and rerun Microsoft Exchange setup.
- If there is a FSMO domain controller running Microsoft Windows Server 2003 Service Pack 1 (SP1) or a later version in the Exchange organization, run Exchange 2007 setup with the /domaincontroller parameter pointing to that FSMO domain controller:

[*/DomainController*, or */dc <FQDN of domain controller>*]

Use the */DomainController* parameter to specify the domain controller to use to read from and write to Active Directory during setup. You can use NetBIOS or the fully qualified domain name (FQDN) format.

To obtain the latest service pack for Windows Server 2003, see the "Windows Server TechCenter" (http://go.microsoft.com/fwlink/?LinkId=45315).

For more information about Exchange Server 2007 Setup parameters, see "How to Install Exchange 2007 in Unattended Mode" (http://go.microsoft.com/fwlink/?LinkId=86476) in the Exchange Server 2007 product documentation.

Domain preparation required_DomainPrepRequired

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because the attempt to install the server role failed.

Microsoft Exchange setup requires that the local domain be prepared for Exchange Server 2007 before certain server roles can be installed.

Preparation of the domain for Exchange Server 2007 consists of the following tasks:

- Setting permissions on the Domain container for the Exchange Servers, Exchange Organization Administrators, Authenticated Users, and Exchange Mailbox Administrators.
- Creating the Microsoft Exchange System Objects container if it does not exist, and setting permissions on this container for the Exchange Servers, Exchange Organization Administrators, and Authenticated Users.
- Creating a new domain global group in the current domain called Exchange Install Domain Servers.
- Adding the Exchange Install Domain Servers group to the Exchange Servers USG in the root domain.

To resolve this issue, run **setup /PrepareDomain** to prepare the local domain and retry the server role installation.

For more information about how to perform the PrepareDomain process, see "How to Prepare Active Directory and Domains" (http://go.microsoft.com/fwlink/?LinkId=78453).

The COM+ Event System Service must be started before setup can

continue_EventSystemStopped

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because its attempt to install the Client Access Server or Edge Transport server roles failed because the COM+ Event System service is not started on the target computer.

Exchange 2007 setup requires the computer that you are installing Microsoft Exchange to have the COM+ Event System service status set to **Started**.

The COM+ Event System service supports system event notification for COM+ components, which provide automatic distribution of events to subscribing COM components.

Both the Client Access Server and Edge Transport server roles have dependencies on COM+ components that subscribe to the COM+ Event System service.

To resolve this issue, verify that the COM+ Event System service status is set to **Started** on the local computer, and then rerun Microsoft Exchange setup.

To set the status of the COM+ Event System service to 'Started'

1. Right-click **My Computer**, and then click **Manage**.
2. Expand the **Services and Applications** node, and then click the **Services** node.
3. In the right pane, locate the **Com+ Event System**.
4. Right-click **Com+ Event System**, and then click **Properties**.
5. Set the **Startup Type** to **Automatic** and the **Service status** to **Started**.
6. Click **Apply**, and then click **OK**.

IIS is in 32-bit compatibility mode_IIS32BitMode

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because the Microsoft Internet Information Service (IIS) is running in 32-bit mode on this 64-bit computer.

Exchange 2007 requires that IIS be in full 64-bit mode when it runs on a 64-bit computer.

To resolve this issue, switch IIS to 64-bit mode, and then rerun Microsoft Exchange setup.

To switch IIS to 64-bit mode

1. Open a command prompt.
2. Type the following:
cscript c:\inetpub\adminscripts\adsutil.vbs SET /w3svc/AppPools/Enable32BitAppOnWin64 False
3. Press enter

Access control list (ACL) inheritance is blocked_InhBlockPublicFolderTree

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 or Exchange Server 2010 setup cannot continue because the required permissions have not been able to propagate.

Exchange setup requires that inheritance for permissions be enabled on the following Exchange objects:

- Exchange Organization object
- Exchange Administrative Group object

- Exchange Servers container object
- Exchange Address List object
- Exchange Public Folder object
- Exchange Public Folder tree object

Failure to enable inheritance for permissions on these objects may result in mail flow problems, store mounting issues, and other service outages.

To resolve this issue, make sure that the "Allow permissions to propagate to this object and child objects" setting is enabled for the object, and then rerun Exchange Server 2007 or Exchange 2010 setup.

To re-enable permissions inheritance for an Exchange configuration object using Exchange Server 2003 Exchange System Manager

1. Enable the **Security** tab for the object properties box of Exchange System Manager by setting a registry parameter.
   a. Start Registry Editor (Regedt32.exe).
   b. Locate the following key in the registry:

HKEY_CURRENT_USER\Software\Microsoft\Exchange\EXAdmin

   c. On the **Edit** menu, click **New**, and then add the following registry value:

Value Name: ShowSecurityPage

Data Type: REG_DWORD

Radix: Binary

Value: 1

   d. Quit Registry Editor.

Note:

By default, the **Security** tab is not enabled in the configuration object properties box.

2. Open Exchange System Manager, find the object in question, right-click the object and select **Properties**.
3. Select the **Security** tab and then click **Advanced**.
4. Select **Allow inheritable permissions from the parent to propagate to this object and all child objects** to re-enable permissions inheritance.
5. Restart Exchange Server.

Caution:

If you incorrectly modify the attributes of Active Directory objects when you use ADSI Edit, the LDP tool, or another LDAP version 3 client, you may cause serious problems. These problems may require that you reinstall Microsoft Windows Server™ 2003, Exchange Server, or both. Modify Active Directory object attributes at your own risk.

To re-enable permissions inheritance for an Exchange configuration object using ADSIEdit from Exchange Server 2007 or Exchange Server 2010
1. Install ADSI Edit. 2. Launch ADSI Edit. Click **Start**, click **Run**, type **adsiedit.msc** in the text box, and then click OK. 3. Navigate to the object in question, right-click the object and select **Properties**. 4. Select the **Security** tab and then click **Advanced**. 5. Select **Allow inheritable permissions from the parent to propagate to this object and all child objects** to re-enable permissions inheritance. 6. Select **Ok** twice to apply the change. 7. Wait for Active Directory replication to propagate the changes or force Active Directory replication by following the guidance in Microsoft Knowledge Base article 232072, "Initiating Replication Between Active Directory Direct Replication Partners" (http://go.microsoft.com/fwlink/?linkid=3052&kbid=232072).

Setup failure occurred while installing a server role_InstallWatermark

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because a previous setup failure occurred while installing a server role.

Exchange 2007 setup requires that a failed server role installation be successfully re-installed, or removed from the setup process, before any other setup task can continue.

To address this issue, either reinstall just the failed server role(s), or remove the server role(s).

To reinstall the failed server role from the command line

1. Open a Command Prompt window, and then navigate to the installation files.

2. Run the following command:

```
Setup /roles:<Failed Server Role>
```

Select from one or more of the following roles, in a comma-separated list:

ClientAccess (or CA, or C)

EdgeTransport (or ET, or E)

Note:
The Edge Transport server role cannot coexist on the same computer with any other server role.

Note:
You must deploy the Edge Transport server role in the perimeter network and outside the Active Directory forest.

HubTransport (or HT, or H)

Mailbox (or MB, or M)

UnifiedMessaging (or UM, or U)

ManagementTools (or MT, or T)

Note:
If you specify ManagementTools, you will install the Exchange Management Console, the Exchange cmdlets for the Exchange Management Shell, the Exchange Help file, the Exchange Best Practices Analyzer Tool, and the Exchange Troubleshooting Assistant. If you install any other server role, the management tools will be installed automatically.

For example, to add the Hub Transport server role to an existing Mailbox server, type the following: **%LocalExchangeInstallationDir%\bin\Setup.com /role:HubTransport /Mode:Install**

Note:
If any Exchange Server 2007 server role previously installed successfully, the Setup wizard will run in maintenance mode. If no Exchange 2007 server roles were previously successfully installed, the Setup wizard will start from where it stopped.

To use the Exchange Server 2007 Setup wizard in maintenance mode to reinstall the failed server role

1. Log on to the server for which you want to reinstall a server role.
2. Open Control Panel and then double-click **Add or Remove Programs**.
3. On the **Change or Remove Programs** page, select **Microsoft Exchange Server**, and then click **Change**.
4. In the Exchange Server 2007 Setup wizard, on the **Exchange Maintenance Mode** page, click **Next**.
5. On the **Server Role Selection** page, select the check boxes for the server roles that you want to install, and then click **Next**.

Note:
The Edge Transport server role cannot coexist on the same computer with any other server

role.

Note:
You must deploy the Edge Transport server role in the perimeter network and outside the Active Directory forest.

Note:
If you select Management Tools, you will install the Exchange Management Console, the Exchange cmdlets for the Exchange Management Shell, and the Exchange Help file. The management tools will be installed automatically if you install any other server role.

6. If you selected **Hub Transport Role**, and if you are installing Exchange 2007 in a forest that has an existing Exchange Server 2003 or Exchange 2000 Server organization, on the **Mail Flow Settings** page, select a bridgehead server in the existing organization that is a member of the Exchange 2003 or Exchange 2000 routing group to which you want to create a routing group connector.
7. On the **Readiness Checks** page, view the status to determine if the organization and server role prerequisite checks completed successfully. If they have completed successfully, click **Install** to install Exchange 2007.
8. On the **Completion** page, click **Finish**.

To use the Exchange Server 2007 Setup wizard to reinstall the failed server role when no other server role was previously successfully installed

1. Follow the guidance in "How to Perform a Custom Installation Using Exchange Server 2007 Setup" (http://go.microsoft.com/fwlink/?LinkId=86648) in the Exchange Server 2007 product documentation.

To remove the failed server role

1. Follow the guidance in "How to Remove Exchange 2007 Server Roles" (http://go.microsoft.com/fwlink/?LinkId=86649) in the Exchange Server 2007 product documentation.

For More Information

For more information about how to install Exchange 2007 in unattended mode, see "How to Install Exchange 2007 in Unattended Mode" (http://go.microsoft.com/fwlink/?LinkId=86476).

Setup failure occurred while uninstalling a server role_InterruptedUninstallNotContinued

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Exchange Server 2010 setup cannot continue because a previous setup failure occurred while uninstalling a server role.

Exchange 2010 setup requires that a failed server role uninstall be successfully uninstalled before any other setup task can continue.

To address this issue, uninstall the failed server role(s), and then rerun setup.

For more information about how to remove a server role, see the Exchange 2007 product documentation topic, "How to Remove Exchange 2007 Server Roles" (http://go.microsoft.com/fwlink/?linkid=86649).

Cannot determine the name of the Active Directory site_InvalidADSite

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because this server does not appear to belong to a valid Active Directory® directory service site.

Microsoft Exchange setup requires that the server that is used for installation of Exchange 2007 belong to a valid Active Directory site.

To resolve this issue, verify that the local server is a member of a valid Active Directory site and rerun Exchange Server 2007 setup.

You can use the **/DsGetSite** option of the Nltest.exe command line tool to verify and display site membership. For more information, see "Nltest.exe: NLTest Overview" in the "Tools and Settings Collection" of *Windows Server 2003 Technical Reference* (http://go.microsoft.com/fwlink/?LinkId=27734).

For more information about Active Directory troubleshooting, see “Troubleshooting Active Directory Operations” in *Windows Server 2003: Operations* (http://go.microsoft.com/fwlink/?linkid=68099).

The local computer is a domain controller of a child domain_LocalComputerIsDCInChildDomain

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because the local computer is a domain controller for a child domain.

Exchange 2007 setup will not install on to a domain controller for a child domain unless the domain controller is a global catalog server.

To resolve this issue, promote the domain controller to a global catalog server or install Exchange 2007 to a non-domain controller, a member server, in the child domain, and then rerun the Microsoft Exchange setup.

To correct this warning by making the Exchange server a global catalog server

1. On the domain controller, click **Start**, point to **Programs**, click **Administrative Tools**, and then click **Active Directory Sites and Services**.
2. In the console tree, double-click **Sites**, double-click the name of the site, and then double-click **Servers**.

3. Double-click the target domain controller.
4. In the results pane, right-click **NTDS Settings**, and then click **Properties**.
5. On the **General** tab, click to select the **Global catalog** check box.
6. Restart the domain controller.

Active Directory domain is mixed mode_LocalDomainModeMixed

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because an existing Active Directory domain is not set to Microsoft Windows®°2000 Server native mode or better.

Exchange 2007 setup will create Universal Security Groups that can only exist in Windows 2000 Server native mode, or better, domains.

To resolve this issue, follow these steps to raise the domain functional level to at least the Windows 2000 Server native level, and then rerun Exchange 2007 setup.

To raise the domain functional level

1. Open Active Directory Domains and Trusts.
2. In the console tree, right-click the domain for which you want to raise functionality, and then click **Raise Domain Functional Level**.
3. In **Select an available domain functional level**, use one of the following procedures:
   - To raise the domain functional level to Windows 2000 Server native, click **Windows 2000 native**, and then click **Raise**.
   - To raise domain functional level to Windows Server® 2003, click **Windows Server 2003**, and then click **Raise.**

Caution:

If you have or will have any domain controllers running Windows NT® 4.0 and earlier, do not raise the domain functional level to Windows 2000 Server native. After the domain functional level is set to Windows 2000 Server native, it cannot be changed back to Windows 2000 Server mixed.

If you have or will have any domain controllers running Windows NT 4.0 and earlier or Windows 2000 Server, do not raise the domain functional level to Windows Server 2003. After the domain functional level is set to Windows Server 2003, it cannot be changed back to Windows 2000 Server mixed or Windows 2000 Server native.

The local domain needs to be updated_LocalDomainPrep

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because the logged on user does not have the account permissions that are required for domain preparation.

Exchange setup requires that the user who is logged on when **Setup /PrepareDomain** is run be a member of the Domain Administrators and Exchange Organization Administrators groups, or the Enterprise Admins group.

To resolve this issue, grant the logged on user Domain Admins and Exchange Organization Administrator permissions, enroll them in the Enterprise Admins group, or log on with an account that has those permissions and rerun Exchange 2007 setup.

For more information about Active Directory permissions that are needed with Microsoft Exchange, see "Working with Active Directory Permissions in Exchange Server" (http://go.microsoft.com/fwlink/?LinkId=47592).

IIS 6 Compatibility components not installed_LonghornIIS6MetabaseNotInstalled

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 Setup cannot continue its attempt to install the Client Access Server server role, the Mailbox server role, or the Exchange 2007 Administrative Tools on the following Windows operating systems:

- Windows Server 2008 R2
- Windows Server 2008
- Windows 7 (Administrative Tools only)
- Windows Vista (Administrative Tools only)

This problem occurs because the Internet Information Server (IIS) 6 Metabase Compatibility component and the IIS 6 Management Console component are not installed.

Exchange 2007 setup requires that the computer on which you are installing the Exchange 2007 Client Access server role, the Mailbox server role, or the Exchange 2007 Administrative Tools has the IIS 6 Metabase Compatibility component and the IIS 6 Management Console component installed.

To resolve this problem, install the IIS 6 Metabase Compatibility component on the destination computer, and then rerun Microsoft Exchange Setup.

Install the IIS 6.0 Management Compatibility Components in Windows Server 2008 R2 or in Windows Server by using the Server Manager tool

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS)**, and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS 6 Management Compatibility**.
4. Click to select the **IIS 6 Metabase Compatibility**, **IIS 6 WMI Compatibility**, and **IIS 6 Management Console** check boxes.
5. In the **Select Role Services** pane, click **Next**.
6. In the **Confirm Installations Selections** pane, click **Install**.
7. Click **Close** to exit the Add Role Services wizard.

Install the IIS 6.0 Management Compatibility Components in Windows 7 or in Windows Vista from Control Panel

1. Click **Start**, click **Control Panel**, click **Programs and Features**, and then click **Turn Windows features on or off**.

2. Open **Internet Information Services**.
3. Open **Web Management Tools**.
4. Open **IIS 6.0 Management Compatibility**.
5. Click to select the **IIS 6 Metabase and IIS 6 configuration compatibility**, **IIS 6 WMI Compatibility**, and **IIS 6 Management Console** check boxes.
6. Click **OK**.

IIS 6 Compatibility components not installed_LonghornIIS6MgmtConsoleNotInstalled

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 Setup cannot continue its attempt to install the Client Access Server server role, the Mailbox server role, or the Exchange 2007 Administrative Tools on the following Windows operating systems:

- Windows Server 2008 R2
- Windows Server 2008
- Windows 7 (Administrative Tools only)
- Windows Vista (Administrative Tools only)

This problem occurs because the Internet Information Server (IIS) 6 Metabase Compatibility component and the IIS 6 Management Console component are not installed.

Exchange 2007 setup requires that the computer on which you are installing the Exchange 2007 Client Access server role, the Mailbox server role, or the Exchange 2007 Administrative Tools has the IIS 6 Metabase Compatibility component and the IIS 6 Management Console component installed.

To resolve this problem, install the IIS 6 Metabase Compatibility component on the destination computer, and then rerun Microsoft Exchange Setup.

Install the IIS 6.0 Management Compatibility Components in Windows Server 2008 R2 or in

Windows Server by using the Server Manager tool

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS)**, and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS 6 Management Compatibility**.
4. Click to select the **IIS 6 Metabase Compatibility**, **IIS 6 WMI Compatibility**, and **IIS 6 Management Console** check boxes.
5. In the **Select Role Services** pane, click **Next**.
6. In the **Confirm Installations Selections** pane, click **Install**.
7. Click **Close** to exit the Add Role Services wizard.

Install the IIS 6.0 Management Compatibility Components in Windows 7 or in Windows Vista from Control Panel

1. Click **Start**, click **Control Panel**, click **Programs and Features**, and then click **Turn Windows features on or off**.
2. Open **Internet Information Services**.
3. Open **Web Management Tools**.
4. Open **IIS 6.0 Management Compatibility**.
5. Click to select the **IIS 6 Metabase and IIS 6 configuration compatibility**, **IIS 6 WMI Compatibility**, and **IIS 6 Management Console** check boxes.
6. Click **OK**.

IIS 7 component not installed_LonghornIIS7BasicAuthNotInstalled

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2010 Setup or Microsoft Exchange Server 2007 Setup cannot install the Client Access Server (CAS) role or the Mailbox server role on a Microsoft Windows Server 2008-

based computer or on a Windows Server 2008 R2-based computer because the required Internet Information Server (IIS) 7 components are not installed.

Exchange 2010 Setup and Exchange 2007 Setup require that the Windows Server 2008-based computer or the Windows Server 2008 R2-based computer on which you are installing the CAS role already has the following IIS 7 components installed.

Required IIS 7 Components for the CAS server role
Dynamic Content Compression
Static Content Compression
Basic Authentication
Windows Authentication
IIS 7 Digest Authentication
ASP.NET
Client Certificate Mapping
Directory Browsing
HTTP Errors
HTTP Logging
HTTP Redirection
Tracing
ISAPI Filters
Request Monitor
Static Content

Exchange 2010 Setup and Exchange 2007 Setup require that the Windows Server 2008-based computer or the Windows Server 2008 R2-based computer on which you are installing the Mailbox server role already has the following IIS 7 components installed.

Required IIS 7 Components for the Mailbox server role
Basic Authentication

Windows Authentication

To address this issue, follow the appropriate steps to install the required IIS 7 components on the destination computer, and then run Microsoft Exchange Setup again.

Install the IIS 7 Components for the CAS server role by using the Windows Server 2008 Server Manager

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS),** and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS**.
4. In the **Security** area, click to select the following check boxes:
   - **Basic Authentication**
   - **Digest Authentication**
   - **Windows Authentication**
5. In the **Performance** area, click to select the following check boxes:
   - **Static Compression**
   - **Dynamic Compression**
6. In the **Select Role Services** pane, click **Next**, and then click **Install** in the **Confirm Installations Selections** pane.
7. Click **Close** to exit the Add Role Services wizard.

Install the IIS 7 Components for the Mailbox server role by using the Windows Server 2008 Server Manager

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS),** and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS**.
4. In the **Security** area, click to select the following check boxes:
   - **Basic Authentication**
   - **Windows Authentication**
5. In the **Select Role Services** pane, click **Next**, and then click **Install** in the **Confirm Installations Selections** pane.
6. Click **Close** to exit the Add Role Services wizard.

IIS 7 component not installed_LonghornIIS7DigestAuthNotInstalled

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2010 Setup or Microsoft Exchange Server 2007 Setup cannot install the Client Access Server (CAS) role or the Mailbox server role on a Microsoft Windows Server 2008-based computer or on a Windows Server 2008 R2-based computer because the required Internet Information Server (IIS) 7 components are not installed.

Exchange 2010 Setup and Exchange 2007 Setup require that the Windows Server 2008-based computer or the Windows Server 2008 R2-based computer on which you are installing the CAS role already has the following IIS 7 components installed.

Required IIS 7 Components for the CAS server role
Dynamic Content Compression
Static Content Compression
Basic Authentication
Windows Authentication
IIS 7 Digest Authentication
ASP.NET
Client Certificate Mapping
Directory Browsing
HTTP Errors
HTTP Logging
HTTP Redirection
Tracing

ISAPI Filters
Request Monitor
Static Content

Exchange 2010 Setup and Exchange 2007 Setup require that the Windows Server 2008-based computer or the Windows Server 2008 R2-based computer on which you are installing the Mailbox server role already has the following IIS 7 components installed.

Required IIS 7 Components for the Mailbox server role
Basic Authentication
Windows Authentication

To address this issue, follow the appropriate steps to install the required IIS 7 components on the destination computer, and then run Microsoft Exchange Setup again.

Install the IIS 7 Components for the CAS server role by using the Windows Server 2008 Server Manager

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS),** and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS**.
4. In the **Security** area, click to select the following check boxes:
   - **Basic Authentication**
   - **Digest Authentication**
   - **Windows Authentication**
5. In the **Performance** area, click to select the following check boxes:
   - **Static Compression**
   - **Dynamic Compression**
6. In the **Select Role Services** pane, click **Next**, and then click **Install** in the **Confirm Installations Selections** pane.
7. Click **Close** to exit the Add Role Services wizard.

Install the IIS 7 Components for the Mailbox server role by using the Windows Server 2008 Server Manager

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS),** and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS**.
4. In the **Security** area, click to select the following check boxes:
   - **Basic Authentication**

- **Windows Authentication**

5. In the **Select Role Services** pane, click **Next**, and then click **Install** in the **Confirm Installations Selections** pane.
6. Click **Close** to exit the Add Role Services wizard.

IIS 7 component not installed_LonghornIIS7HttpCompressionDynamicNotInstalled

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2010 Setup or Microsoft Exchange Server 2007 Setup cannot install the Client Access Server (CAS) role or the Mailbox server role on a Microsoft Windows Server 2008-based computer or on a Windows Server 2008 R2-based computer because the required Internet Information Server (IIS) 7 components are not installed.

Exchange 2010 Setup and Exchange 2007 Setup require that the Windows Server 2008-based computer or the Windows Server 2008 R2-based computer on which you are installing the CAS role already has the following IIS 7 components installed.

Required IIS 7 Components for the CAS server role
Dynamic Content Compression
Static Content Compression
Basic Authentication
Windows Authentication
IIS 7 Digest Authentication

ASP.NET
Client Certificate Mapping
Directory Browsing
HTTP Errors
HTTP Logging
HTTP Redirection
Tracing
ISAPI Filters
Request Monitor
Static Content

Exchange 2010 Setup and Exchange 2007 Setup require that the Windows Server 2008-based computer or the Windows Server 2008 R2-based computer on which you are installing the Mailbox server role already has the following IIS 7 components installed.

Required IIS 7 Components for the Mailbox server role
Basic Authentication
Windows Authentication

To address this issue, follow the appropriate steps to install the required IIS 7 components on the destination computer, and then run Microsoft Exchange Setup again.

Install the IIS 7 Components for the CAS server role by using the Windows Server 2008 Server Manager

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS),** and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS**.
4. In the **Security** area, click to select the following check boxes:
   - **Basic Authentication**
   - **Digest Authentication**
   - **Windows Authentication**
5. In the **Performance** area, click to select the following check boxes:

- **Static Compression**
- **Dynamic Compression**

6. In the **Select Role Services** pane, click **Next**, and then click **Install** in the **Confirm Installations Selections** pane.
7. Click **Close** to exit the Add Role Services wizard.

Install the IIS 7 Components for the Mailbox server role by using the Windows Server 2008 Server Manager

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS),** and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS**.
4. In the **Security** area, click to select the following check boxes:
   - **Basic Authentication**
   - **Windows Authentication**
5. In the **Select Role Services** pane, click **Next**, and then click **Install** in the **Confirm Installations Selections** pane.
6. Click **Close** to exit the Add Role Services wizard.

IIS 7 component not installed_LonghornIIS7HttpCompressionStaticNotInstalled

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2010 Setup or Microsoft Exchange Server 2007 Setup cannot install the Client Access Server (CAS) role or the Mailbox server role on a Microsoft Windows Server 2008-based computer or on a Windows Server 2008 R2-based computer because the required Internet Information Server (IIS) 7 components are not installed.

Exchange 2010 Setup and Exchange 2007 Setup require that the Windows Server 2008-based

computer or the Windows Server 2008 R2-based computer on which you are installing the CAS role already has the following IIS 7 components installed.

Required IIS 7 Components for the CAS server role
Dynamic Content Compression
Static Content Compression
Basic Authentication
Windows Authentication
IIS 7 Digest Authentication
ASP.NET
Client Certificate Mapping
Directory Browsing
HTTP Errors
HTTP Logging
HTTP Redirection
Tracing
ISAPI Filters
Request Monitor
Static Content

Exchange 2010 Setup and Exchange 2007 Setup require that the Windows Server 2008-based computer or the Windows Server 2008 R2-based computer on which you are installing the Mailbox server role already has the following IIS 7 components installed.

Required IIS 7 Components for the Mailbox server role
Basic Authentication
Windows Authentication

To address this issue, follow the appropriate steps to install the required IIS 7 components on the destination computer, and then run Microsoft Exchange Setup again.

Install the IIS 7 Components for the CAS server role by using the Windows Server 2008 Server Manager

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS),** and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS**.
4. In the **Security** area, click to select the following check boxes:
   - **Basic Authentication**
   - **Digest Authentication**
   - **Windows Authentication**
5. In the **Performance** area, click to select the following check boxes:
   - **Static Compression**
   - **Dynamic Compression**
6. In the **Select Role Services** pane, click **Next**, and then click **Install** in the **Confirm Installations Selections** pane.
7. Click **Close** to exit the Add Role Services wizard.

Install the IIS 7 Components for the Mailbox server role by using the Windows Server 2008 Server Manager

1. Click **Start**, click **Administrative Tools**, and then click **Server Manager**.
2. In the navigation pane, expand **Roles**, right-click **Web Server (IIS),** and then click **Add Role Services**.
3. In the **Select Role Services** pane, scroll down to **IIS**.
4. In the **Security** area, click to select the following check boxes:
   - **Basic Authentication**
   - **Windows Authentication**
5. In the **Select Role Services** pane, click **Next**, and then click **Install** in the **Confirm Installations Selections** pane.
6. Click **Close** to exit the Add Role Services wizard.

The World Wide Web Publishing Service is disabled or missing_W3SVCDisabledOrNotInstalled

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft® Exchange Server 2007 setup cannot continue because its attempt to install the Mailbox Server or Client Access role found that the World Wide Web Publishing Service is either disabled or not installed on this computer.

Exchange 2007 setup requires the computer that you are installing Microsoft Exchange to have the World Wide Web Publishing Service installed and set to something other than disabled.

To resolve this issue, verify that the World Wide Web Publishing service is installed and not disabled on the local computer, and then rerun Microsoft Exchange setup.

To verify that the World Wide Web Publishing Service is installed and not disabled

1. Right-click **My Computer** on the desktop, and then click **Manage**.
2. Expand the **Services and Applications** node, and then click the **Services** node.
3. In the right pane, locate the **World Wide Web Publishing Service**.

If the **World Wide Web Publishing Service** is not displayed in the list of services installed, follow the steps in the procedure below to install it.

4. If the **World Wide Web Publishing Service** is displayed but has a status other than **Started**, continue with the steps below to start it.
5. Right-click **World Wide Web Publishing Service**, and then click **Properties**.
6. Verify the **Startup Type** is **Automatic** and the **Service status** is set to **Started**.
7. Click **Apply**, and then click **OK**.

To install the World Wide Web Publishing Service

1. On the **Start** menu, select **Settings**, **Control Panel**, and then click **Add or Remove Programs**
2. Click **Add/Remove Windows Components**.
3. In the **Components** list, select the **Application Server** check box, and then click **Details**.
4. Select **Internet Information Services Manager**, and then click **Details**.
5. Select **World Wide Web Service**, and then select the check box.
6. Click **OK** two times to return to the **Components** list, and then click **Next**.
7. Click **Finish** when the IIS service is installed.

OAB server has been

deleted_OffLineABServerDeleted

Planning and deployment > Deployment reference > Exchange 2013 readiness checks >

Applies to: *Exchange Server*

Topic Last Modified: *2012-06-05*

The content in this topic hasn't been updated for Microsoft Exchange Server 2013. While it hasn't been updated yet, it may still be applicable to Exchange 2013. If you still need help, check out the community resources below.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Microsoft Exchange Server 2007 setup cannot continue because its attempt to install the Client Access server role failed. The Exchange Mailbox server designated to host the Offline Address Book (OAB) has been deleted.

Exchange 2007 setup requires that there be a valid Exchange Mailbox server hosting OAB generation for the Client Access server role installation to continue.

To address this issue, designate a valid Exchange Mailbox server as OAB generation host, and then rerun Exchange 2007 setup.

For information about how to designate an Exchange server to be the host for Offline Address Book generation, see "How to Create an Offline Address Book" (http://go.microsoft.com/fwlink/?LinkId=86314) in the Exchange 2007 product documentation.

Multi-tenancy in Exchange 2013

Exchange Server 2013 > Planning and deployment >

Applies to: *Exchange Server 2013, Exchange Server, Exchange Online*

Topic Last Modified: *2012-12-10*

A multi-tenant (hosted) Exchange 2013 deployment is defined as one where the Exchange organization is configured to host multiple and discrete organizations or business units (the tenants) that ordinarily don’t share email, data, users, global address lists (GALs), or other commonly used Exchange objects. This sharing of hardware, software and resources (all while maintaining a logical separation between tenants), allows organizations to leverage the simplicity of a standard Exchange deployment while providing multi-tenant functionality and services to meet their hosting needs.

Multi-tenancy in Exchange 2013 organizations

In Exchange 2013, we continue to support hosting by using a standard, on-premises Exchange installation similar to the approach used in Exchange 2010 Service Pack 2 (SP2). We discontinued the `/hosting` mode switch and are emphasizing the use of address book policies (ABPs) in combination with hosting management solutions and automation tools provided by approved Independent Software Vendors (ISVs). These solutions are built on a framework of Microsoft-approved configuration guidelines and practices and will offer Exchange organizations an easier, more robust way to provide hosting services and features.

Exchange 2013 supports multi-tenancy by leveraging the following primary components and features:

- **Active Directory** Instead of having separate *ExchangeOrganization* Active Directory containers for each business unit in a multi-tenant Exchange organization, Exchange 2013 multi-tenancy is supported by using a single *ExchangeOrganization* Active Directory container. This allows for a simpler Active Directory structure and reduces the likelihood of Active Directory-related permission problems.

To learn more about Active Directory changes in Exchange 2013, see Active Directory.

- **Address book policies (ABPs)** Introduced in Exchange 2010 SP2, ABPs are used in Exchange 2013 to control user access to an address list, the global address list (GAL), and an offline address books (OABs) in the Exchange organization. ABPs group these different Active Directory objects into a single, virtual object that can be assigned to individual users and to create a logical grouping of these resources along a multi-tenant organizational structure. ABP functionality in Exchange 2013 is similar to what it was in Exchange 2010 SP2.

To learn more about ABPs in Exchange 2013, see Address book policies.

- **Hosting management solutions** Some administrators using Exchange 2013 to provide a hosted Exchange solution will benefit from using a customized hosting management approach. Due to some limitations of the Exchange Administration Center (EAC), Microsoft works with third-party vendors to assist them in the development of control panel and automation solutions that are in compliance with the guidelines and approved framework for hosted Exchange 2013 organizations. We recommend that organizations configuring a hosted Exchange solution leverage these tools to manage their hosted organizations where circumstances require it.

To learn more about hosted management solutions, including validated solution vendors, see Exchange Server 2013 hosting and multi-tenancy solutions and guidance

Permissions

Exchange Server 2013 >

Applies to: *Exchange Server*

Topic Last Modified: *2013-05-17*

Microsoft Exchange Server 2013 includes a large set of predefined permissions, based on the Role Based Access Control (RBAC) permissions model, which you can use right away to easily grant permissions to your administrators and users. You can use the permissions features in Exchange 2013 so that you can get your new organization up and running quickly.

Note:

Several RBAC features and concepts aren't discussed in this topic because they're advanced features. If the functionality discussed in this topic doesn't meet your needs, and you want to further customize your permissions model, see Understanding Role Based Access Control.

Looking for a list of all permissions topics? See Permissions documentation.

Contents

Role-based permissions

In Exchange 2013, the permissions that you grant to administrators and users are based on management roles. A role defines the set of tasks that an administrator or user can perform. For example, a management role called `Mail Recipients` defines the tasks that someone can perform on a set of mailboxes, contacts, and distribution groups. When a role is assigned to an administrator or user, that person is granted the permissions provided by the role.

There are two types of roles, administrative roles and end-user roles:

- **Administrative roles** These roles contain permissions that can be assigned to administrators or specialist users using role groups that manage a part of the Exchange organization, such as recipients, servers, or databases.
- **End-user roles** These roles, assigned using role assignment policies, enable users to manage aspects of their own mailbox and distribution groups that they own. End-user roles begin with the prefix `My`.

Roles give permissions to perform tasks to administrators and users by making cmdlets available to those who are assigned the roles. Because the Exchange Administration Center (EAC) and Exchange Management Shell use cmdlets to manage Exchange, granting access to a cmdlet gives the administrator or user permission to perform the task in each of the Exchange management interfaces.

Exchange 2013 includes approximately 86 roles that you can use to grant permissions. For a list of roles included with Exchange 2013, see Built-in management roles.

Return to top

Role groups and role assignment policies

Roles grant permissions to perform tasks in Exchange 2013, but you need an easy way to assign them to administrators and users. Exchange 2013 provides you with the following to help you do that:

- **Role groups** Role groups enable you to grant permissions to administrators and specialist users.
- **Role assignment policies** Role assignment policies enable you to grant permissions to end users to change settings on their own mailbox or distribution groups that they own.

For more information about role groups and role assignment policies, see the following sections.

Role groups

Every administrator that manages Exchange 2013 must be assigned at least one or more roles. Administrators might have more than one role because they may perform job functions that span multiple areas in Exchange. For example, one administrator might manage both recipients and Exchange servers. In this case, that administrator might be assigned both the `Mail Recipients` and `Exchange Servers` roles.

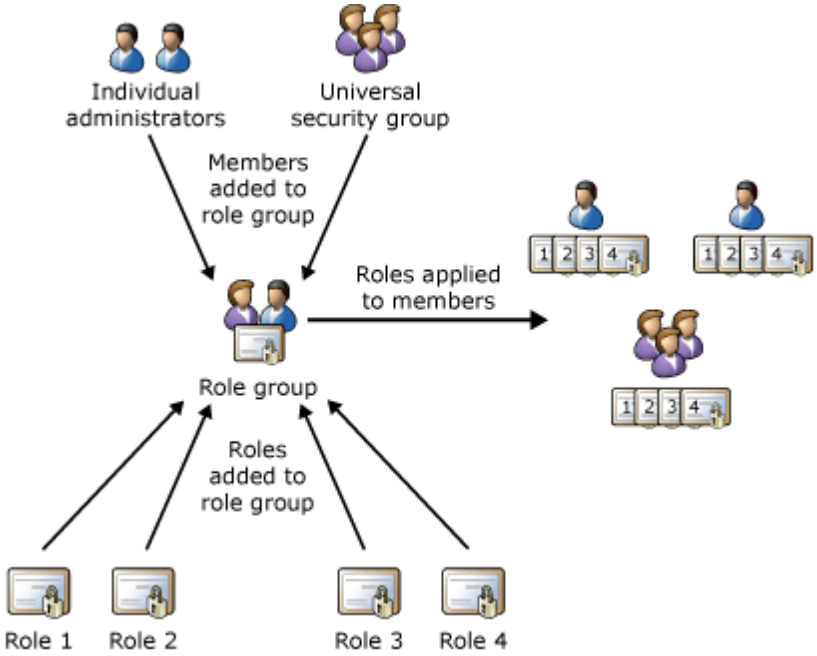
To make it easier to assign multiple roles to an administrator, Exchange 2013 includes role groups. Role groups are special universal security groups (USGs) used by Exchange 2013 that can contain Active Directory users, USGs, and other role groups. When a role is assigned to a role group, the permissions granted by the role are granted to all the members of the role group. This enables you to assign many roles to many role group members at once. Role groups typically encompass broader management areas, such as recipient management. They're used only with administrative roles, and not end-user roles.

Note:

It's possible to assign a role directly to a user or USG without using a role group. However, that method of role assignment is an advanced procedure and isn't covered in this topic. We recommend that you use role groups to manage permissions.

The following figure shows the relationship between users, role groups, and roles.

Roles, role groups, and role group members



Exchange 2013 includes several built-in role groups, each one providing permissions to manage specific areas in Exchange 2013. Some role groups may overlap with others. The following table lists each role group with a description of its use. If you want to see the roles assigned to each role group, click the name of the role group in the "Role group" column, and then open the "Management Roles Assigned to This Role Group" section.

Built-in role groups

Role group	Description
Organization Management	Administrators who are members of the Organization Management role group have administrative access to the entire Exchange 2013 organization and can perform almost any task against any Exchange 2013 object, with some exceptions, such as the `Discovery Management` role. **Important:** Because the Organization Management role group is a powerful role, only users or USGs that perform organizational-level administrative tasks that can potentially impact the entire Exchange organization should be members of this role group.
View-only Organization Management	Administrators who are members of the View Only Organization Management role group can view the properties of any object in the

	Exchange organization.
Recipient Management	Administrators who are members of the Recipient Management role group have administrative access to create or modify Exchange 2013 recipients within the Exchange 2013 organization.
UM Management	Administrators who are members of the UM Management role group can manage features in the Exchange organization such as Unified Messaging (UM) service configuration, UM properties on mailboxes, UM prompts, and UM auto attendant configuration.
Help Desk	The Help Desk role group, by default, enables members to view and modify the Microsoft Office Outlook Web App options of any user in the organization. These options might include modifying the user's display name, address, and phone number. They don't include options that aren't available in Outlook Web App options, such as modifying the size of a mailbox or configuring the mailbox database on which a mailbox is located.
Hygiene Management	Administrators who are members of the Hygiene Management role group can configure the antivirus and anti-spam features of Exchange 2013. Third-party programs that integrate with Exchange 2013 can add service accounts to this role group to grant those programs access to the cmdlets required to retrieve and configure the Exchange configuration.

Records Management	Users who are members of the Records Management role group can configure compliance features, such as retention policy tags, message classifications, and transport rules.
Discovery Management	Administrators or users who are members of the Discovery Management role group can perform searches of mailboxes in the Exchange organization for data that meets specific criteria and can also configure legal holds on mailboxes.
Public Folder Management	Administrators who are members of the Public Folder Management role group can manage public folders on servers running Exchange 2013.
Server Management	Administrators who are members of the Server Management role group can configure server-specific configuration of transport, Unified Messaging, client access, and mailbox features such as database copies, certificates, transport queues and Send connectors, virtual directories, and client access protocols.
Delegated Setup	Administrators who are members of the Delegated Setup role group can deploy servers running Exchange 2013 that have been previously provisioned by a member of the Organization Management role group.
Compliance Management	Users who are members of the Compliance Management role group can configure and manage Exchange compliance settings in

	accordance with their organization's policy.

If you work in a small organization that has only a few administrators, you might need to add those administrators to the Organization Management role group only, and you may never need to use the other role groups. If you work in a larger organization, you might have administrators who perform specific tasks administering Exchange, such as recipient or server management. In those cases, you might add one administrator to the Recipient Management role group, and another administrator to the Server Management role group. Those administrators can then manage their specific areas of Exchange 2013 but won't have permissions to manage areas they're not responsible for.

If the built-in role groups in Exchange 2013 don't match the job function of your administrators, you can create role groups and add roles to them. For more information, see Work with Role Groups later in this topic.

Return to top

Role assignment policies

Exchange 2013 provides role assignment policies so that you can control what settings your users can configure on their own mailboxes and on distribution groups they own. These settings include their display name, contact information, voice mail settings, and distribution group membership.

Your Exchange 2013 organization can have multiple role assignment policies that provide different levels of permissions for the different types of users in your organizations. Some users can be allowed to change their address or create distribution groups, while others can't, depending on the role assignment policy associated with their mailbox. Role assignment policies are added directly to mailboxes, and each mailbox can only be associated with one role assignment policy at a time.

Of the role assignment policies in your organization, one is marked as default. The default role assignment policy is associated with new mailboxes that aren't explicitly assigned a specific role assignment policy when they're created. The default role assignment policy should contain the permissions that should be applied to the majority of your mailboxes.

Permissions are added to role assignment policies using end-user roles. End-user roles begin with `My` and grant permissions for users to manage only their mailbox or distribution groups they own. They can't be used to manage any other mailbox. Only end-user roles can be assigned to role assignment policies.

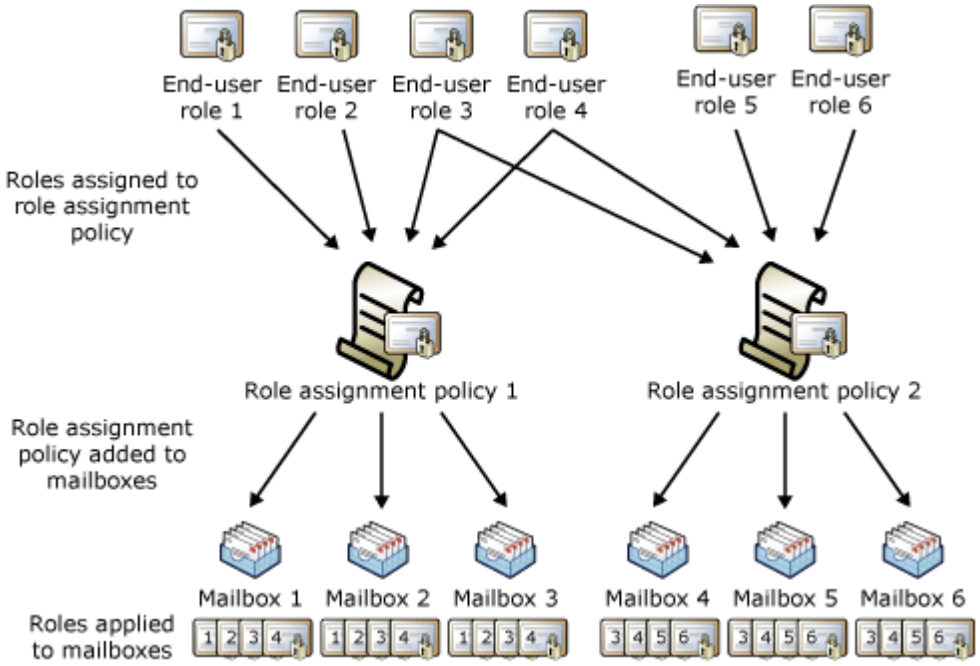
When an end-user role is assigned to a role assignment policy, all of the mailboxes associated with that role assignment policy receive the permissions granted by the role. This enables you to add or remove permissions to sets of users without having to configure individual mailboxes. The following figure shows:

- End-user roles are assigned to role assignment policies. Role assignment policies can share the same end-user roles.
- Role assignment policies are associated with mailboxes. Each mailbox can only be associated

with one role assignment policy.

- After a mailbox is associated with a role assignment policy, the end-user roles are applied to that mailbox. The permissions granted by the roles are granted to the user of the mailbox.

Roles, role assignment policies, and mailboxes

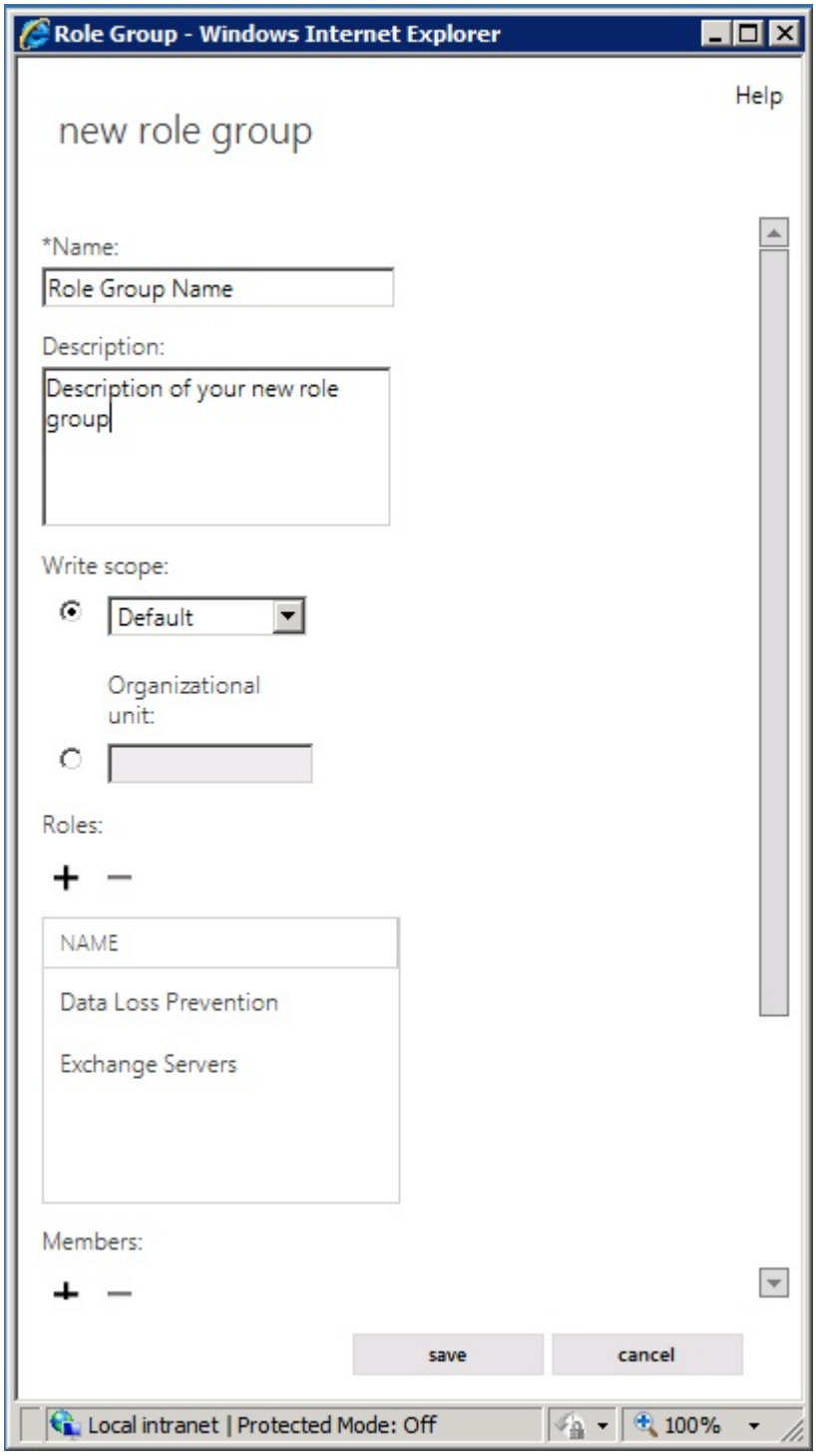


The Default Role Assignment Policy role assignment policy is included with Exchange 2013. As the name implies, it's the default role assignment policy. If you want to change the permissions provided by this role assignment policy, or if you want to create role assignment policies, see Work with Role Assignment Policies later in this topic.

Work with role groups

To manage your permissions using role groups in Exchange 2013, we recommend that you use the Exchange Administration Center. When you use the EAC to manage role groups, you can add and remove roles and members, create role groups, and copy role groups with a few clicks of your mouse. The EAC provides simple dialog boxes, such as the **new role group** dialog box, shown in the following figure, to perform these tasks.

New role group dialog box in the EAC



Exchange 2013 includes several role groups that separate permissions into specific administrative areas. If these existing role groups provide the permissions your administrators need to manage your Exchange 2013 organization, you need only add your administrators as members of the appropriate role groups. After you add administrators to a role group, they can administer the features that relate to that role group. To add or remove members to or from a role group, open the role group in the EAC, and then add or remove members from the membership list. For a list of built-in role groups, see Built-in role groups.

Important:
If an administrator is a member of more than one role group, Exchange 2013 grants the administrator all of the permissions provided by the role groups he or she is a member of.

If none of the role groups included with Exchange 2013 have the permissions you need, you can use

the EAC to create a role group and add the roles that have the permissions you need. For your new role group, you will:

1. Choose a name for your role group.
2. Select the roles you want to add to the role group.
3. Add members to the role group.
4. Save the role group.

After you create the role group, you manage it like any other role group.

If there's an existing role group that has some, but not all of the permissions you need, you can copy it and then make changes to create a role group. You can copy an existing role group and make changes to it, without affecting the original role group. As part of copying the role group, you can add a new name and description, add and remove roles to and from the new role group, and add new members. When you create or copy a role group, you use the same dialog box that's shown in the preceding figure.

Existing role groups can also be modified. You can add and remove roles from existing role groups, and add and remove members from it at the same time, using an EAC dialog box similar to the one in the preceding figure. By adding and removing roles to and from role groups, you turn on and off administrative features for members of that role group. For a list of roles you can add to a role group, see Built-in management roles.

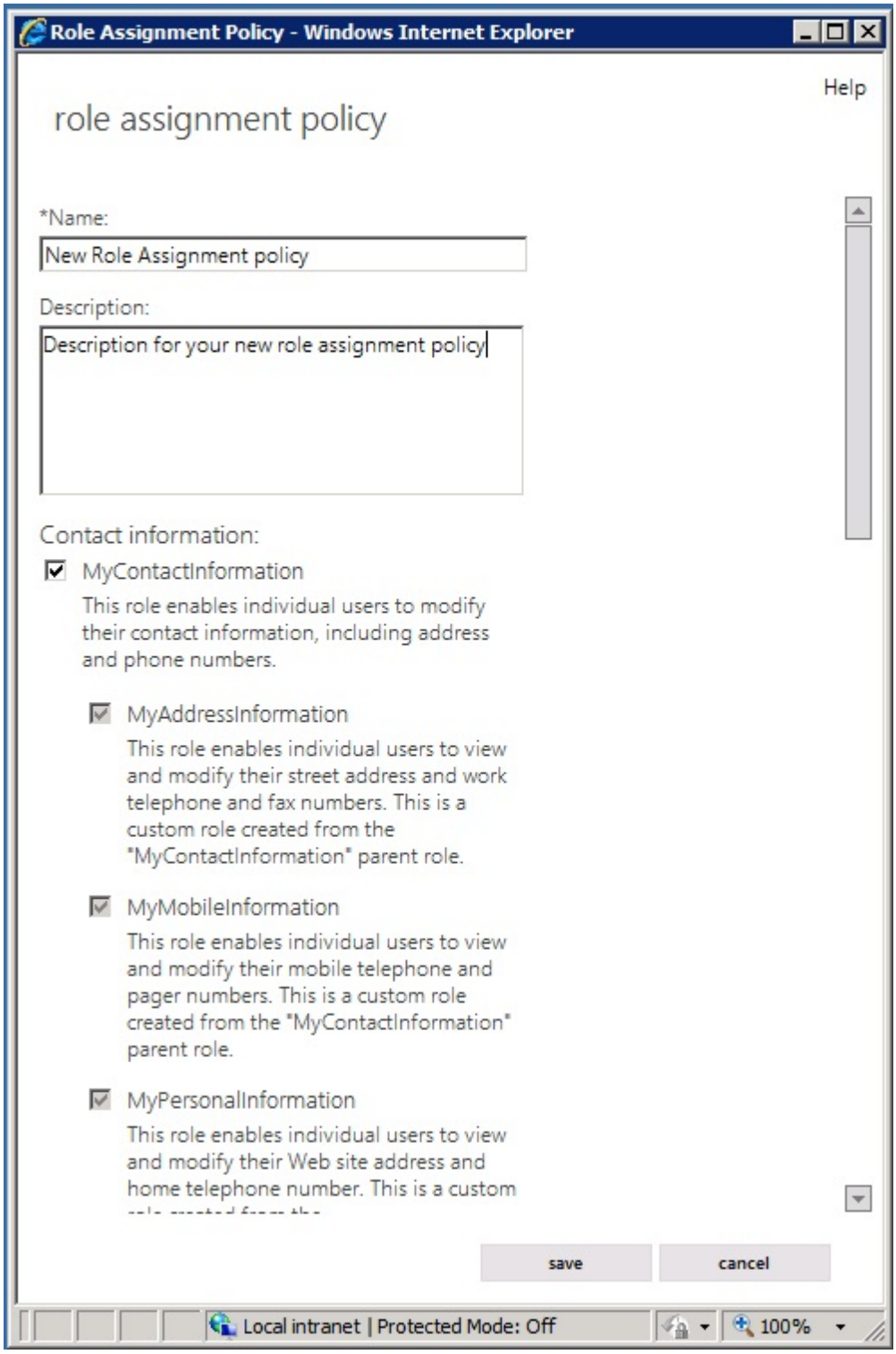
Note:
Although you can change which roles are assigned to built-in role groups, we recommend that you copy built-in role groups, modify the role group copy, and then add members to the role group copy.

Return to top

Work with role assignment policies

To manage the permissions that you grant end users to manage their own mailbox in Exchange 2013, we recommend that you use the EAC. When you use the EAC to manage end-user permissions, you can add roles, remove roles, and create role assignment policies with a few clicks of your mouse. The EAC provides simple dialog boxes, such as the **role assignment policy** dialog box, shown in the following figure, to perform these tasks.

Role assignment policy dialog box in the EAC



Exchange 2013 includes a role assignment policy named Default Role Assignment Policy. This role assignment policy enables users whose mailboxes are associated with it to do the following:

- Join or leave distribution groups that allow members to manage their own membership.
- View and modify basic mailbox settings on their own mailbox, such as Inbox rules, spelling behavior, junk mail settings, and Microsoft ActiveSync devices.
- Modify their contact information, such as work address and phone number, mobile phone number, and pager number.
- Create, modify, or view text message settings.
- View or modify voice mail settings.
- View and modify their marketplace apps.
- Create team mailboxes and connect them to Microsoft SharePoint lists.

If you want to add or remove permissions from the Default Role Assignment Policy or any other role assignment policy, you can use the EAC. The dialog box you use is similar to the one in the preceding figure. When you open the role assignment policy in the EAC, select the check box next to the roles you want to assign to it or clear the check box next to the roles you want to remove. The change you make to the role assignment policy is applied to every mailbox associated with it.

If you want to assign different end-user permissions to the various types of users in your organization, you can create role assignment policies. When you create a role assignment policy, you see a dialog box similar to the one in the preceding figure. You can specify a new name for the role assignment policy, and then select the roles you want to assign to the role assignment policy. After you create a role assignment policy, you can associate it with mailboxes using the EAC.

If you want to change which role assignment policy is the default, you must use the Shell. When you change the default role assignment policy, any mailboxes that are created will be associated with the new default role assignment policy if one wasn't explicitly specified. The role assignment policy associated with existing mailboxes doesn't change when you select a new default role assignment policy.

Note:

If you select a check box for a role that has child roles, the check boxes for the child roles are also selected. If you clear the check box for a role with child roles, the check boxes for the child roles are also cleared.
For detailed steps about how to create role assignment policies or make changes to existing role assignment policies, see the following topics:

- Manage role assignment policies
- Change the assignment policy on a mailbox

Return to top

Permissions documentation

The following table contains links to topics that will help you learn about and manage permissions in Exchange 2013.

Topic	Description
Understanding Role Based Access Control	Learn about each of the components that make up RBAC and how you can create advanced permissions models if role groups and management roles aren’t enough.
Understanding multiple-forest permissions	Learn about implementing RBAC permissions in organizations with account and resource forests.

Understanding split permissions	Learn about splitting Exchange and security principal management using RBAC and Active Directory split permissions.
Understanding permissions coexistence with Exchange 2007 and Exchange 2010	Configure permissions to enable administration of Exchange 2013 in existing Exchange 2007 and Exchange 2010 organizations.
Manage role groups	Configure permissions for Exchange administrators and specialist users using role groups.
Manage role group members	Add members to and from role groups. By adding and removing members to and from role groups, you configure who's able to administer Exchange features.
Manage linked role groups	Configure permissions for Exchange administrators and specialist users in multi-forest Exchange deployments.
Manage role assignment policies	Configure which features end-users have access to on their mailboxes using role assignment policies.
Change the assignment policy on a mailbox	Configure which role assignment policy is applied to one or more mailboxes.
Create linked role groups that mirror built-in role groups	Re-create the built-in role groups as linked role groups in multi-forest Exchange deployments.
View effective permissions	View who has permissions to administer Exchange features.
Feature permissions	Learn more about the permissions required to manage Exchange features and services.

Advanced permissions	Use advanced RBAC features to create highly customized permission models to fit the needs of any organization.

Understanding Role Based Access Control

Exchange Server 2013 > Permissions >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-05*

Role Based Access Control (RBAC) is the permissions model used in Microsoft Exchange Server 2013. With RBAC, you don't need to modify and manage access control lists (ACLs), which was done in Exchange Server 2007. ACLs created several challenges in Exchange 2007, such as modifying ACLs without causing unintended consequences, maintaining ACL modifications through upgrades, and troubleshooting problems that occurred due to using ACLs in a nonstandard way.

RBAC enables you to control, at both broad and granular levels, what administrators and end-users can do. RBAC also enables you to more closely align the roles you assign users and administrators to the actual roles they hold within your organization. In Exchange 2007, the server permissions model applied only to the administrators who managed the Exchange 2007 infrastructure. In Exchange 2013, RBAC now controls both the administrative tasks that can be performed and the extent to which users can now administer their own mailbox and distribution groups.

RBAC has two primary ways of assigning permissions to users in your organization, depending on whether the user is an administrator or specialist user, or an end-user: management role groups and management role assignment policies. Each method associates users with the permissions they need to perform their jobs. A third, more advanced method, direct user role assignment, can also be used. The following sections in this topic explain RBAC and provide examples of its use.

Note:

This topic focuses on advanced RBAC functionality. If you want to manage basic Exchange 2013 permissions, such as using the Exchange admin center (EAC) to add and remove members to and from role groups, create and modify role groups, or create and modify role assignment policies, see Permissions.

Contents

Management role groups

Management role groups associate management roles to a group of administrators or specialist users. Administrators manage a broad Exchange organization or recipient configuration. Specialist users manage the specific features of Exchange, such as compliance. Or they may have limited management abilities, such as Help desk members, but aren't given broad administrative rights. Role groups typically associate administrative management roles that enable administrators and specialist users to manage the configuration of their organization and recipients. For example, whether administrators can manage recipients or use mailbox discovery features is controlled using role groups.

Adding or removing users to or from role groups is how you most often assign permissions to administrators or specialist users. For more information, see Understanding management role groups.

Role groups consist of the following components that define what administrators and specialist users can do:

- **Management role group** The *management role group* is a special universal security group (USG) that contains mailboxes, users, USGs, and other role groups that are members of the role group. This is where you add and remove members, and it's also what management roles are assigned to. The combination of all the roles on a role group defines everything that users added to a role group can manage in the Exchange organization.
- **Management role** A *management role* is a container for a grouping of management role entries. Roles are used to define the specific tasks that can be performed by the members of a role group that's assigned the role. A *management role entry* is a cmdlet, script, or special permission that enables each specific task in a role to be performed. For more information, see Understanding management roles.
- **Management role assignment** A *management role assignment* links a role and a role group. Assigning a role to a role group grants members of the role group the ability to use the cmdlets and parameters defined in the role. Role assignments can use management scopes to control where the assignment can be used. For more information, see Understanding management role assignments.
- **Management role scope** A *management role scope* is the scope of influence or impact on a role assignment. When a role is assigned with a scope to a role group, the management scope targets specifically what objects that assignment is allowed to manage. The assignment, and its scope, are then given to the members of the role group, and restrict what those members can manage. A scope can consist of a list of servers or databases, organizational units (OUs), or filters on server, database or recipient objects. For more information, see Understanding management role

scopes.

When you add a user to a role group, the user is given all of the roles assigned to the role group. If scopes are applied to any of the role assignments between the role group and the roles, those scopes control what server configuration or recipients the user can manage.

If you want to change what roles are assigned to role groups, you need to change the role assignments that link the role groups to roles. Unless the assignments built into Exchange 2013 don't suit your needs, you won't have to change these assignments. For more information, see Understanding management role assignments.

For more information about role groups, see Understanding management role groups.

Management role assignment policies

Management role assignment policies associate end-user management roles to users. Role assignment policies consist of roles that control what a user can do with his or her mailbox or distribution groups. These roles don't allow management of features that aren't directly associated with the user. When you create a role assignment policy, you define everything a user can do with his or her mailbox. For example, a role assignment policy may allow a user to set the display name, set up voice mail, and configure Inbox rules. Another role assignment policy might allow a user to change the address, use text messaging, and set up distribution groups. Every user with an Exchange 2013 mailbox, including administrators, is given a role assignment policy by default. You can decide which role assignment policy should be assigned by default, choose what the default role assignment policy should include, override the default for certain mailboxes, or not assign role assignment policies by default at all.

Assigning a user to an assignment policy is how you most often manage permissions for users to manage their own mailbox and distribution group options. For more information, see Understanding management role assignment policies.

Role assignment policies consist of the following components that define what users can do with their own mailboxes. Notice that some of the same components also apply to role groups. When used with role assignment policies, these components are limited to enable users to manage only their own mailbox:

- **Management role assignment policy** The *management role assignment policy* is a special object in Exchange 2013. Users are associated with the role assignment policy when their mailboxes are created or if you change the role assignment policy on a mailbox. This is also what you assign end-user management roles to. The combination of all the roles on a role assignment policy defines everything that the user can manage on his or her mailbox or distribution groups.
- **Management role** A *management role* is a container for a grouping of management role entries. Roles are used to define the specific tasks that a user can do with his or her mailbox or distribution groups. A *management role entry* is a cmdlet, script or special permission that enables each specific task in a management role to be performed. You can only use end-user roles with role assignment policies. For more information, see Understanding management roles.

- **Management role assignment** A *management role assignment* is the link between a role and a role assignment policy. Assigning a role to a role assignment policy grants the ability to use the cmdlets and parameters defined in the role. When you create a role assignment between a role assignment policy and a role, you can't specify any scope. The scope applied by the assignment is either `Self` or `MyGAL`. All role assignments are scoped to the user's mailbox or distribution groups. For more information, see Understanding management role assignments.

If you want to change what roles are assigned to role assignment policies, you need to change the role assignments that link the role assignment policies to roles. Unless the assignments built into Exchange 2013 don't suit your needs, you won't have to change these assignments. For more information, see Understanding management role assignments.

For more information, see Understanding management role assignment policies.

Direct user role assignment

Direct role assignment is an advanced method for assigning management roles directly to a user or USG without using a role group or role assignment policy. Direct role assignments can be useful when you need to provide a granular set of permissions to a specific user and no others. However, using direct role assignments can significantly increase the complexity of your permissions model. If a user changes jobs or leaves the company, you need to manually remove the assignments and add them to the new employee. We recommend that you use role groups to assign permissions to administrators and specialist users, and role assignment policies to assign permissions to users.

For more information about direct user assignment, see Understanding management role assignments.

Summary and examples

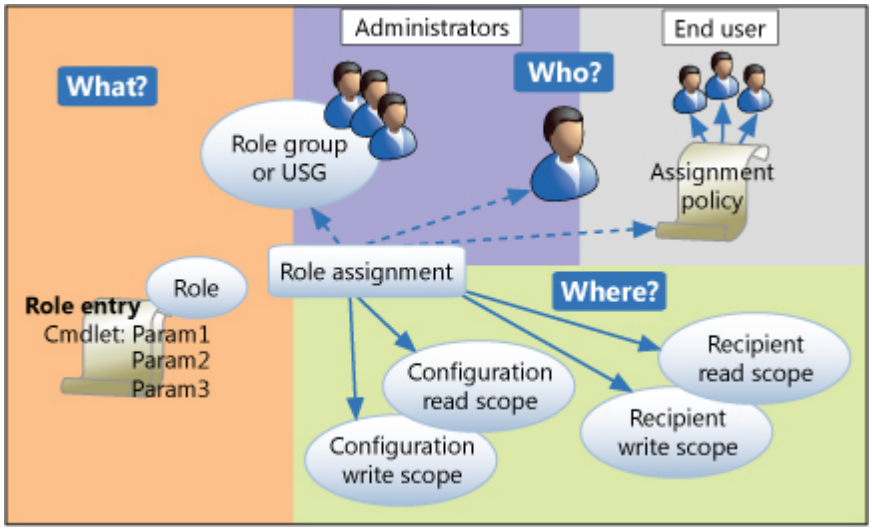
The following figure shows the components in RBAC and how they fit together:

- Role groups:
  - One or more administrators can be members of a role group. They can also be members of more than one role group.
  - The role group is assigned one or more role assignments. These link the role group with one or more administrative roles that define what tasks can be performed.
  - The role assignments can contain management scopes that define where the users of the role group can perform actions. The scopes determine where the users of the role group can modify configuration.
- Role assignment policies:
  - One or more users can be associated with a role assignment policy.
  - The role assignment policy is assigned one or more role assignments. These link the role assignment policy with one or more end-user roles. The end-user roles define what the user can configure on his or her mailbox.
  - The role assignments between role assignment policies and roles have built-in scopes that

restrict the scope of assignments to the user's own mailbox or distribution groups.

- Direct role assignment (advanced):
  - A role assignment can be created directly between a user or USG and one or more roles. The role defines what tasks the user or USG can perform.
  - The role assignments can contain management scopes that define where the user or USG can perform actions. The scopes determine where the user or USG can modify configuration.

RBAC overview



As shown in the preceding figure, many components in RBAC are related to each other. It's how each component is put together that defines the permissions applied to each administrator or user. The following examples provide some additional context about how role groups and role assignment policies are used in an organization.

Jane the Administrator

Jane is an administrator for the medium-size company, Contoso. She's responsible for managing the company's recipients in their Vancouver office. When the permissions model for Contoso was created, Jane was made a member of the Recipient Management - Vancouver custom role group. The Recipient Management - Vancouver custom role group most closely matches her job's duties, which include creating and removing recipients, such as mailboxes and contacts, managing distribution group membership and mailbox properties, and similar tasks.

In addition to the Recipient Management - Vancouver custom role group, Jane also needs a role assignment policy to manage her own mailbox's configuration settings. The organization administrators have decided that all users, except for senior management, receive the same permissions when they manage their own mailboxes. They can configure their voice mail, set up retention policies and change their address information. The default role assignment policy provided with Exchange 2013 now reflects these requirements.

Note:
You may have noticed that because Jane is a member of the Recipient Management - Vancouver custom role group, that should give her permissions to manage her own mailbox. This is true; however, the role group doesn't provide her all of the permissions necessary to

manage all of the features of her mailbox. The permissions needed to manage voice mail and retention policy settings aren't included in her role group. Those are provided only by the default role assignment policy assigned to her.

To allow for this, consider the role group, which provides Jane's administrative permissions over the recipients in Vancouver:

1. A custom role group called Recipient Management - Vancouver was created. When it was created, the following occurred:
   a. The role group was assigned all of the same management roles that are also assigned to the Recipient Management built-in role group. This gives users added to the Recipient Management - Vancouver custom role group the same permissions as those users added to the Recipient Management role group. However, the following steps limit where they can use those permissions.
   b. The Vancouver Recipients custom management scope was created, which matches only recipients who are located in Vancouver. This was done by creating a scope that filters on a user's city or other unique information.
   c. The role group was created with the Vancouver Recipients custom management scope. This means while administrators added to the Recipient Management - Vancouver custom role group have full recipient management permissions, they can only use those permissions against recipients based in Vancouver.

For more information about creating a custom role group, see Manage role groups.

2. Jane is then added as a member of the Recipient Management - Vancouver custom role group.

For more information about adding members to a role group, see Manage role group members.

To give Jane the ability to manage her own mailbox settings, a role assignment policy needs to be configured with the required permissions. The default role assignment policy is used to provide users with the permissions they need to configure their own mailbox. All end-user roles are removed from the default role assignment policy, except for: `MyBaseOptions`, `MyContactInformation`, `MyVoicemail`, and `MyRetentionPolicies`. `MyBaseOptions` is included because this management role provides the basic user functionality in Outlook Web App, such as Inbox rules, calendar configuration, and other tasks.

Nothing else needs to be done because Jane is already assigned the default role assignment policy. This means that the changes made to that role assignment policy are immediately applied to her mailbox, and any other mailboxes also assigned to the default role assignment policy.

For more information about customizing the default role assignment policy, see Manage role assignment policies.

Joe the Specialist

Joe works for Contoso, the same company that Jane works for. He's responsible for performing legal discovery, setting the retention policies, and configuring transport rules and journaling for the whole organization. As with Jane, when the permissions model for Contoso was created, Joe was

added to the role groups that match his job duties. The Records Management role group provides Joe with the permissions to configure retention policies, journaling, and transport rules. The Discovery Management role group provides him with the ability to perform mailbox searches.

As with Jane, Joe also needs permissions to manage his own mailbox. He is given the same permissions as Jane: He can set up his voice mail and retention policies, and change his address information.

To give Joe the permissions to perform his job duties, Joe is added to the Records Management and Discovery Management role groups. The role groups don't need to be changed in any way because they already provide him with the permissions he needs, and the management scopes applied to them encompass the entire organization.

For more information about adding a user to a role group, see Manage role group members.

Joe's mailbox is also assigned the same default role assignment policy that's applied to Jane's mailbox. This gives him all the permissions he needs to manage the features of his mailbox that he's allowed to manage.

Isabel the Vice President

Isabel is the Vice President of Marketing at Contoso. Isabel, as part of the senior leadership team of Contoso, is given more permissions than the average user. This includes the permissions she's provided to manage her mailbox, with one exception: Isabel isn't allowed to manage her own retention policies for legal compliance reasons. Isabel can configure her voice mail, change her contact information, change her profile information, create and manage her own distribution groups, and add or remove herself from existing distribution groups owned by others.

So, Isabel is given different permissions on her own mailbox. Most users at Contoso are assigned to the default role assignment policy. However, senior leadership is assigned to the Senior Leadership role assignment policy. The following is done to create the custom role assignment policy:

1. A custom role assignment policy called Senior Leadership is created. The role assignment policy is assigned the `MyBaseOptions`, `MyContactInformation`, `MyVoicemail`, `MyProfileInformation`, `MyDistributionGroupMembership`, and `MyDistributionGroups` roles. `MyBaseOptions` is included because this role provides the basic user functionality in Outlook Web App, such as Inbox rules, calendar configuration, and other tasks.
2. Isabel is then manually assigned the Senior Leadership role assignment policy.

Isabel's mailbox now has the permissions provided by the Senior Leadership role assignment policy. Any changes made to this role assignment policy are automatically applied to her mailbox, and any other mailboxes also assigned to the same role assignment policy.

For more information

Manage role assignment policies

Understanding management role groups

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-05*

A *management role group* is a universal security group (USG) used in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. A management role group simplifies the assignment of management roles to a group of users. All members of a role group are assigned the same set of roles. Role groups are assigned administrator and specialist roles that define major administrative tasks in Exchange 2013 such as organization management, recipient management, and other tasks. Role groups enable you to more easily assign a broader set of permissions to a group of administrators or specialist users.

Note:

This topic focuses on advanced RBAC functionality. If you want to manage basic Exchange 2013 permissions, such as using the Exchange admin center (EAC) to add and remove members to and from role groups, create and modify role groups, or create and modify role assignment policies, see Permissions.

Contents

Note:

If you want to assign permissions to users to manage their own mailbox or distribution groups, see Understanding management role assignment policies.

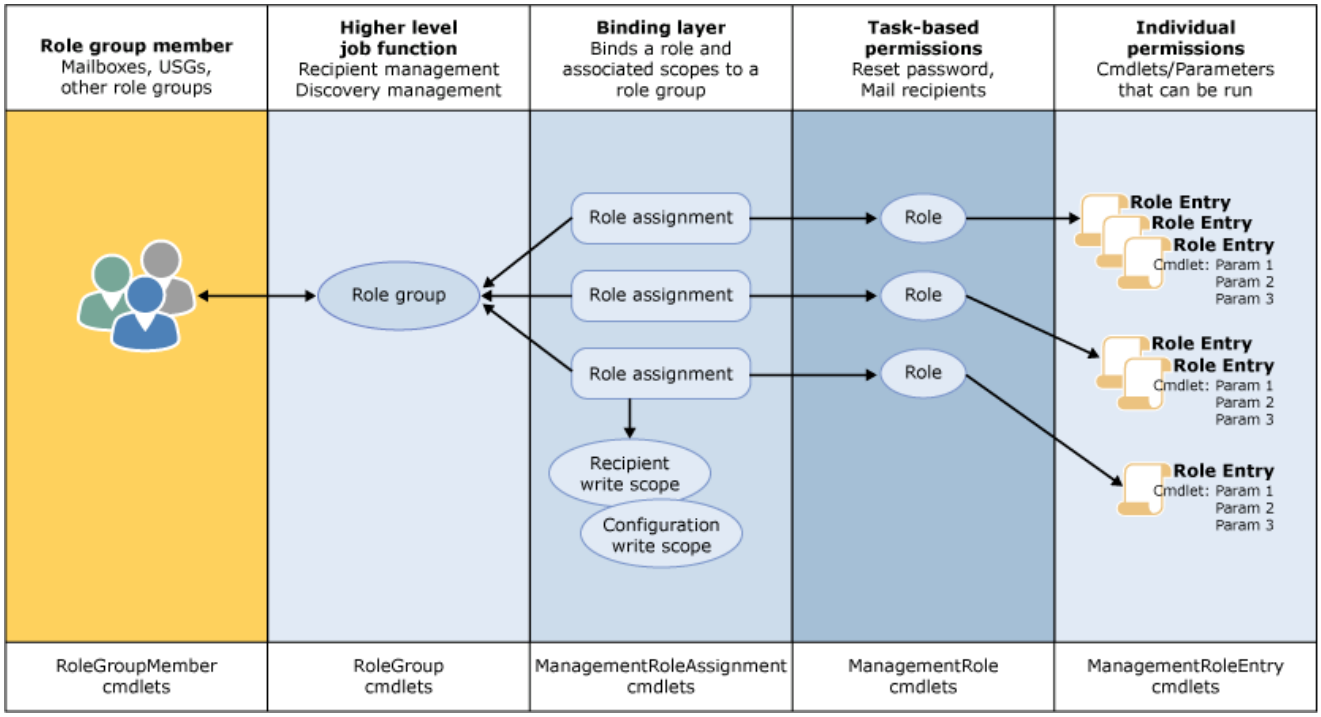
Role group layers

The following are the layers that make up the role group model:

- **Role group member** A *role group member* is a mailbox, universal security group (USG), or other role group that can be added as a member of a role group. When a mailbox, USG, or another role group is added as a member of a role group, the assignments that have been made between management roles and a role group are applied to the new member. This grants the new member all of the permissions provided by the management roles.
- **Management role group** The *management role group* is a special USG that contains mailboxes, USGs, and other role groups that are members of the role group. This is where you add and remove members, and it's also what management roles are assigned to. The combination of all the roles on a role group defines everything that members added to a role group can manage in the Exchange organization.
- **Management role assignment** A *management role assignment* links a management role and a role group. Assigning a management role to a role group grants members of the role group the ability to use the cmdlets and parameters defined in the management role. Role assignments can use management scopes to control where the assignment can be used. For more information, see Understanding management role assignments.
- **Management role scope** A *management role scope* is the scope of influence or impact on a role assignment. When a role is assigned with a scope to a role group, the management scope targets specifically what objects that assignment is allowed to manage. The assignment, and its scope, are then given to the members of the role group, which restricts what those members can manage. A scope can be made up of lists of servers or databases, organizational units, or filters on server, database or recipient objects. For more information, see Understanding management role scopes.
- **Management role** A *management role* is a container for a grouping of management role entries. Roles are used to define the specific tasks that can be performed by the members of a role group assigned the role. For more information, see Understanding management roles.
- **Management role entries** *Management role entries* are the individual entries on a management role that provide access to cmdlets, scripts, and other special permissions that enable access to perform a specific task. Most often, role entries consist of a single cmdlet or script and the parameters that can be accessed by the management role, and therefore the role group to which the role is assigned.

The following figure shows each of the role group layers in the preceding list and how each of the layers relates to the other.

Management role group layers



For more information about RBAC, see Understanding Role Based Access Control.

Return to top

Role group management

When you create a role group, you create the USG that holds the members of the role group, and you create the assignments between the role group and the management roles you specify. Optionally, you can also specify a management scope to apply to the role assignments, and you can add any mailboxes that you want to be members of the new role group.

After you create a role group, each layer becomes an independent object. The role group continues to be the central point at which all of the layers are joined together, however, each layer is managed individually. For example, to modify the management scope that you applied to the role group when it was created, you need to change the scope on each individual role assignment after the role group is created. The management of the role group model is performed using the cmdlets that manage the individual layers of the role group model.

The following table lists the role group layer and the procedural topics that you can use to manage each layer.

Role group management topics

Role group model layer	Management topic
Role group member	Manage role group members
Role group	Manage role groups
Management roles and assignments	Manage role groups

Management role entries	Add a role entry to a role Change a role entry Remove a role entry from a role **Note:** Changing the management role entries in management roles in a role group is an advanced task and is generally not required in most cases. Instead, you may be able to use a pre-existing management role that suits your requirements. For more information, see Built-in role groups.

Return to top

Built-in role groups

Built-in roles groups are roles shipped with Exchange 2013. They provide you with a set of role groups that you can use to provide varying levels of administrative permissions to groups of users. You can add or remove users to or from any built-in role group. You can also add or remove role assignments to or from most role groups. The only exceptions are the following:

- You can't remove any delegating role assignments from the Organization Management role group.
- You can't remove the Role Management role from the Organization Management role group.

The following table lists all the built-in role groups included with Exchange 2013. For more information about built-in role groups, see Built-in role groups.

Built-in role groups

Role group	Description
Organization Management	Administrators who are members of the Organization Management role group have administrative access to the entire Exchange 2013 organization and can perform almost any task against any Exchange 2013 object, with some exceptions. By default, members of this role group can't perform mailbox searches and management of unscoped top-level management roles.

View-only Organization Management	Administrators who are members of the View Only Organization Management role group can view the properties of any object in the Exchange organization.
Recipient Management	Administrators who are members of the Recipient Management role group have administrative access to create or modify Exchange 2013 recipients within the Exchange 2013 organization.
UM Management	Administrators who are members of the UM Management role group can manage features in the Exchange organization such as Unified Messaging (UM) service configuration, UM properties on mailboxes, UM prompts, and UM auto attendant configuration.
Discovery Management	Administrators or users who are members of the Discovery Management role group can perform searches of mailboxes in the Exchange organization for data that meets specific criteria and can also configure litigation holds on mailboxes.
Records Management	Users who are members of the Records Management role group can configure compliance features, such as retention policy tags, message classifications, transport rules, and more.
Server Management	Administrators who are members of this role group can configure server-specific configuration of transport , client access, and mailbox features such as database copies,

	certificates, transport queues and Send connectors, virtual directories, and client access protocols.
Help Desk	Users who are members of the Help Desk role group can perform limited recipient management of Exchange 2013 recipients.
Hygiene Management	Users who are members of the Hygiene Management role group can configure the anti-spam and anti-malware features of Exchange 2013. Third-party programs that integrate with Exchange 2013 can add service accounts to this role group to grant those programs access to the cmdlets required to retrieve and configure the Exchange configuration.
Compliance Management	Users who are members of the Compliance Management role group can configure and manage Exchange compliance configuration in accordance with their policies.
Public Folder Management	Administrators who are members of the Public Folder Management role group can manage public folders on servers running Exchange 2013.
Delegated Setup	Administrators who are members of the Delegated Setup role group can deploy servers running Exchange 2013 that have been previously provisioned by a member of the Organization Management role group.

Return to top

Linked role groups

Linked role groups are used in organizations that install Exchange 2013 in a dedicated resource forest and place users in other, trusted foreign forests. Linked role groups, as the name implies, create a link between a role group in the Exchange forest and a USG in a foreign forest. This is useful when the Active Directory Domain Services (AD DS) user accounts of the administrators you want to administer Exchange don't reside in the same resource forest as Exchange. Linked role groups can only be associated with one foreign USG. Additionally, you don't need to create a two-way trust between the Exchange forest and the foreign forest. The Exchange forest needs to trust the foreign forest but the foreign forest doesn't need to trust the Exchange forest.

For more information about permissions in multiple-forest topologies, see Understanding multiple-forest permissions.

A linked role group consists of two parts:

- **Linked role group** The linked role group is a container object that associates the foreign USG with the management role assignments assigned to the role group.
- **Foreign USG** The foreign USG contains the members that should be granted the permissions provided by the linked role group.

When you create a linked role group, you provide a domain controller in the foreign forest that contains the users you want to manage the Exchange forest and the USG that contains those users as members, the foreign USG name, and the credentials required to access the foreign forest. Exchange adds the security identifier (SID) of the foreign USG to the linked role group. Because the USG SID is the only identification of the foreign USG, we strongly recommend that you specify the foreign forest in the name of the role group if you have multiple foreign forests.

A linked role group doesn't contain any members. All of the members of that role group are managed using the foreign USG. This means you can't use the **Update-RoleGroupMember**, **Add-RoleGroupMember**, or **Remove-RoleGroupMember** cmdlets to add or remove role group members. When you add members to the foreign USG, they are given the permissions provided by the linked role group.

You can't change a standard role group, which contains its own members, to a linked role group and vice versa. If you want to change a role group from a standard role group to a linked role group, you must create a new linked role group and replicate the management role assignments that are present on the standard role group on the linked role group. This is also the case for built-in role groups because they're standard role groups. If you want to perform all of the management of your Exchange forest from a foreign forest, you need to create new linked role groups and add the management roles that exist on the built-in role groups to the new linked role groups. For more information about how to accomplish this, see Create linked role groups that mirror built-in role groups.

Return to top

Role group delegation

By default, members of the Organization Management role group can add and remove members to and from role groups. However, you might want to enable users who aren't members of the Organization Management role group to add and remove role group members. If so, you can use role group delegation.

Role group delegation is controlled by the **ManagedBy** property on each role group. The **ManagedBy** property contains a list of users who can add and remove members to and from that role group or change the configuration of a role group. The users aren't assigned any permissions given by the role group unless they're also members of the role group.

If the **ManagedBy** property is set on a role group, only those users who are listed as role group managers on that property can modify a role group or the membership of a role group by default. However, an optional parameter on cmdlets that modify role groups or role group membership can override that restriction. The *BypassSecurityGroupManagerCheck* switch can be used by users who are members of the Organization Management role or are assigned, either directly or indirectly, the Role Management management role. When this switch is used, the **ManagedBy** property is ignored and the user can modify the role group or role group membership.

If the **ManagedBy** property isn't set on a role group, only users who are members of the Organization Management role or are assigned, either directly or indirectly, the Role Management management role can modify a role group or role group membership.

Note:

Roles assigned to a role group may be assigned using delegating role assignments. With delegating role assignments, members of a role group that's assigned a delegated role can assign that role to another role group, assignment policy, user, or USG. Members of the role group can assign only that role and can't delegate the role group, unless they're also added to the **ManagedBy** property. For more information about delegated role assignments, see Understanding management role assignments.

For more information about how to manage role group delegation, see Manage role groups.

Return to top

Role group membership

When a user is made a member of a role group, the management roles assigned to the role group are assigned to the user. If a user is a member of multiple role groups, the management roles from each role group are aggregated and assigned to the user. Users, USGs, and other role groups can be members of role groups.

Only users who are members of the Organization Management or Role Management role groups and users who have been delegated the ability to add and remove users to or from role groups can manage role group membership.

For more information about how to manage role group membership, see Manage role group members.

Role group creation workflow

As mentioned previously, a role group is made up of several layers. To help you understand what happens when a role group is created, consider the following example, which creates a new role group.

```
New-RoleGroup -Name "Seattle Recipient Management" -Roles
"Mail Recipients", "Distribution Groups", "Move Mailboxes",
"UM Mailboxes" -CustomRecipientWriteScope "Seattle Users",
-ManagedBy "Brian", "David", "Katie" -Members "Ray",
"Jenn", "Maria", "Chris", "Maija", "Carter", "Jenny",
"Sam", "Lukas", "Isabel", "Katie"
```

When the preceding command is run, the following happens:

1. A new role group object, which is a special USG, called Seattle Recipient Management is created.
2. The mailboxes for Ray, Jenn, Maria, Chris, Maija, Carter, Jenny, Sam, Lukas, Isabel, and Katie are added as members of the role group. These users receive the permissions provided by this role group.
3. The users Brian and David are added to the **ManagedBy** property of the role group. These users can add and remove members to and from the role group but won't be given any permissions provided by the role group because they're not members. Katie is also added to the **ManagedBy** property of the role group. Because she's added to the **ManagedBy** property, and she's a member of the role group, she can add or remove members to and from the role group, and she also receives the permissions provided by the role group.
4. The following management role assignments are created. The role assignments assign each management role specified in the command to the role group. The management scope Seattle Users is added to each role assignment. The name of each role assignment is a combination of the management role being assigned and the role group name.
   - `Mail Recipients_Seattle Recipient Management`
   - `Distribution Groups_Seattle Recipient Management`
   - `Move Mailboxes_Seattle Recipient Management`
   - `UM Mailboxes_Seattle Recipient Management`

If you compare the results of this command to the Management role group layers figure earlier in this topic, you can see where each step correlates to the role group layers. You can then refer to the Management role group management topics shown in "Role group management" earlier in this topic to manage each role group layer.

Return to top

Understanding management role assignment policies

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-05*

A *management role assignment policy* is a collection of one or more end-user management roles that enables end users to manage their own Microsoft Exchange Server 2013 mailbox and distribution group configuration. Role assignment policies, which are part of the Role Based Access Control (RBAC) permissions model in Exchange 2013, enable you to control what specific mailbox and distribution group configuration settings your end users can modify. Different groups of users can have role assignment policies specialized to them.

Note:

This topic focuses on advanced RBAC functionality. If you want to manage basic Exchange 2013 permissions, such as using the Exchange admin center (EAC) to add and remove members to and from role groups, create and modify role groups, or create and modify role assignment policies, see Permissions.

For more information about RBAC, see Understanding Role Based Access Control.

Contents

Role assignment policy layers

The following list describes the layers that make up the role assignment policy model:

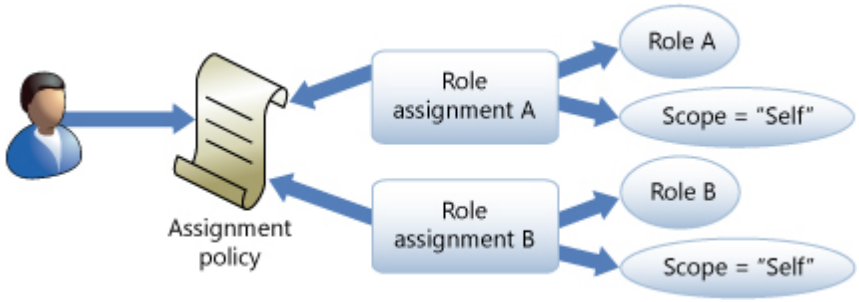
- **Mailbox** Mailboxes are assigned a single role assignment policy. When a mailbox is assigned a role assignment policy, the assignments between management roles and a role assignment policy are applied to the mailbox. This grants the mailbox all of the permissions provided by the management roles.
- **Management role assignment policy** The *management role assignment policy* is a special object in Exchange 2013. Users are associated with a role assignment policy when their mailboxes are created, or if you change the role assignment policy on a mailbox. This is also what you assign end-user management roles to. The combination of all the roles on a role assignment policy

defines everything that the user can manage on his or her mailbox or distribution groups.

- **Management role assignment** A *management role assignment* is the link between a management role and a role assignment policy. Assigning a management role to a role assignment policy grants the ability to use the cmdlets and parameters defined in the management role. When you create a role assignment between a role assignment policy and a management role, you can't specify any scope. The scope applied by the assignment is based on the management role and is either `Self` or `MyGAL`. For more information, see Understanding management role assignments.
- **Management role** A *management role* is a container for a grouping of management role entries. Roles are used to define the specific tasks that a user can do with his or her mailbox or distribution groups. A *management role entry* is a cmdlet, script, or special permission that enables each specific task in a management role to be performed. You can only use end-user management roles with role assignment policies. For more information, see Understanding management roles.
- **Management role entry** Management role entries are the individual entries on a management role that determine what cmdlets and parameters are available to the management role and the role group. Each role entry consists of a single cmdlet and the parameters that can be accessed by the management role.

The following figure shows each of the role assignment policy layers in the preceding list and how each of the layers relates to the other.

Management role assignment policy model



For more information about management roles, role assignments, and scopes, see Understanding Role Based Access Control.

Default and explicit role assignment policies

The following sections describe the two types of role assignment policies in Exchange 2013.

Default role assignment policy

A default role assignment policy is one assigned to a mailbox when the mailbox is created or moved to a server running Exchange 2013, and a role assignment policy wasn't provided using the *RoleAssignmentPolicy* parameter on the **New-Mailbox** or **Enable-Mailbox** cmdlets.

Exchange 2013 includes a default role assignment policy that provides end users with the

permissions most commonly used. You can change the default permissions on the default role assignment policy by adding or removing management roles to or from it.

If you want to replace the built-in default role assignment policy with your own default role assignment policy, you can use the **Set-RoleAssignmentPolicy** cmdlet to select a new default. When you do this, any new mailboxes are assigned the role assignment policy you specified by default if you don't explicitly specify a role assignment policy.

When you change the default role assignment policy, mailboxes assigned the default role assignment policy aren't automatically assigned the new default role assignment policy. If you want to update previously created mailboxes to use the role assignment policy you've set as default, you must use the **Set-Mailbox** cmdlet to do so.

Explicit Role Assignment Policy

An explicit role assignment policy is a policy that you assign to a mailbox manually using the *RoleAssignmentPolicy* parameter on the **New-Mailbox**, **Set-Mailbox**, or **Enable-Mailbox** cmdlets. When you assign an explicit role assignment policy, the new policy takes effect immediately and replaces the previously assigned explicit role assignment policy.

Using role assignment policies

Role assignment policies enable you to tailor permissions based on what business needs your users need to be able to configure. If the default role assignment policy meets your needs, you don't need to do any customization. However, if you have many different user groups with specialized needs, you can create role assignment policies for each of them.

The default role assignment policy you use should contain the permissions that apply to your broadest set of users. Then, create role assignment policies for each of your specialized user groups and tailor those role assignment policies to grant more or less restrictive permissions to them. When you organize your role assignment policies this way, you reduce complexity by only explicitly assigning role assignment policies to your specialized users while the majority of your users receive the more common permissions provided by the default role assignment policy.

A mailbox can have only one role assignment policy. All users, including administrators and specialist users, are assigned one role assignment policy. If you want a user to have a different set of permissions, you must assign that user's mailbox another role assignment policy with the required permissions.

Role assignment policy management

To add a new role assignment policy, you first create one and decide whether it should be the default role assignment policy. After you create a role assignment policy, you assign management roles to the role assignment policy, and then assign the role assignment policy to mailboxes. You

can later choose to add or remove management roles or choose a different role assignment policy to be the default.

The following table lists the role assignment policy layer and the procedural topics that you can use to manage each layer.

Role assignment policy management topics

Role assignment policy model layer	Management topics
Mailbox	Manage user mailboxes Change the assignment policy on a mailbox
Role assignment policy	Manage role assignment policies
Management roles and assignments	Manage role assignment policies
Management role entries	Add a role entry to a role Change a role entry Remove a role entry from a role **Note:** Changing the management role entries in management roles in a role assignment policy is an advanced task and is generally not required in most cases. You may, instead, be able to use a preexisting management role that suits your requirements. For more information, see Built-in role groups.

Understanding management roles

Exchange Server 2013 > Permissions > Understanding Role Based Access Control >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-11-19*

Management roles are part of the Role Based Access Control (RBAC) permissions model used in Microsoft Exchange Server 2013. Roles act as a logical grouping of cmdlets that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailboxes, transport rules, and recipients. Management roles can be further combined into larger groupings called management role groups and management role assignment policies, which

enable management of feature areas and recipient configuration. Role groups and role assignment policies assign permissions to administrators and end users, respectively. For more information about management role groups and management role assignment policies, see Understanding Role Based Access Control.

Note:

This topic focuses on advanced RBAC functionality. If you want to manage basic Exchange 2013 permissions, such as using the Exchange admin center (EAC) to add and remove members to and from role groups, create and modify role groups, or create and modify role assignment policies, see Permissions.

Contents

Management role scopes and management role assignments are important components for the operation of management roles. For more information about these components, see the following topics:

- Understanding management role scopes
- Understanding management role assignments

Looking for management tasks related to management roles? See Permissions.

Built-in management roles

Exchange 2013 provides many built-in management roles that you can use to administer your organization. Each role includes the cmdlets and parameters necessary for users to manage specific Exchange components. The following are examples of some built-in management roles:

- **Mail Recipients** Enables administrators to manage mailboxes, contacts, and mail users.
- **Transport Rules** Enables administrators or specialist users assigned the role to manage the transport rules feature.
- **Distribution Groups** Enables administrators or specialist users assigned the role to manage distribution groups and distribution group members.
- **MyPersonalInformation** Enables end users to modify their own home phone number and Web site address.

For a complete list of the management roles included with Exchange 2013, see Built-in management roles.

You can take the built-in roles provided with Exchange 2013 and combine them in any way to create a permissions model that works with your business. For example, if you want members of a role group to manage recipients and public folders, you assign both the Mail Recipients and Public Folders roles to the role group. Most often, you assign roles to role groups or role assignment policies. You can also assign management roles directly to users if you want to control permissions at a granular level. We recommend that you use role groups and role assignment policies rather than direct user role assignment to simplify your permissions model.

Note:
You can only assign end-user management roles to role assignment policies.

Built-in management roles can't be changed. You can, however, create management roles based on the built-in management roles, and then assign those new roles to role groups or role assignment policies. You can then change the new management roles to suit your needs. Doing so is an advanced task that you should rarely, if ever, need to do.

For more information about creating custom roles based on the built-in Exchange roles, see Custom Management Roles later in this topic.

You need to assign management roles for them to take effect. Most often, you assign management roles to role groups and role assignment policies. In certain circumstances, you might also assign roles directly to users, although this is an advanced task that you should rarely, if ever, need to do.

For more information about assigning management roles, see the following topics:

- Manage role groups
- Manage role assignment policies
- Add a role to a user or USG

For more information about management role assignments, see Understanding management role assignments.

Return to top

Unscoped top-level management roles

Unscoped top-level management roles are a special type of management role that enables you to grant access to custom scripts and non-Exchange cmdlets to users using RBAC. Regular management roles provide access only to Exchange cmdlets. If you need to provide access to non-Exchange cmdlets that run on an Exchange server, or if you need to publish a script that can be run by your users, you need to add them to an unscoped role. They're called a top-level role because if an unscoped role is created without deriving it from a parent role, it has no parent and is a peer of the built-in management roles provided with Exchange 2013. To indicate that you want to create an unscoped top-level role entry, you need to use the *UnscopedTopLevel* switch with the **New-ManagementRole** cmdlet.

Unscoped roles are named as such because, unlike regular management roles, they can't be scoped to a specific target. Unscoped roles are always organization wide. This means that someone assigned an unscoped role can modify any object in the Exchange 2013 organization. For this reason, care must be taken to make sure that scripts and cmdlets made available through an unscoped role are thoroughly tested so that you know what they will modify, and that you carefully assign unscoped roles.

Unscoped roles can be assigned to role assignees such as role groups, management roles, users, and universal security groups (USGs). They can't be assigned to management role assignment policies.

Although Exchange cmdlets can't be added as a management role entry on an unscoped role, they can be included in scripts added as role entries. This enables you to create custom scripts that perform Exchange tasks that you can then assign to users. A useful scenario is where a user must perform a highly privileged task that's normally outside his or her administrative level and where crafting a new management role or role group would be problematic. You can create a script that performs this specific function, add it to an unscoped role, and then assign the unscoped role to the user. The user can then perform only the specific function provided by the script.

The role entries that you add to an unscoped role must also be designated as an unscoped top-level role entry. For more information about unscoped top-level role entries, see Unscoped Top-Level Role Entries later in this topic.

The Organization Management role group doesn't, by default, have permissions to create or manage unscoped role groups. This is to prevent unscoped role groups from mistakenly being created or modified. The Organization Management role group can delegate the Unscoped Role Management management role to itself and other role assignees. For more information about how to create an unscoped top-level management role, see Create an unscoped role.

Return to top

Custom management roles

You can create custom management roles based on built-in Exchange roles when the built-in management roles don't match the needs of your users. When you create a custom management role, the new child role inherits all of the management role entries of the parent role. You can then choose which management role entries you want to keep in the custom management role and remove all of the entries you don't want.

Custom roles become children of the role used to create the new role. You can only use management role entries in the new child role that exist in the parent role. For more information, see the following sections later in this topic:

- Management Role Hierarchy
- Management Role Entries

Creating custom management roles requires multiple steps and is an advanced task that you should

rarely, if ever, need to perform. Before you create a custom management role, make sure one of the existing built-in management roles doesn't provide the permissions you need. For more information about the built-in management roles, or if you want to create custom management roles, see the following topics:

- Built-in management roles
- Advanced permissions

For more information about how to create a management role, see Create a role.

Return to top

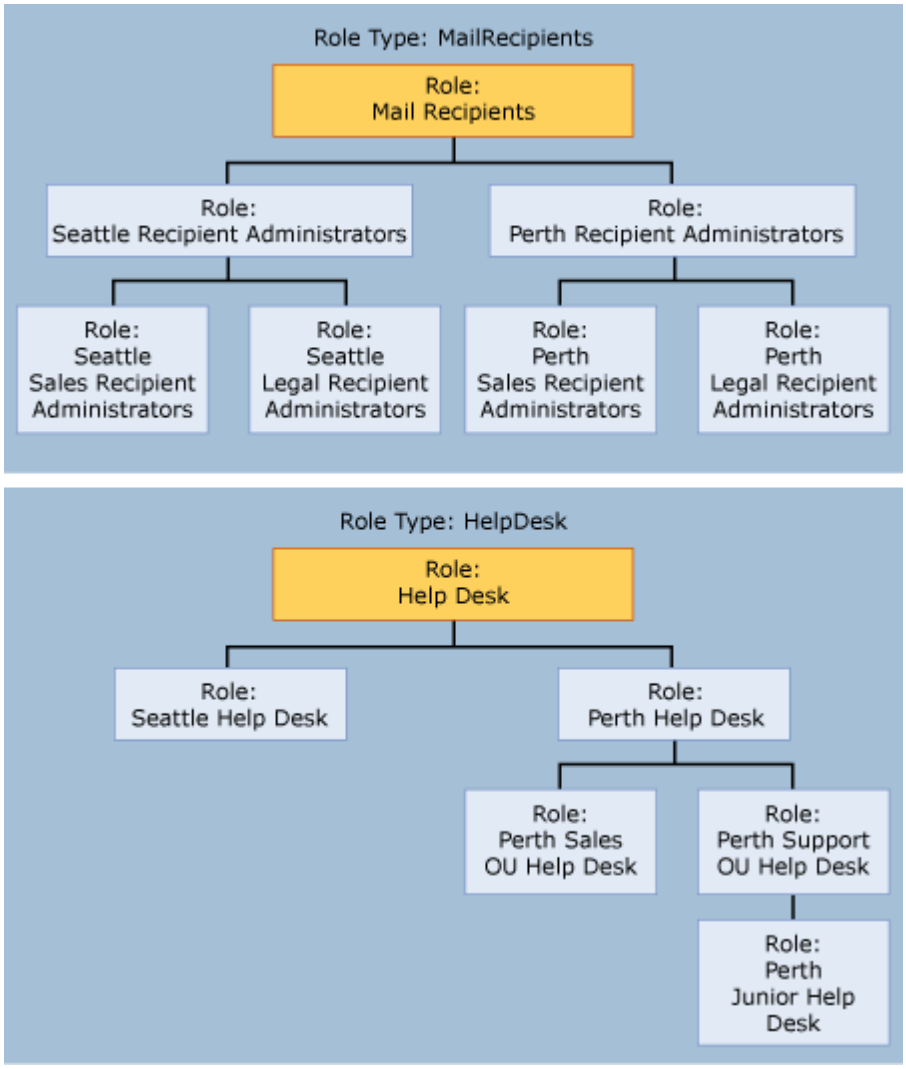
Management role hierarchy

Management roles exist in a parent and child hierarchy. At the top of the hierarchy are the built-in management roles provided in Exchange 2013 by default. When you create a role, a copy of a parent role is made. The new role is a child of the role you copied from. You can then customize the new role to suit the needs of the administrators or users you want to assign it to.

Customized roles can be used to create roles. When you create a role from an existing customized role, the existing customized role remains a child of its parent role, but also becomes the parent for the new role. Each time a role is copied, the new child role can contain only the role entries that exist in the immediate parent role.

Each management role is given a role type that can't be changed. The role type defines the base context of use for the role. The role type is copied from the parent role when the child role is created.

Management role hierarchy



The preceding figure illustrates the hierarchical relationship of several management roles. The Mail Recipients and Help Desk roles are built-in roles. All of the child roles derived from these roles inherit the role type of each built-in role. For example, all child roles derived either directly or indirectly from the Mail Recipients role inherit the `MailRecipients` role type.

The Seattle Recipient Administrators custom role is a child of the Mail Recipients built-in role but it's also the parent of the Seattle Sales Recipient Administrators custom role and the Seattle Legal Recipient Administrators custom role. The Seattle Recipient Administrators custom role contains only a subset of cmdlets that are available in the Mail Recipients role. The child roles of the Seattle Recipient Administrators custom role can only contain cmdlets that also exist in that role. For example, if a cmdlet exists in the Mail Recipients role, but the cmdlet doesn't exist in the Seattle Recipient Administrators custom role, the cmdlet can't be added to the Seattle Sales Recipient Administrators custom role.

All of the custom roles follow the same pattern as the roles discussed previously. For more information about how access to cmdlets is controlled on management roles, see Management Role Entries next in this topic.

Return to top

Management role entries

Every management role, whether it's a custom Exchange role or an unscoped role, must have at least one management role entry. An entry consists of a single cmdlet and its parameters, a script, or a special permission that you want to make available. If a cmdlet or script doesn't appear as an entry on a management role, that cmdlet or script isn't accessible via that role. Likewise, if a parameter doesn't exist in an entry, the parameter on that cmdlet or script isn't accessible via that role.

Exchange 2013 enables you to manage role entries based on built-in Exchange management top-level roles and role entries based on unscoped top-level management roles. Roles based on built-in Exchange top-level roles can only contain role entries that are Exchange 2013 cmdlets. To add custom scripts or non-Exchange cmdlets so that your users can use them, you need to add them as unscoped role entries to an unscoped top-level role. For more information about unscoped role entries, see Unscoped Top-Level Role Entries later in this topic.

All role entries, regardless of whether the role entry is an Exchange cmdlet-based role entry or an unscoped role entry, adhere to the same principles explained in the following sections.

For more information about managing role entries, see Management roles and role entries.

Parent and child management role relationship

As mentioned previously, a management role entry, including the cmdlet and its parameters, must exist in the immediate parent role to add the entry to the child role. For example, if the parent role doesn't have an entry for **New-Mailbox**, the child role can't be assigned that cmdlet. Additionally, if **Set-Mailbox** is on the parent role but the *Database* parameter has been removed from the entry, the *Database* parameter on the **Set-Mailbox** cmdlet can't be added to the entry on the child role.

Because you can't add management role entries to child roles if the entries don't appear in parent roles, and because the role is based on a specific role type, you must carefully choose the parent role to copy when you want to create a customized role.

Return to top

Management role entry names

Management role entry names are a combination of the management role that they're associated with, and the name of the cmdlet or script. The role name and the cmdlet or script are separated by a backslash character (\). For example, the role entry name for the **Set-Mailbox** cmdlet on the Mail Recipients role is `Mail Recipients\Set-Mailbox`. If the name of a role entry contains spaces, enclose the entire name in quotation marks (").

The wildcard character (*) can be used in the role entry name to return all of the role entries that match the input you provide. The wildcard character can be used on either side of the backslash character. The following table contains a few variations on how you can use the wildcard character in a role entry name.

Management role entry name with wildcard characters

Example	Description
`*\*`	Returns a list of all role entries for all roles.
`*\Set-Mailbox`	Returns a list of all role entries that contain the **Set-Mailbox** cmdlet.
`Mail Recipients\*`	Returns a list of all role entries on the Mail Recipients role.
`Mail Recipients\*Mailbox`	Returns a list of all role entries on the Mail Recipients role that contain cmdlets that end in Mailbox.
`My*\*Group*`	Returns a list of all role entries that contain the string Group in the cmdlet name for all roles that begin with My.

Unscoped top-level role entries

Unscoped top-level role entries are used with unscoped top-level management roles to create roles based on custom scripts or non-Exchange cmdlets. Each unscoped role entry is associated with a single custom script or a non-Exchange cmdlet. To indicate that you want to create an unscoped role entry on an unscoped role, you need to specify the *UnscopedTopLevel* parameter on the **Add-ManagementRoleEntry** cmdlet.

When you add the unscoped role entry, you need to specify all of the parameters that can be used with the script or non-Exchange cmdlet. Exchange attempts to verify the parameters that you provide when you add the role entry. Only the parameters that you add to the role entry when it's created will be available to the users assigned to the unscoped role. If you add parameters to the script or non-Exchange cmdlet, or if a parameter is renamed, you must update the role entry manually. Exchange doesn't check whether existing parameters on an unscoped role entry have changed. If a parameter on a role entry changes in a script and you try to use that parameter, the command fails.

Scripts that you add to an unscoped role entry must reside in the Exchange 2013 scripts directory on every server where administrators and users connect using the Exchange Management Shell. If you try to add an unscoped role entry based on a script that doesn't exist in the Exchange 2013 scripts directory on the server you're using to add the role entry, an error occurs. The default installation location of the Exchange 2013 scripts directory is C:\Program Files\Microsoft\Exchange Server\V15\RemoteScripts.

Non-Exchange cmdlets that you add to an unscoped role entry must be installed on every Exchange 2013 server where administrators and users connect using the Shell and want to use the cmdlets. If you try to add an unscoped role entry based on a non-Exchange cmdlet that isn't installed on the Exchange 2013 server you're using to add the role entry, an error occurs. When you add a non-Exchange cmdlet, you must specify the Windows PowerShell snap-in name that contains the non-Exchange cmdlet.

For more information about how to add an unscoped management role entry, see Add a role entry to a role.

Return to top

Management role types

Management role types are the foundation of all management roles. Types define the implicit scopes defined on all management roles of a specified role type and also act as a logical grouping of related roles. All management roles derived from the parent built-in management role have the same role type. Refer to the Management role hierarchy figure earlier in this topic for an illustration of this relationship. Management role types also represent the maximum set of cmdlets and their parameters that can be added to a role associated with a role type.

Management role types are split into the following categories:

- **Administrative or specialist** Roles associated with an administrative or specialist role types have a broader scope of impact in the Exchange organization. Roles of this role type enable tasks such as server or recipient management, organization configuration, compliance administration, auditing, and more.
- **User-focused** Roles associated with a user-focused role type have a scope of impact closely tied with an individual user. Roles of this role type enable tasks such as user profile configuration and self management, management of user-owned distribution groups, and more.

The names of roles associated with user-focused role types and user-focused role type names begin with My.

- **Specialty** Roles associated with specialty role types enable tasks that aren't administrative or user-focused role types. Roles of this role type enable tasks such as application impersonation and the use of non-Exchange cmdlets or scripts.

The following table lists all of the administrative management role types in Exchange 2013 and whether the configuration that's permitted by the role type is applied across the whole Exchange organization or only to an individual server. For more information about each of the management roles associated with these role types, including a description of each role, who may benefit from being assigned the role, and other information, see Built-in management roles.

Administrative role types

Management role type	Built-in management role	Description	Organization or server

`ActiveDirectoryPermissions`	Active Directory Permissions role	This role type is associated with roles that enable administrators to configure Active Directory permissions in an organization. Some features that use Active Directory permissions or an access control list (ACL) include transport Receive and Send connectors, and Send As and send on behalf permissions for mailboxes. **Note:** Permissions set directly on Active Directory objects may not be enforced through RBAC.	Organization
`AddressLists`	Address Lists role	This role type is associated with roles that enable administrators to manage address lists, the global address list (GAL), and offline address lists in an organization.	Organization
`ApplicationImpersonation`	ApplicationImpersonation role	This role type is associated with roles that enable applications to impersonate users in an organization to perform tasks on behalf of the	Organization

		user.	
ArchiveApplication	ArchiveApplication role	This role type is associated with roles that enable partner applications to archive items in user mailboxes in an organization.	Organization
AuditLogs	Audit Logs role	This role type is associated with roles that enable administrators to manage the administrator audit logging configuration in an organization.	Organization
CmdletExtensionAgents	Cmdlet Extension Agents role	This role type is associated with roles that enable administrators to manage cmdlet extension agents in an organization.	Organization
DataLossPrevention	Data Loss Prevention role	This role type is associated with roles that create and manage data loss prevention (DLP) policies and the rules within them in an organization.	Organization
DatabaseAvailabilityGroups	Database Availability Groups role	This role type is associated with roles that enable administrators to manage database	Organization

		availability groups (DAGs) in an organization. Administrators assigned this role either directly or indirectly are the highest level administrators responsible for the high availability configuration in an organization.	
DatabaseCopies	Database Copies role	This role type is associated with roles that enable administrators to manage database copies on individual servers.	Server
Databases	Databases role	This role type is associated with roles that enable administrators to create, manage, mount, and dismount mailbox databases on individual servers.	Server
DisasterRecovery	Disaster Recovery role	This role type is associated with roles that enable administrators to restore mailboxes and mailbox databases, create mailbox databases, and perform datacenter switchovers and switchbacks for database availability groups in an	Organization

		organization.	
DistributionGroups	Distribution Groups role	This role type is associated with roles that enable administrators to create and manage distribution groups and distribution group members in an organization.	Organization
EdgeSubscriptions	Edge Subscriptions role	This role type is associated with roles that enable administrators to manage edge synchronization and subscription configuration between Exchange 2010 Edge Transport servers and Exchange 2013 Mailbox servers in an organization.	Organization
EmailAddressPolicies	E-Mail Address Policies role	This role type is associated with roles that enable administrators to manage email address policies in an organization.	Organization
ExchangeConnectors	Exchange Connectors role	This role type is associated with roles that enable administrators to create, modify, view, and remove delivery agent	Organization

		connectors.	
ExchangeServerCertificates	Exchange Server Certificates role	This role type is associated with roles that enable administrators to create, import, export, and manage Exchange server certificates on individual servers.	Server
ExchangeServers	Exchange Servers role	This role type is associated with roles that enable administrators to manage Exchange server configuration on individual servers.	Server
ExchangeVirtualDirectories	Exchange Virtual Directories role	This role type is associated with roles that enable administrators to manage Outlook Web App, Microsoft ActiveSync, offline address book (OAB), Autodiscover, Windows PowerShell, and Exchange Administration Center virtual directories on individual servers.	Server
FederatedSharing	Federated Sharing role	This role type is associated with roles that enable administrators to manage cross-forest and cross-organization sharing in an	Organization

		organization.	
`InformationRightsManagement`	Information Rights Management role	This role type is associated with roles that enable administrators to manage the Information Rights Management (IRM) features of Exchange in an organization.	Organization
`Journaling`	Journaling role	This role type is associated with roles that enable administrators to manage journaling configuration in an organization.	Organization
`LegalHold`	Legal Hold role	This role type is associated with roles that enable administrators to configure whether data within a mailbox should be retained for litigation purposes in an organization.	Organization
`MailboxImportExport`	Mailbox Import Export role	This role type is associated with roles that enable administrators to import and export mailbox content and to purge unwanted content from a mailbox.	Organization
`MailboxSearch`	Mailbox Search role	This role type is	Organization

		associated with roles that enable administrators to search the content of one or more mailboxes in an organization.	
`MailboxSearchApplication`	MailboxSearchApplication role	This role type is associated with roles that enable partner applications to set and view the legal hold status of a mailbox in an organization.	Organization
`MailEnabledPublicFolders`	Mail Enabled Public Folders role	This role type is associated with roles that enable administrators to configure whether individual public folders are mail-enabled or mail-disabled in an organization. This role type enables you to manage the email properties of public folders only. It doesn't enable you to manage properties of public folders that aren't email properties. To manage properties of public folders that aren't email properties, you need to be assigned a role associated	Organization

		with the `PublicFolders` role type.	
`MailRecipientCreation`	Mail Recipient Creation role	This role type is associated with roles that enable administrators to create mailboxes, mail users, mail contacts, distribution groups, and dynamic distribution groups in an organization. Roles associated with this role type can be combined with roles associated with the `MailRecipients` role type to enable the creation and management of recipients. This role type doesn't enable you to mail-enable public folders. To mail-enable public folders, you must be assigned a role associated with the `MailEnabledPublicFolders` role type. If your organization maintains a split permissions model where recipient creation is performed by a different group from the group that performs recipient	Organization

		management, assign the `MailRecipientCreation` role to the group that performs recipient creation, and the `MailRecipients` role to the group that performs recipient management.	
`MailRecipients`	Mail Recipients role	This role type is associated with roles that enable administrators to manage existing mailboxes, mail users, mail contacts, distribution groups, and dynamic distribution groups in an organization. Roles associated with this role type can't create these recipients but can be combined with roles associated with the `MailRecipientCreation` role type to enable the creation and management of recipients. This role type doesn't enable you to manage mail-enabled public folders or distribution groups. To manage mail-enabled public folders, you must be assigned a	Organization

		role associated with the `MailEnabledPublicFolders` role type. To manage distribution groups, you must be assigned a role associated with the `DistributionGroups` role type. If your organization maintains a split permissions model where recipient creation is performed by a different group from the group that performs recipient management, assign the `MailRecipientCreation` role to the group that performs recipient creation, and the `MailRecipients` role to the group that performs recipient management.	
`MailTips`	Mail Tips role	This role type is associated with roles that enable administrators to manage MailTips in an organization.	Organization
`MessageTracking`	Message Tracking role	This role type is associated with roles that enable administrators to track messages in an	Organization

		organization.	
`Migration`	Migration role	This role type is associated with roles that enable administrators to migrate mailboxes and mailbox content into or out of a server.	Server
`Monitoring`	Monitoring role	This role type is associated with roles that enable administrators to monitor the Microsoft Exchange services and component availability in an organization. In addition to administrators, roles associated with this role type can be used with the service account used by monitoring applications to collect information about the state of Exchange servers.	Organization
`MoveMailboxes`	Move Mailboxes role	This role type is associated with roles that enable administrators to move mailboxes between servers in an organization and between servers in the local organization and another organization.	Organization
`OfficeExtensionAppl ication`	OfficeExtensionApplic	This role type is	Organization

	ation role	associated with roles that enable Microsoft Office extension applications to access user mailboxes in an organization.	
OrgCustomApps	Org Custom Apps role	This role type is associated with roles that enable administrators to view and modify their organization's custom apps in an organization.	Organization
OrgMarketplaceApps	Org Marketplace Apps role	This role type is associated with roles that enable administrators to view and modify their organization's marketplace apps in an organization.	Organization
OrganizationClientAccess	Organization Client Access role	This role type is associated with roles that enable administrators to manage Client Access server settings in an organization.	Organization
OrganizationConfiguration	Organization Configuration role	This role type is associated with roles that enable administrators to manage organization-wide settings in an organization. Organization	Organization

		configuration that can be controlled with this role type include the following and more: • Whether MailTips are enabled or disabled for the organization. • The URL for the managed folder home page. • The Microsoft Exchange recipient SMTP address and alternate email addresses. • The resource mailbox property schema configuration. • The Help URLs for the Exchange Administration Center and Outlook Web App. This role type doesn't include the permissions included in the `OrganizationClientAccess` or `OrganizationTransportSettings` role types.	
`OrganizationTransportSettings`	Organization Transport Settings role	This role type is associated with roles that enable administrators to manage organization-wide transport settings,	Organization

		such as system messages, Active Directory site configuration, and other organization-wide transport settings in an organization. This role doesn't enable you to create or manage transport Receive or Send connectors, queues, hygiene, agents, remote and accepted domains, or rules. To create or manage each of the transport features, you must be assigned roles associated with the following role types: • **Receive connectors** `ReceiveConnectors` • **Send connectors** `SendConnectors` • **Transport queues** `TransportQueues` • **Transport hygiene** `TransportHygiene` • **Transport agents** `TransportAgents` • **Remote and accepted domains** `RemoteAndAcceptedDomains` • **Transport rules** `TransportRules`	
`POP3AndIMAP4Protocols`	POP3 and IMAP4 Protocols role	This role type is associated with roles that	Server

		enable administrators to manage POP3 and IMAP4 configuration, such as authentication and connection settings, on individual servers.	
`PublicFolders`	Public Folders role	This role type is associated with roles that enable administrators to manage public folders in an organization. This role type doesn't enable you to manage whether public folders are mail-enabled. To mail-enable or disable a public folder, you must be assigned a role associated with the `MailEnabledPublicFolders` role type.	Organization
`ReceiveConnectors`	Receive Connectors role	This role type is associated with roles that enable administrators to manage transport Receive connector configuration, such as size limits on an individual server.	Server
`RecipientPolicies`	Recipient Policies role	This role type is associated with roles that enable administrators to manage recipient policies,	Organization

		such as provisioning and mobile device policies, in an organization.	
`RemoteAndAcceptedDomains`	Remote and Accepted Domains role	This role type is associated with roles that enable administrators to manage remote and accepted domains in an organization.	Organization
`ResetPassword`	Reset Password role	This role type is associated with roles that enable users to reset their own passwords and administrators to reset users' passwords in an organization.	Organization
`RetentionManagement`	Retention Management role	This role type is associated with roles that enable administrators to manage retention policies in an organization.	Organization
`RoleManagement`	Role Management role	This role type is associated with roles that enable administrators to manage management role groups, role assignment policies, management roles, role entries, assignments, and scopes in an organization. Users assigned roles	Organization

		associated with this role type can override the **role group managed by** property, configure any role group, and add or remove members to or from any role group.	
`SecurityGroupCreationAndMembership`	Security Group Creation and Membership role	This role type is associated with roles that enable administrators to create and manage USGs and their memberships in an organization. If your organization maintains a split permissions model where USG creation and management is performed by a different group from the group that manages Exchange servers, assign roles associated with this role type to that group.	Organization
`SendConnectors`	Send Connectors role	This role type is associated with roles that enable administrators to manage transport Send connectors in an organization.	Organization
`SupportDiagnostics`	Support Diagnostics	This role type is	Organization

	role	associated with roles that enable administrators to perform advanced diagnostics under the direction of Microsoft support services in an organization. **Caution:** Roles associated with this role type grant permissions to cmdlets and scripts that should only be used under the direction of Microsoft Customer Service and Support.	
`TeamMailboxes`	Team Mailboxes role	This role type is associated with roles that enable administrators to define one or more site mailbox provisioning policies and manage site mailboxes in an organization. Administrators assigned roles associated with this role type can manage site mailboxes they don't own.	Organization
`TeamMailboxLifecycleApplication`	TeamMailboxLifecycle Application role	This role type is associated with roles that enable partner applications to update site mailbox lifecycle states in an organization.	Organization

TransportAgents	Transport Agents role	This role type is associated with roles that enable administrators to manage transport agents in an organization.	Organization
TransportHygiene	Transport Hygiene role	This role type is associated with roles that enable administrators to manage anti-spam and anti-malware features in an organization.	Organization
TransportQueues	Transport Queues role	This role type is associated with roles that enable administrators to manage transport queues on an individual server.	Server
TransportRules	Transport Rules role	This role type is associated with roles that enable administrators to manage transport rules in an organization.	Organization
UMMailboxes	UM Mailboxes role	This role type is associated with roles that enable administrators to manage the Unified Messaging (UM) configuration of mailboxes and other recipients in an organization.	Organization

`UMPrompts`	UM Prompts role	This role type is associated with roles that enable administrators to create and manage custom UM voice prompts in an organization.	Organization
`UnifiedMessaging`	Unified Messaging role	This role type is associated with roles that enable administrators to manage Unified Messaging services in an organization. This role doesn't enable you to manage UM-specific mailbox configuration or UM prompts. To manage UM-specific mailbox configuration, use roles associated with the `UMMailboxes` role type. To manage UM prompts, use the roles associated with the `UMPrompts` role type.	Organization
`UnScopedRoleManagement`	Unscoped Role Management role	This role type is associated with roles that enable administrators to create and manage unscoped top-level management roles in an organization.	Organization

UserOptions	User Options role	This role type is associated with roles that enable administrators to view the Outlook Web App options of a user in an organization. Roles associated with this role type can be used to help a user with diagnosing problems with his or her configuration.	Organization
UserApplication	UserApplication role	This role type is associated with roles that enable partner applications to act on behalf of end users in an organization.	Organization
ViewOnlyAuditLogs	View-Only Audit Logs role	This role type is associated with roles that enable administrators to search the administrator audit log in an organization.	Organization
ViewOnlyConfiguration	View-Only Configuration role	This role type is associated with roles that enable administrators to view all of the non-recipient Exchange configuration settings in an organization. Examples of configuration that are viewable are server	Organization

		configuration, transport configuration, database configuration, and organization-wide configuration. Roles associated with this role type can be combined with roles associated with the `ViewOnlyRecipients` role type to create a role that can view every object in an organization.	
`ViewOnlyRecipients`	View-Only Recipients role	This role type is associated with roles that enable administrators to view the configuration of recipients, such as mailboxes, mail users, mail contacts, distribution groups, and dynamic distribution groups. Roles associated with this role type can be combined with roles associated with the `ViewOnlyConfiguration` role type to create a role that can view every object in the organization.	Organization
`WorkloadManagement`	WorkloadManagement role	This role type is associated with roles that enable administrators to	Organization

		manage workload management policies in an organization. Administrators can configure resource health definitions, workload classifications, and enable or disable workload management.	

The following table lists all of the user-focused management role types and their associated built-in management roles in Exchange 2013.

User-focused role types

Management role type	Built-in user-focused roles	Description
`MyBaseOptions`	MyBaseOptions role	This role type is associated with roles that enable individual users to view and modify the basic configuration of their own mailbox and associated settings.
`MyContactInformation`	MyAddressInformation role MyContactInformation role MyMobileInformation role MyPersonalInformation role	This role type is associated with roles that enable individual users to modify their contact information. This information includes their address and phone numbers.
MyCustomApps	My Custom Apps role	This role type is associated with roles that enable individual users to view and modify their custom apps.
MyDiagnostics	MyDiagnostics role	This role type is associated with roles that enable individual

		users to perform basic diagnostics on their mailbox, such as retrieving calendar diagnostic information.
`MyDistributionGroupMembership`	MyDistributionGroupMembership role	This role type is associated with roles that enable individual users to view and modify their membership in distribution groups in an organization, provided that those distribution groups allow manipulation of group membership.
`MyDistributionGroups`	MyDistributionGroups role	This role type is associated with roles that enable individual users to create, modify, and view distribution groups and modify, view, remove, and add members to distribution groups they own.
`MyProfileInformation`	MyDisplayName role MyName role MyProfileInformation role	This role type is associated with roles that enable individual users to modify their name.
MyMarketplaceApps	My Marketplace Apps role	This role type is associated with roles that enable individual users to view and modify their marketplace apps.
`MyRetentionPolicies`	MyRetentionPolicies role	This role type is associated with roles that enable individual users to view their retention

		tags and view and modify their retention tag settings and defaults.
MyTeamMailboxes	MyTeamMailboxes role	This role type is associated with roles that enable individual users to create site mailboxes and connect them to Microsoft SharePoint sites.
`MyTextMessaging`	MyTextMessaging role	This role type is associated with roles that enable individual users to create, view, and modify their text messaging settings.
`MyVoiceMail`	MyVoiceMail role	This role type is associated with roles that enable individual users to view and modify their voice mail settings.

Return to top

For more information

New-ManagementRole

New-ManagementRoleAssignment

Set-ManagementRoleAssignment

New-ManagementScope

Set-ManagementScope

New-ManagementRoleAssignment

Set-ManagementRoleAssignment

Understanding management role

scopes

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-15*

Management role scopes enable you to define the specific scope of impact or influence of a management role when a management role assignment is created. When you apply a scope, the role assignee assigned to the role can only modify the objects contained within that scope. A role assignee can be a management role group, management role, management role assignment policy, user, or universal security group (USG). For more information about management roles, see Understanding Role Based Access Control.

Note:

This topic focuses on advanced RBAC functionality. If you want to manage basic Exchange 2013 permissions, such as using the Exchange admin center (EAC) to add and remove members to and from role groups, create and modify role groups, or create and modify role assignment policies, see Permissions.

Every management role, whether it's a built-in role or a custom role, has management scopes. Management scopes can be either of the following:

- **Regular** A *regular scope* isn't exclusive. It determines where, in Active Directory, objects can be viewed or modified by users assigned the management role. In general, a management role indicates what you can create or modify, and a management role scope indicates where you can create or modify. Regular scopes can be either implicit or explicit scopes, both of which are discussed later in this topic.
- **Exclusive** An *exclusive scope* behaves almost the same as a regular scope. The key difference is that it enables you to deny users access to objects contained within the exclusive scope if those users aren't assigned a role associated with the exclusive scope. All exclusive scopes are explicit scopes, which are discussed later in this topic.

For more information about exclusive scopes, see Understanding exclusive scopes.

Scopes can be inherited from the management role, specified as a predefined relative scope on a management role assignment, or created using custom filters and added to a management role assignment. Scopes inherited from management roles are called *implicit scopes* while predefined and custom scopes are called *explicit scopes*. The following sections describe each type of scope:

- Implicit Scopes
- Explicit Scopes
- Predefined Relative Scopes
- Custom Scopes
  - Recipient Filter Scopes
  - Configuration Scopes

Each role can have the following types of scopes:

- **Recipient read scope** The implicit recipient read scope determines what recipient objects the user assigned the management role is allowed to read from Active Directory.
- **Recipient write scope** The implicit recipient write scope determines what recipient objects the user assigned the management role is allowed to modify in Active Directory.
- **Configuration read scope** The implicit configuration read scope determines what configuration objects the user assigned the management role is allowed to read from Active Directory.
- **Configuration write scope** The implicit configuration write scope determines what organizational, database, and server objects the user assigned the management role is allowed to modify in Active Directory.

Recipient objects include mailboxes, distribution groups, mail enabled users, and other objects. Configuration objects include servers running Microsoft Exchange Server 2013, and databases located on servers running Exchange. Each type of scope can be either an implicit scope or explicit scope.

Implicit scopes

Implicit scopes are the default scopes that apply to a management role type. Because implicit scopes are associated with a management role type, all of the parent and child management roles with the same role type also have the same implicit scopes. Implicit scopes apply to both built-in management roles and also to custom management roles. For more information about management roles and management role types, see Understanding management roles.

The following tables list all of the implicit scopes that can be defined on management roles.

Implicit scopes defined on management roles

Implicit scopes	Description
`Organization`	If `Organization` is present in the role's recipient write scope, the role can create or modify recipient objects across the Exchange organization. If `Organization` is present in the role's recipient read scope, roles can view any recipient object across the Exchange organization. This scope is used only with recipient read and write scopes.
`MyGAL`	If `MyGAL` is present in the role's recipient write scope, the role can view the properties of any

	recipient within the current user's global address list (GAL). If `MyGAL` is present in the role's recipient read scope, the role can view the properties of any recipient within the current GAL. This scope is used only with recipient read scopes.
`Self`	If `Self` is present in the role's recipient write scope, the role can modify only the properties of the current user's mailbox. If `Self` is present in the role's recipient read scope, the role can view only the properties of the current user's mailbox. This scope is used only with recipient read and write scopes.
`MyDistributionGroups`	If `MyDistributionGroups` is present in the role's recipient write scope, the role can create or modify distribution list objects owned by the current user. If `MyDistributionGroups` is present in the role's recipient read scope, the role can view distribution list objects owned by the current user. This scope is used only with recipient read and write scopes.
`OrganizationConfig`	If `OrganizationConfig` is present in the role's configuration write scope, the role can create or modify any server or database configuration object across the Exchange organization. If `OrganizationConfig` is present in the role's

	configuration read scope, the role can view any server or database configuration object across the Exchange organization. This scope is used only with configuration read and write scopes.
None	If None is in a scope, that scope isn't available to the role. For example, a role that has None in the recipient write scope can't modify recipient objects in the Exchange organization.

If a role is assigned to a role assignee and no predefined or custom scopes are specified, the implicit scopes defined on the role are used to control the recipient or organization objects the user can view or modify.

The implicit write scope of a role is always equal to, or less than, the implicit read scope. This means that a role can never modify objects that can't be seen by the scope.

You can't change the implicit scopes defined on management roles. You can, however, override the implicit write scope and configuration scope on a management role. When a predefined relative scope or custom scope is used on a role assignment, the implicit write scope of the role is overridden, and the new scope takes precedence. The implicit read scope of a role can't be overridden and always applies. For more information, see Predefined Relative Scopes and Custom Scopes.

Expand the following table to see a list of all the built-in management roles and their implicit scopes. For more information about each built-in role, see Built-in management roles.

Built-in management role implicit scopes

Management role	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Active Directory Permissions	Organization	Organization	OrganizationConfig	OrganizationConfig
Address Lists	Organization	Organization	OrganizationConfig	OrganizationConfig
ApplicationImpersonation	Organization	Organization	None	None
ArchiveApplication	Organization	Organization	OrganizationConfig	OrganizationConfig
Audit Logs	Organization	Organization	OrganizationConfig	OrganizationConfig
Cmdlet Extension Agents	Organization	Organization	OrganizationConfig	OrganizationConfig

Data Loss Prevention	Organization	Organization	OrganizationConfig	OrganizationConfig
Database Availability Groups	Organization	Organization	OrganizationConfig	OrganizationConfig
Database Copies	Organization	Organization	OrganizationConfig	OrganizationConfig
Databases	Organization	Organization	OrganizationConfig	OrganizationConfig
Disaster Recovery	Organization	Organization	OrganizationConfig	OrganizationConfig
Distribution Groups	Organization	Organization	OrganizationConfig	OrganizationConfig
Edge Subscriptions	Organization	Organization	OrganizationConfig	OrganizationConfig
E-Mail Address Policies	Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Connectors	Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Server Certificates	Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Servers	Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Virtual Directories	Organization	Organization	OrganizationConfig	OrganizationConfig
Federated Sharing	Organization	Organization	OrganizationConfig	OrganizationConfig
Information Rights Management	Organization	Organization	OrganizationConfig	OrganizationConfig
Journaling	Organization	Organization	OrganizationConfig	OrganizationConfig
Legal Hold	Organization	Organization	OrganizationConfig	None
LegalHoldApplication	Organization	Organization	OrganizationConfig	OrganizationConfig
Mail Enabled Public Folders	Organization	Organization	OrganizationConfig	OrganizationConfig
Mail Recipient Creation	Organization	Organization	OrganizationConfig	OrganizationConfig
Mail Recipients	Organization	Organization	OrganizationConfig	OrganizationConfig
Mail Tips	Organization	Organization	OrganizationConfig	OrganizationConfig
Mailbox Import Export	Organization	Organization	OrganizationConfig	OrganizationConfig

Mailbox Search	Organization	Organization	None	None
MailboxSearchApplication	Organization	Organization	OrganizationConfig	OrganizationConfig
Message Tracking	Organization	Organization	OrganizationConfig	OrganizationConfig
Migration	Organization	Organization	OrganizationConfig	OrganizationConfig
Monitoring	Organization	Organization	OrganizationConfig	OrganizationConfig
Move Mailboxes	Organization	Organization	OrganizationConfig	OrganizationConfig
OfficeExtensionApplication	Self	Self	OrganizationConfig	OrganizationConfig
My Custom Apps	Self	Self	OrganizationConfig	OrganizationConfig
My Marketplace Apps	Self	Self	OrganizationConfig	OrganizationConfig
MyAddressInformation	Self	Self	OrganizationConfig	OrganizationConfig
MyBaseOptions	Self	Self	OrganizationConfig	OrganizationConfig
MyContactInformation	Self	Self	OrganizationConfig	OrganizationConfig
MyDiagnostics	Self	Self	OrganizationConfig	OrganizationConfig
MyDisplayName	Self	Self	OrganizationConfig	OrganizationConfig
MyDistributionGroupMembership	MyGAL	MyGAL	None	None
MyDistributionGroups	MyGAL	MyDistributionGroups	OrganizationConfig	None
MyMobileInformation	Self	Self	OrganizationConfig	OrganizationConfig
MyName	Self	Self	OrganizationConfig	OrganizationConfig
MyPersonalInformation	Self	Self	OrganizationConfig	OrganizationConfig
MyProfileInformation	Self	Self	OrganizationConfig	OrganizationConfig
MyRetentionPolicies	Self	Self	OrganizationConfig	OrganizationConfig
MyTeamMailboxes	Organization	Organization	OrganizationConfig	OrganizationConfig
MyTextMessaging	Self	Self	OrganizationConfig	OrganizationConfig

MyVoiceMail	Self	Self	OrganizationConfig	OrganizationConfig
Organization Client Access	Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Configuration	Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Transport Settings	Organization	Organization	OrganizationConfig	OrganizationConfig
POP3 And IMAP4 Protocols	Organization	Organization	OrganizationConfig	OrganizationConfig
Public Folders	Organization	Organization	OrganizationConfig	OrganizationConfig
Receive Connectors	Organization	Organization	OrganizationConfig	OrganizationConfig
Recipient Policies	Organization	Organization	OrganizationConfig	OrganizationConfig
Remote and Accepted Domains	Organization	Organization	OrganizationConfig	OrganizationConfig
Reset Password	Organization	Organization	OrganizationConfig	OrganizationConfig
Retention Management	Organization	Organization	OrganizationConfig	OrganizationConfig
Role Management	Organization	Organization	OrganizationConfig	OrganizationConfig
Security Group Creation and Membership	Organization	Organization	OrganizationConfig	OrganizationConfig
Send Connectors	Organization	Organization	OrganizationConfig	OrganizationConfig
Support Diagnostics	Organization	Organization	OrganizationConfig	OrganizationConfig
TeamMailboxLifecycleApplication	Self	Self	OrganizationConfig	OrganizationConfig
Transport Agents	Organization	Organization	OrganizationConfig	OrganizationConfig
Transport Hygiene	Organization	Organization	OrganizationConfig	OrganizationConfig
Transport Queues	Organization	Organization	OrganizationConfig	OrganizationConfig
Transport Rules	Organization	Organization	OrganizationConfig	OrganizationConfig
UM Mailboxes	Organization	Organization	OrganizationConfig	OrganizationConfig
UM Prompts	Organization	Organization	OrganizationConfig	OrganizationConfig

Unified Messaging	Organization	Organization	OrganizationConfig	OrganizationConfig
UnScoped Role Management	Organization	Organization	OrganizationConfig	OrganizationConfig
UserApplication	Organization	Organization	OrganizationConfig	OrganizationConfig
User Options	Organization	Organization	OrganizationConfig	OrganizationConfig
View-Only Audit Logs	Organization	None	OrganizationConfig	None
View-Only Configuration	Organization	None	OrganizationConfig	None
View-Only Recipients	Organization	None	OrganizationConfig	None
WorkloadManagement	Organization	Organization	OrganizationConfig	OrganizationConfig

Explicit scopes

Explicit scopes are scopes that you set yourself to control which objects a management role can modify. Although implicit scopes are defined on a management role, explicit scopes are defined on a management role assignment. This enables the implicit scopes to be applied consistently across all management roles unless you choose to use an overriding explicit scope. For more information about management role assignments, see Understanding management role assignments.

Explicit scopes override the implicit write and configuration scopes of a management role. They don't override the implicit read scope of a management role. The implicit read scope continues to define what objects the management role can read.

Explicit scopes are useful when the implicit write scope of a management role doesn't meet the needs of your business. You can add an explicit scope to include nearly anything you want as long as the new scope doesn't exceed the bounds of the implicit read scope. The cmdlets that are part of a management role must be able to read information about the objects or containers that contain objects for the cmdlets to create or modify objects. For example, if the implicit read scope on a management role is set to `Self`, you can't add an explicit write scope of `Organization` because the explicit write scope exceeds the bounds of the implicit read scope.

For more information, see the following sections:

- Predefined Relative Scopes
- Custom Scopes

Predefined relative scopes

Exchange 2013 provides several predefined relative write scopes that you can use to modify scope of a management role. Predefined relative scopes provide an easy way for you to more closely

match the needs of your business without having to create custom scopes manually. They're called relative scopes because they're relative to the role assignee to which the associated role assignment is assigned. For example, the `Self` predefined relative scope restricts that write scope to the current user only. The `MyDistributionGroups` predefined relative scope restricts the write scope to the distribution group the current user owns only. Predefined relative scopes can only be used to scope recipient objects. Predefined relative scopes can't be used to scope configuration objects. The following table lists the predefined relative scopes that you can use.

Predefined relative scopes

Implicit scopes	Description
`Organization`	If `Organization` is present in the role's recipient write scope, the role can create or modify recipient objects across the Exchange organization. If `Organization` is present in the role's recipient read scope, roles can view any recipient object across the Exchange organization. This scope is used only with recipient read and write scopes.
`Self`	If `Self` is present in the role's recipient write scope, the role can modify only the properties of the current user's mailbox. If `Self` is present in the role's recipient read scope, the role can view only the properties of the current user's mailbox. This scope is used only with recipient read and write scopes.
`MyDistributionGroups`	If `MyDistributionGroups` is present in the role's recipient write scope, the role can create or modify distribution list objects owned by the current user. If `MyDistributionGroups` is present in the

	role's recipient read scope, the role can view distribution list objects owned by the current user. This scope is used only with recipient read and write scopes.

Predefined relative scopes are applied when you create a new management role assignment. During the creation of the role assignment, using the **New-ManagementRoleAssignment** cmdlet, you can specify a predefined relative scope using the *RecipientRelativeWriteScope* parameter. When the new role assignment is created, the new predefined role overrides the implicit write scope of the management role. You can't specify a custom recipient scope when you create a role assignment with a predefined relative scope. You can, however, specify a custom configuration scope if needed.

For more information about how to add a management role assignment with a predefined relative scope, see Add a role to a user or USG.

Custom scopes

Custom scopes are needed when neither the implicit write scope nor the predefined relative scopes meet the needs of your business. Custom scopes enable you to define, at a granular level, the scope to which your management role will be applied. For example, you might want to target a specific organizational unit (OU), a specific type of recipient, or both. Or, you might only want to allow a group of administrators to be able to manage a specific set of mailbox databases.

As with predefined relative scopes, custom scopes override the implicit write and organization configuration scopes defined on management roles. The implicit read scope on management roles continue to apply and the resulting custom scope must not exceed the boundaries of the implicit read scope. You can create the following three types of custom scopes:

- **OU scope** An OU scope, which is the simplest custom scope, is created using the *RecipientOrganizationalUnitScope* parameter on the **New-ManagementRoleAssignment** cmdlet. By specifying an OU scope when a role is assigned, the role assignee assigned the role can modify only recipient objects within that OU. For more information about how to add a management role assignment with an OU scope, see Add a role to a user or USG.
- **Recipient filter scope** Recipient filter scopes use filters to target specific recipients based on recipient type or other recipient properties such as department, manager, location, and more. For more information, see the Recipient Filter Scopes section.
- **Configuration scope** Configuration scopes use filters or lists to target specific servers based on server lists or filterable properties that can be defined on servers, such as an Active Directory site or a server role. Configuration scopes can also use database scopes to target specific databases based on database lists or filterable database properties. For more information, see the

Configuration Scopes section.

Simple and broad or complex and granular recipient and configuration custom scopes can be created by using the **New-ManagementScope** cmdlet. When you create either a recipient or configuration scope, only the recipient, server, or database objects that match their respective scopes are returned. When these scopes are applied to a role assignment using the **New-ManagementRoleAssignment** or **Set-ManagementRoleAssignment** cmdlets, only the objects that match the scopes can be modified by the role assignees who are assigned the role. After a custom scope has been created, you can't change the scope type. A recipient scope is always a recipient scope and a configuration scope is always a configuration scope.

By default, a custom scope enables a role assignee to access a set of objects that match the scopes you define. However, they don't actively exclude access to other role assignees who aren't also assigned the same or equivalent scope. Any custom scope can access the same objects if the lists or filters on those scopes match the same objects. There might be objects where this behavior isn't wanted, such as in the case of executives. For these objects, you can define exclusive scopes. Exclusive scopes use filters or lists in the same way as regular scopes but unlike regular scopes, deny access to objects included in the scope to anyone who isn't part of the same or equivalent exclusive scope. For more information about exclusive scopes, see Understanding exclusive scopes.

Recipient filter scopes

Recipient filter scopes enable you to control which recipient objects role assignees can manage by evaluating one or more properties on a recipient object against a value that you specify in a filter statement. Recipients included in recipient scopes are mailboxes, mail-enabled users, distribution groups and mail contacts. Only the recipients that match the filter you specify can be managed by the role assignees assigned that role assignment. An example of a filter statement is `{ Name -Eq "David" }` where **Name** is the property on the recipient object that's being evaluated and **David** is the value you want to evaluate against the property. The **-Eq** comparison operator indicates that the value stored in the property must be equal to the value that was specified for the filter to be true. If the filter is true, that recipient is included in scope.

Recipient filter scopes are created by specifying the recipient filter to use with the *RecipientRestrictionFilter* parameter on the **New-ManagementScope** cmdlet. By default, the **New-ManagementScope** cmdlet creates regular scopes. If you want to create an exclusive scope, include the *Exclusive* switch along with the *RecipientRestrictionFilter* parameter.

When you create a recipient restriction filter, Exchange evaluates the filter you provided against every recipient object in the organization by default. If you want to limit which recipients the scope evaluates, you can use the *RecipientRoot* parameter along with the *RecipientRestrictionFilter* parameter. The *RecipientRoot* parameter accepts an OU. When you use the *RecipientRoot* parameter, Exchange evaluates only the recipients included in the specified OU against the filter you provided.

When you add a recipient filter scope to a role assignment, specify the name of the recipient scope in the *CustomRecipientWriteScope* parameter on the **New-ManagementRoleAssignment** if you're

creating a new role assignment, or the **Set-ManagementRoleAssignment** cmdlet if you're updating an existing role assignment. Each role assignment can have one recipient scope, including predefined relative scopes. You can add one configuration scope to the same role assignment you added a recipient scope to.

For more information about filter syntax and for a full list of filterable recipient properties on recipients, see Understanding management role scope filters.

Configuration scopes

The following are the two types of configuration scopes offered in Exchange 2013:

- **Server scopes** There are two types of server scopes, server filter scopes and server list scopes. Server configuration, including Receive connectors, transport queues, server certificates, virtual directories, and so on, can be managed if a server object is included in a server scope.
  - **Server filter scopes** Server filter scopes enable you to control which server objects role assignees can manage by evaluating one or more properties on a server object against a value that you specify in a filter statement. To create a server filter scope, use the *ServerRestrictionFilter* parameter on the **New-ManagementScope** cmdlet.
  - **Server list scopes** Server list scopes enable you to control which server objects role assignees can manage by defining a list of servers that a role assignee can access. To create a server list scope, use the *ServerList* parameter on the **New-ManagementScope** cmdlet.
- **Database scopes** There are two types of database scopes, database filter scopes and database list scopes. Database configuration that can be managed if a database object is included in a database scope include database quota limits, database maintenance, public folder replication, whether a database is mounted, and so on. In addition to database configuration, database scopes can also be used to control which databases recipients can be created in. If you have pre-Exchange 2010 SP1 servers in your organization, see the Database scopes and previous versions of Exchange section later in this topic.
  - **Database filter scopes** Database filter scopes enable you to control which database objects role assignees can manage by evaluating one or more properties on a database object against a value that you specify in a filter statement. To create a database filter scope, use the *DatabaseRestrictionFilter* parameter on the **New-ManagementScope** cmdlet.
  - **Database list scopes** Database list scopes enable you to control which database objects role assignees can manage by defining a list of databases that a role assignee can access. To create a database list scope, use the *DatabaseList* parameter on the **New-ManagementScope** cmdlet.

For more information about filter syntax and for a full list of filterable server and database properties, see Understanding management role scope filters.

Server and database lists can be defined by specifying each server and database you want to include in their respective scopes. Multiple servers or databases can be specified in their respective scopes by separating the server and database names with commas.

When you add a server or database configuration scope to a role assignment, specify the name of

the server or database configuration scope in the *CustomConfigWriteScope* parameter on the **New-ManagementRoleAssignment** cmdlet if you're creating a new role assignment, or the **Set-ManagementRoleAssignment** cmdlet if you're updating an existing role assignment. Each role assignment can only have one configuration scope.

In addition to controlling which databases role assignees can manage, database scopes also enable you to control which databases role assignees can create mailboxes on. This is separate from controlling which recipients a role assignee can manage. If a role assignee has permissions to create a new mailbox, mail-enable an existing user, or move mailboxes, you can further refine their permissions by using database scopes to control the database on which the mailbox is created, or which database a mailbox is moved to. Controlling which recipients a role assignee can manage is done using a recipient scope specified in the *CustomRecipientWriteScope* parameter on the **New-ManagementRoleAssignment** or **Set-ManagementRoleAssignment** cmdlet. Controlling which databases a mailbox can be created on or moved to is controlled using a database scope specified in the *CustomConfigurationWriteScope* parameter on the same cmdlets.

Note:
Automatic mailbox distribution can be controlled using database scopes.

Exchange features may require either server scopes, database scopes, or both, to be managed. If a feature requires both server and database scopes to be managed, two role assignments must be created and assigned to the role assignee that should have access to manage the feature. One role assignment should be associated with the server scope, and one role assignment should be associated with the database scope.

Some cmdlets may use configuration scopes that aren't immediately obvious. The following table includes a list of cmdlets and the configuration scopes that you can use to control their usage. For cmdlets included in the recipients feature area, configuration scopes enable you to control on which databases recipients can be created. They don't control which recipients can be managed. The **Required scopes** column can contain the following:

- **Database** To run the cmdlet, the role assignee must be assigned a role assignment with a database scope that includes the database to be managed or the role's implicit configuration write scope must include the database to be managed.
- **Server** To run the cmdlet, the role assignee must be assigned a role assignment with a server scope that includes the server to be managed or the role's implicit configuration write scope must include the server to be managed.
- **Server or database** To run the cmdlet, the role assignee must be assigned a role assignment where either a database scope includes the database being managed, or where a server scope includes the server where the database is located. Or, the role's implicit configuration write scope must contain the database to be managed, or contain the server where the database is located, and the role assignment can't have a custom write scope.
- **Server and database** To run this cmdlet, the role assignee must be assigned two role assignments. The first role assignment must include a database scope that includes the database to be managed. The second role assignment must include a server scope that includes the server

where the database is located. The role assignments can have custom configuration scopes defined, or the role assignments can inherit the implicit configuration write scope from the role. To inherit the implicit write scope from the role, the role assignment can't have a custom write scope.

Feature areas and applicable database and server scopes

Feature area	Cmdlet	Required scopes
Databases	**Dismount-Database**	Database
Databases	**Mount-Database**	Database
Databases	**Move-DatabasePath**	Server and database
Databases	**Remove-MailboxDatabase**	Server or database
Databases	**Set-MailboxDatabase**	Database
High availability	**Add-DatabaseAvailabilityGroupServer**	Server
High availability	**Add-MailboxDatabaseCopy**	Server
High availability	**Move-ActiveMailboxDatabase**	Server
High availability	**Remove-DatabaseAvailabilityGroupServer**	Server
High availability	**Remove-MailboxDatabaseCopy**	Server or database
High availability	**Resume-MailboxDatabaseCopy**	Server or database
High availability	**Set-MailboxDatabaseCopy**	Server or database
High availability	**Suspend-MailboxDatabaseCopy**	Server or database

High availability	**Update-MailboxDatabaseCopy**	Server or database
Recipients	**Connect-Mailbox**	Database
Recipients	**Enable-Mailbox**	Database
Recipients	**New-Mailbox**	Database
Recipients	**New-MoveRequest**	Database
Troubleshooting	**Test-MapiConnectivity**	Database

Database scopes and previous versions of Exchange

Database scopes were first introduced in Microsoft Exchange 2010 Service Pack 1 (SP1) and continue to be supported in Exchange 2013. Versions of Exchange prior to Exchange 2010 SP1 support only recipient scopes and server configuration scopes. When you create a new database scope on an Exchange 2010 SP1 or later server, you'll receive the following warning:

```
WARNING: Database management scopes will only be applied
when a user connects to a server running Exchange 2010 SP1
or later. Servers running a version of Exchange prior to
Exchange 2010 SP1 won't apply any roles from a role
assignment linked to a database scope. Database management
scopes also won't be visible to the Get-ManagementScope
cmdlet when it's run from a pre-Exchange 2010 SP1 server.
```

When you create a database scope, it's only applied to users who connect to servers running Exchange 2010 SP1 or later. Users who connect to pre-Exchange 2010 SP1 servers won't have any role assignments associated with database scopes applied to them. This means that any permissions provided by these role assignments won't be granted to users when they connect to pre-Exchange 2010 SP1 servers. Database scopes can't be created, removed, modified, or viewed from pre-Exchange 2010 SP1 servers.

A database scope can include any database in your Exchange organization. This includes Exchange Server 2007, Exchange 2010, and Exchange 2013 servers. This enables you to control which databases, regardless of Exchange version, that users can manage. As with other database scopes, role assignments associated with database scopes that contain Exchange 2007 and Exchange 2010 databases are only applied to users when they connect to an Exchange 2010 SP1 or later server.

Users who connect to a pre-Exchange 2010 SP1 server can view and modify role assignments associated with database scopes. This includes changing the configuration scope on an existing

role assignment to a server scope if it's currently associated with a database scope. However, if the configuration scope on a role assignment is changed to a server scope and a user later wants to change it back to a database scope, or if the user wants to change the configuration scope to another database scope, the user must make the change while connected to an Exchange 2010 SP1 or later server. Users can only specify server scopes when they change the configuration scope on a role assignment if they're connected to a pre-Exchange 2010 SP1 server.

Understanding management role scope filters

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-29*

Management role scope filters can be used to define management scopes that are highly customizable. Using scope filters, you can create a scope that matches how you segment your recipients, databases, and servers so that administrators can manage only those objects they should have access to. Scope filters can use nearly any recipient, database, or server object property.

To use management role scope filters, you must be familiar with management role scopes. For more information about management role scopes, see Understanding management role scopes.

Filtered custom scopes in Microsoft Exchange Server 2013 are created by using the **New-ManagementScope** cmdlet. The two types of filtered scopes, recipient and configuration (which consists of server and database scopes), are divided into regular scopes and exclusive scopes. The following list shows which parameter on the **New-ManagementScope** cmdlet to use to create each type of filtered scope:

- **Recipient regular filtered scope** To create this type of filtered scope, use the *RecipientRestrictionFilter* parameter.
- **Recipient exclusive filtered scope** To create this type of filtered scope, use the *RecipientRestrictionFilter* parameter along with the *Exclusive* switch.
- **Server-based configuration regular filtered scope** To create this type of filtered scope, use the *ServerRestrictionFilter* parameter.
- **Server-based configuration exclusive filtered scope** To create this type of filtered scope, use the *ServerRestrictionFilter* parameter along with the *Exclusive* switch.
- **Database-based configuration regular filtered scope** To create this type of filtered scope, use the *DatabaseRestrictionFilter* parameter.
- **Database-based configuration exclusive filtered scope** To create this type of filtered scope, use the *DatabaseRestrictionFilter* parameter along with the *Exclusive* switch.

When you create a filtered custom scope, the scope attempts to match the filter with any objects accessible within the implicit read scope of the management role. If an object is found, it's included in the results returned by the filter, and the object is made available to the management role by the custom scope. A filter can't return results that are outside of the implicit read scope of the management role.

If you specify a recipient filter using the *RecipientRestrictionFilter* parameter, you can use the *RecipientRoot* parameter to specify an organizational unit (OU) to restrict the filter to. When you specify an OU in the *RecipientRoot* parameter, the recipient filter attempts to match recipients that reside in that OU only, rather than within the entire implicit read scope.

To create a management scope using the filterable properties included in this topic, see Create a regular or exclusive scope.

Filter syntax

Both recipient and configuration filters use the same syntax to create a filter query. All filter queries must have, at minimum, the following components:

- **Opening bracket** The opening brace ({) indicates the start of the filter query.
- **Property to examine** The property is the value on an object that you want to test. For example, this can be the city or department on a recipient object, an Active Directory site name or server name on a server configuration object, or a database name on a database configuration object.
- **Comparison operator** The comparison operator directs how the query should evaluate the value that you specify against the value that's stored in the property. For example, comparison operators can be **Eq**, which means equal to; **Ne**, which means not equal to; **Like**, which means similar to, and so on. For a full list of operators that you can use in the Exchange Management Shell, see Comparison operators.
- **Value to compare** The value you specify in the filter query will be compared to the value that's stored in the property you specified. The value you specify must be enclosed in quotation marks ("). If you want to specify a partial string, you can enclose the string you provide in wildcard characters (*) and use a comparison operator that supports wildcard characters, such as **Like**. Any string that contains the partial string will match the filter query.
- **Closing bracket** The closing brace (}) indicates the end of the filter query.

The following components are optional and enable you to create more complex filter queries:

- **Parentheses** As in mathematics, parentheses, ( ), in a filter query enable you to force the order in which an operation occurs. Innermost parentheses are evaluated first and the filter query works outward to the outermost parentheses.
- **Logical operators** Logical operators tie together one or more comparison operations and require the filter query to evaluate the entire statement. For example, logical operators include **And**, **Or**, and **Not**.

When put together, a simple query looks like `{ City -Eq "Vancouver" }`. This filter matches any recipient where the value in the property **City** equals the string "Vancouver".

Another, more complex, query is `{ ((City -Eq "Vancouver") -And (Department -Eq "Sales")) -Or (Title -Like "*Manager*") }`. The filter query is evaluated in the following order:

1. The properties **City** and **Department** are evaluated. Each is set to either `True` or `False`, depending on the values stored in each property.
2. The results of the **City** and **Department** statements are then evaluated. If both are `True`, the entire **And** statement becomes `True`. If one or both are `False`, the entire **And** statement becomes `False`. The following applies:
   - If the **And** statement evaluates as `True`, the entire filter query becomes `True` because the **Or** operator indicates that one side of the query, or the other, must be `True`. The object is exposed to the role assignment.
   - If the **And** statement is `False`, the filter query continues on to evaluate the **Title** property.
3. The **Title** property is then evaluated. It's set to `True` or `False`, depending on the value that's stored in the **Title** property. The following applies:
   - If the **Title** property evaluates as `True`, the entire filter query becomes `True` because the **Or** operator indicates that one side of the query, or the other, must be `True`. The object is exposed to the role assignment.
   - If the **Title** property evaluates as `False`, the entire filter query evaluates as `False`, and the object isn't exposed to the role assignment.

The following table shows an example with values, which indicates when the complex query would evaluate as `True`, and when it would evaluate as `False`.

Complex query

City	Department	Title	Result
Vancouver (True)	Sales (True)	CEO (False)	True because both **City** and **Department** evaluated as True. **Title** isn't evaluated because the filter query conditions are already satisfied.
Seattle (False)	Sales (True)	IT Manager (True)	True because **Title** evaluated as True. The results of the **City** and **Department** comparison are discarded because **Title** evaluated as True, which

			satisfied the filter query conditions. **Note:** IT Manager matches the filter query because the **Like** comparison operator was used, which matches partial strings when wildcard characters (*) are used in the filter query.
Vancouver (True)	Marketing (False)	Writer (False)	False because **City** and **Department** didn't both evaluate as True, and **Title** also didn't evaluate as True.

Filterable recipient properties

You can use almost any property on a recipient object when you create a recipient filter. For a list of filterable recipient properties, see Filterable properties for the -RecipientFilter parameter. Although this topic discusses the properties that can be used with the *RecipientFilter* parameter on other cmdlets, most of these properties also work with the *RecipientRestrictionFilter* parameter on the **New-ManagementScope** cmdlet.

Filterable server properties

You can use the following server properties when you create a management scope with the *ServerRestrictionFilter* parameter:

- **CurrentServerRole**
- **CustomerFeedbackEnabled**
- **DataPath**
- **DistinguishedName**
- **ExchangeLegacyDN**
- **ExchangeLegacyServerRole**
- **ExchangeVersion**
- **Fqdn**
- **Guid**
- **InternetWebProxy**

- **Name**
- **NetworkAddress**
- **ObjectCategory**
- **ObjectClass**
- **ProductID**
- **ServerRole**
- **ServerSite**
- **WhenChanged**
- **WhenChangedUTC**
- **WhenCreated**
- **WhenCreatedUTC**

Filterable database properties

You can use the following database properties when you create a management scope with the *DatabaseRestrictionFilter* parameter:

- **AdminDisplayName**
- **AllowFileRestore**
- **BackgroundDatabaseMaintenance**
- **CircularLoggingEnabled**
- **DatabaseCreated**
- **DeletedItemRetention**
- **Description**
- **DistinguishedName**
- **EdbFilePath**
- **EventHistoryRetentionPeriod**
- **ExchangeLegacyDN**
- **ExchangeVersion**
- **Guid**
- **IssueWarningQuota**
- **LogFilePrefix**
- **LogFileSize**
- **LogFolderPath**
- **MasterServerOrAvailabilityGroup**
- **MountAtStartup**
- **Name**
- **ObjectCategory**
- **ObjectClass**
- **RetainDeletedItemsUntilBackup**
- **Server**
- **WhenChanged**
- **WhenChangedUTC**

- **WhenCreated**
- **WhenCreatedUTC**

Understanding exclusive scopes

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-05*

Exclusive scopes are a special type of explicit management scope that can be associated with management role assignments. Exclusive scopes are designed to enable situations where you have a group of highly valuable objects, such as a CEO mailbox, and you want to tightly control who has access to manage those objects.

A role assignment that has an exclusive scope is called an *exclusive role assignment*.

When you create an exclusive scope, only those who are assigned that exclusive scope, or an equivalent exclusive scope, can modify the objects that match the scope. Role assignees who aren't assigned that exclusive scope, or an equivalent, can't modify the objects that match the scope, even if their own roles have scopes that would otherwise include the objects. Exclusive scopes override any other regular scope that isn't exclusive. This behavior is similar to how a deny access control entry (ACE) on an Active Directory access control list (ACL) functions.

An *equivalent exclusive scope* refers to another exclusive scope that matches some of the same objects as another exclusive scope. The scopes don't have to match the same complete set of objects. Both scopes may be able to modify some, or all, of the objects that match them.

Creating exclusive scopes

Exclusive scopes can be created like any other explicit scope. You can specify a prebuilt relative scope; a recipient, database, or server filter; or a database or server list. Unlike regular scopes, which don't take effect until you associate a scope to a management role assignment, the deny aspect of an exclusive scope takes effect immediately. This means that as soon as an exclusive scope is created, the objects contained within that scope are immediately no longer accessible by any user until the role assignment has been created.

After the assignment has been created, the exclusive scope provides access to those assigned the management role and scope. If another equivalent exclusive scope matches the same objects, the role assignment associated with that exclusive scope is still able to access the objects.

For more information about management scope filters, see Understanding management role scope filters.

Important:
Active Directory replication times should be taken into account when making changes to any management role components, including exclusive scopes.

If you have objects contained within more than one exclusive scope, being assigned to any one of the exclusive scopes provides access to the objects. For more information, see Exclusive and regular scope interaction later in this topic.

Exclusive scopes control only the explicit recipient or configuration write scope of a role assignment. The implicit recipient or configuration read scope of the role assigned to a user or group still applies. This means that the following applies:

- Those assigned a role continue to see objects that match the role's implicit read scope.
- Those assigned other roles may be able to see objects contained within an exclusive scope, if the read scopes of the other roles include the objects. However, the objects can only be modified by those who are assigned a role associated with the exclusive scope.

Exclusive scopes can only be used with administrative or specialist roles and can't be used with end-user roles. For more information about roles, see Understanding management roles.

Exclusive and regular scope interaction

The figure at the end of this section illustrates how exclusive scopes interact with each other, and with regular scopes. The users in the figure all have the following attributes associated with them.

User	City	Title	Department
Terry	Vancouver	Accountant	Accounting
David	Vancouver	Writer	Marketing
Walter	Vancouver	Manager	Marketing
Bob	Vancouver	CEO	Board
Christine	Vancouver	President	Board
Fred	Vancouver	CFO	Executives
Martin	Vancouver	CIO	Executives
Kim	Vancouver	Vice President, Operations	Executives
Jennifer	Vancouver	Vice President, Technology	Executives

The following three management role assignments in the figure manage the users in the preceding table. Each has an associated scope, some of which are exclusive scopes.

Role assignment	Scope filter	Exclusive or regular
Recipient Administrators	City = Vancouver	Regular
VIP Administrators	Title = CEO or CFO or CIO or President	Exclusive
Executive Administrators	Department = Executives	Exclusive

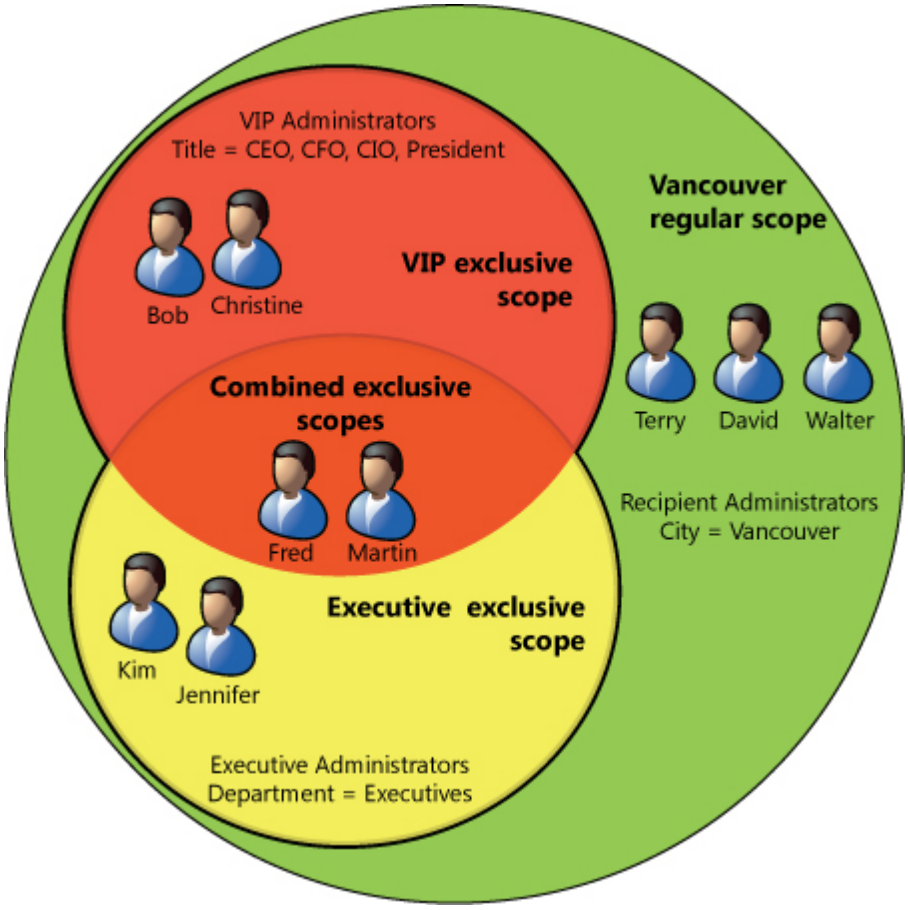
The Recipient Administrators role assignment has a scope that matches all of the users because every user is located in Vancouver. Without any exclusive scopes, this would mean that the Recipient Administrators role assignment could manage any of the users. However, this organization has created two exclusive scopes: VIP Administrators and Executive Administrators. These exclusive scopes restrict who can manage the users that match their respective scope filters. The VIP Administrators role assignment has a scope filter that matches any user who has a title of CEO, CFO, CIO, or President. The Executive Administrators role assignment has a scope filter that matches any user who is in the Executives department.

When the regular and exclusive scopes are evaluated, the following is the result:

- The Recipient Administrators role assignment can manage the users Terry, David, and Walter. This role assignment can't manage any of the other users because they match the exclusive scope filters of the VIP Administrators and Executive Administrators role assignments.
- The VIP Administrators role assignment can manage the users Bob, Christine, Fred, and Martin. This is because the exclusive scope filter associated with this role assignment matches the attributes on these objects. This role assignment can't manage the users Kim and Jennifer because their attributes don't match this exclusive scope.
- The Executive Administrators role assignment can manage the users Kim, Jennifer, Fred, and Martin. This is because the exclusive scope filter associated with this role assignment matches the attributes on these objects. This role assignment can't manage the users Bob and Christine because their attributes don't match this exclusive scope.

Notice that Fred and Martin are accessible by both exclusive scopes. This is because the attributes on these users match the filters of both exclusive scopes.

Interaction between exclusive scopes and regular scopes



For more information about management scopes, see Understanding management role scopes.

Understanding management role assignments

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-04*

A *management role assignment*, which is part of the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013, is the link between a management role and a role assignee. A *role assignee* is a role group, role assignment policy, user, or universal security group (USG). A role must be assigned to a role assignee for it to take effect. For more information about RBAC, see Understanding Role Based Access Control.

Note:

This topic focuses on advanced RBAC functionality. If you want to manage basic Exchange 2013 permissions, such as using the Exchange admin center (EAC) to add and remove members to and from role groups, create and modify role groups, or create and modify role assignment policies, see Permissions.

This topic discusses the assignment of roles to role groups and role assignment policies and direct role assignment to users and USGs. It doesn't talk about assignment of role groups or role assignment policies to users. For more information about role groups and role assignment policies, which are the recommended way to assign permissions to users, see the following topics:

- Understanding management role groups
- Understanding management role assignment policies

You can create the following types of role assignments, which are explained in detail later in this topic:

- Regular and delegating role assignments
- Exclusive role assignments

Managing role assignments

When you change role assignments, the changes you make will probably be between role groups and role assignment policies. By adding, removing, or modifying role assignments to or from these role assignees, you can control what permissions are given to your administrators and users, in effect turning on and off management of related features.

You might also want to assign roles directly to users or USGs. This is a more advanced task that enables you to define at a granular level what permissions your users are given. Although this provides you with flexibility, it also increases the complexity of your permissions model. For example, if the user changes jobs, you might need to manually reassign the roles assigned to that user to another user. This is why we recommend that you use role groups and role assignment policies to give permissions to your users. You can assign the roles to a role group or role assignment policy, and then just add or remove members of the role group, or change role assignment policies as needed.

You can add, remove, and enable role assignments, modify the management scope on an existing role assignment, and move role assignments to other role assignees. The process of assigning roles to role groups, role assignment policies, users, and USGs is largely the same for each role assignee. The following are the only exceptions:

- Role assignment policies can only be assigned end-user management roles.
- Role assignment policies can't be assigned delegating role assignments.
- You can't specify a management scope when creating a role assignment to role assignment policies.

For more information about managing role assignments, see the following topics:

- Role groups:
  - Manage role groups
- Role assignment policies:
  - Manage role assignment policies
  - Change a role assignment
- Users and USGs:
  - Add a role to a user or USG

- Remove a role from a user or USG
- Change a role assignment
- Change a role scope
- Delegate role assignments

Regular and delegating role assignments

Regular role assignments enable the role assignee to access the management role entries made available by the associated management role. If multiple management roles are assigned to a role assignee, the management role entries from each management role are aggregated and applied. This means that if a role assignee is assigned the Transport Rules and Journaling roles, the roles are combined, and all the associated management role entries are given to the role assignee. If the role assignee is a role group or role assignment policy, the permissions provided by the roles are then given to the users assigned to the role group or role assignment policy. For more information about management roles and role entries, see Understanding management roles.

Delegating role assignments doesn't give access to manage features. Delegating role assignments gives a role assignee the ability to assign the specified role to other role assignees. If the role assignee is a role group, any member of the role group can assign the role to another role assignee. By default, only the Organization Management role group has the ability to assign roles to other role assignees. Only the user that installed Exchange 2013 is a member of the Organization Management role group by default. You can, however, add other users to this role group as needed, or create other role groups and assign delegating role assignments to those groups.

Note:

Delegating role assignments enables role assignees to delegate management roles to other role assignees. This doesn't enable users to delegate role groups. For more information about role group delegation, see Understanding management role groups.

If you want a user to be able to manage a feature and assign the role that gives permissions to use the feature to other users, assign the following:

1. A regular role assignment for each management role that grants access to the features that need to be managed.
2. A delegating role assignment for each management role that you allow to be assigned to other role assignees.

The regular and delegating role assignments for a role assignee don't need to be identical. For example, a user is a member of a role group assigned the Transport Rules role using a regular role assignment. This enables the user to manage the Transport Rules feature. However the user isn't assigned a delegating role assignment for the Transport Rules role so the user can't assign this role to other users. However, the user is a member of a role group assigned the Journaling management role using a delegating role assignment. The role group the user is a member of doesn't have a regular role assignment for the Journaling role but because it has a delegating role assignment, the user can assign the role to other role assignees.

Management scopes

When you create either a regular or delegating management role assignment, you have the option of creating the assignment with a management scope to limit the objects that the user can manipulate. You can create recipient scopes or configuration scopes. Recipient scopes enable you to control who can manipulate mailboxes, mail users, distribution groups, and so on. Configuration scopes enable you to control who can manipulate servers and databases.

Recipient and configuration scopes enable you to segment the management of server, database or recipient objects in your organization. For example, a recipient scope can be added to a role assignment so that administrators in Vancouver can only manage recipients in the same office. A server configuration scope could be added to a different role assignment so that administrators in Sydney can only manage servers in their Active Directory site.

Scopes enable permissions to be assigned to groups of users and enable you to direct where those administrators can perform their administration. This enables you to create a permissions model that maps to your geographic or organizational boundaries.

You can create an assignment with a predefined scope, or you can add a custom scope to the assignment. Predefined scopes, such as limiting a user to only his or her mailbox or distribution groups, can be applied using options available on the assignment itself. Alternatively, you can create a custom recipient or configuration scope, and then add that scope to the role assignment. Custom scopes give you more granularity over which objects are included in the scope.

You can't specify predefined and custom scopes on the same assignment. You also can't mix exclusive and regular scopes on the same assignment.

Each role assignment can only have one recipient scope and one configuration scope. If you want to apply more than one recipient scope, or one configuration scope, to a role assignee for the same management role, you must create multiple role assignments.

With neither a custom or predefined scope, role assignments are limited to the recipient and configuration scopes that are defined on the role itself. These scopes are called implicit scopes. Any role assignment that doesn't have a predefined or custom scope inherits the implicit scopes from the role it's associated with.

For more information about scopes, see Understanding management role scopes.

Exclusive role assignments

Exclusive role assignments are created when you associate an exclusive scope with a role assignment. Exclusive scopes work like regular scopes and enable role assignees to manage recipients that match the exclusive scope. However, unlike regular scopes, all other role assignees are denied the ability to manage the recipient, even if the recipient matches scopes applied to their role assignments. This can be useful when you want to limit who can manage a recipient to a few administrators. Only those specific administrators can manage the recipient, and all other

administrators are denied access.

For example, consider the following:

- John is an executive at Contoso. His mailbox matches an exclusive scope called VIP Users, which is associated with the VIP Restricted exclusive assignment.
- John's mailbox is also included in a regular scope called Redmond Users, which is associated with the Redmond Administration regular assignment.
- Bill is an administrator who is associated with the VIP Restricted exclusive assignment.
- Chris is an administrator who is associated with the Redmond Administration regular assignment.

Because John's mailbox matches the VIP Users exclusive scope, only Bill can manage his mailbox. Even though John's mailbox also matches the Redmond Users regular scope, Chris isn't associated with the VIP Restricted exclusive assignment. Therefore, Exchange denies Chris the ability to manage John's mailbox. For Chris to manage John's mailbox, Chris needs to be assigned an exclusive assignment that has an exclusive scope that matches John's mailbox.

For more information, see Understanding exclusive scopes.

Built-in role groups

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-08-21*

Microsoft Exchange Server 2013 includes several management role groups by default. The following built-in role groups provide you with a preconfigured set of roles that you can assign to various administrator and specialist users in your organization.

Note:

Role groups don't control access to end-user mailbox features. To control access to end-user mailbox features, see Understanding management role assignment policies.

- Organization Management
- View-only Organization Management
- Recipient Management
- UM Management
- Help Desk
- Hygiene Management
- Compliance Management
- Records Management
- Discovery Management
- Public Folder Management
- Server Management

- Delegated Setup

For more information about role groups, see Understanding management role groups.

Organization Management

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Organization Management management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Administrators that are members of the Organization Management role group have administrative access to the entire Exchange 2013 organization and can perform almost any task against any Exchange 2013 object, with some exceptions. By default, members of this role group can't perform mailbox searches and management of unscoped top-level management roles. For more information, see the "Delegating Only Role Assignments" section later in this topic.

Important:
The Organization Management role group is a very powerful role and as such, only users or universal security groups (USGs) that perform organizational-level administrative tasks that can potentially impact the entire Exchange organization should be members of this role group.

This role group is equivalent to the Exchange Organization Administrators role in Exchange Server 2007.

For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

By default, the account that's used to install Exchange 2013 in the organization is added as a member of the Organization Management role group. This account can then add other members to the role group as needed.

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group

delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or USGs that are members of this role group.

```
Get-RoleGroupMember "Organization Management"
```

For more information about the members of a role group, see Manage role groups.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the “Create a role group” section in Manage role groups.

The following are some ways you might want to customize this role:

- **Permissions owner** If the permissions in your organization are controlled by a specific group other than the Exchange administrators, you can create a role group and move the regular and delegating role assignments for the Role Management role to the new role group. Doing so prevents members of the Organization Management role group from managing any RBAC permissions.
- **Active Directory split permissions** If the creation of security principals in your organization, such as user accounts, is controlled by a specific group other than the Exchange administrators, you can create a role group and move the regular and delegating role assignments for the Mail Recipient Creation role and the Security Group Creation and Membership role to the new role group. Doing so prevents members of the Organization Management role group from creating

Active Directory objects. They can, however, continue to mail-enable the new Active Directory objects. For more information about split permissions, see Understanding split permissions.

Customization limitations

Any role can be added to or removed from this role group, with the following limitations:

- Every role must have at least one delegating role assignment to a role group or USG before the delegating role assignment can be removed from this role group.
- The Role Management role must have at least one regular role assignment to a role group or USG before the regular role assignment can be removed from this role group.

These limitations are intended to help prevent you from inadvertently locking yourself out of the system. By requiring that at least one delegating role assignment exists between every role and one or more role groups or USGs, you will always be able to assign roles to role assignees. By requiring that at least one regular role assignment exists between the Role Management role and one or more role groups or USGs, you will always be able to configure role groups and role assignments.

Important:
These limitations require that role groups or USGs be the targets of the delegating and regular role assignments. You can't remove a delegating role assignment or the regular assignment for the Role Management role if the last assignment is to a user.

Delegating only role assignments

Some role assignments between the Organization Management role group and management roles, such as Mailbox Search and Unscoped Role Management, are delegating only role assignments. These roles allow access to sensitive or personal information, such as the contents of mailboxes, or enable the creation of powerful unscoped management roles.

Delegating only role assignments enable members of the Organization Management role group only to assign the associated roles to other role groups, management role assignment policies, users, or USGs. Members of the Organization Management role group aren't given, by default, any permissions that the roles provide. This helps avoid accidental exposure to personal information or accidental elevation of privileges.

Members of the Organization Management role group can, however, assign themselves any role, in effect enabling them to perform any task. For example, a member of the Organization Management role group can assign the Mailbox Search role to the Organization Management role group. After this role assignment is made, members of the Organization Management role group can perform tasks enabled by the Mailbox Search role.

For more information about delegating role assignments, see Understanding management role assignments.

Additional permissions

The permissions granted to members of the Organization Management role group are primarily determined by the management roles assigned to the role group. However, not all tasks that you need to perform are covered by management roles. Some tasks occur outside of the Exchange management tools, and therefore the RBAC permissions model doesn't apply. For these tasks, permissions are provided by adding the Organization Management role group to the access control lists (ACLs) of certain Active Directory objects.

The following tasks are granted permissions by way of ACLs on Active Directory objects and not by management roles assigned to the Organization Management role group:

- Running DomainPrep and ForestPrep using Setup.exe
- Deploying additional servers in the organization

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.
- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Active Directory Permissions role	X	X	`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Address Lists role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
ApplicationImpersonation role		X	Organization	Organization	None	None
ArchiveApplication role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Audit Logs role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Cmdlet Extension Agents role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Data Loss Prevention role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Database Availability Groups role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Database Copies role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Databases role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Disaster Recovery role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Distribution Groups role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Edge Subscriptions role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
E-Mail Address Policies role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Connectors role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Server Certificates role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Servers role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Virtual Directories role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Federated Sharing role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Information Rights Management role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Journaling role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Legal Hold	X	X	Organization	Organization	OrganizationConfig	None

role						
LegalHoldApplication role		X	Organization	Organization	OrganizationConfig	OrganizationConfig
Mail Enabled Public Folders role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Mail Recipient Creation role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Mail Recipients role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Mail Tips role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Mailbox Import Export role		X	Organization	Organization	OrganizationConfig	OrganizationConfig
Mailbox Search role		X	Organization	Organization	None	None
MailboxSearchApplication role		X	Organization	Organization	OrganizationConfig	OrganizationConfig
Message Tracking role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Migration role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Monitoring	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

role						
Move Mailboxes role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
OfficeExtensionApplication role		X	Self	Self	OrganizationConfig	OrganizationConfig
Organization Client Access role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Configuration role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Transport Settings role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
POP3 and IMAP4 Protocols role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Public Folders role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Receive Connectors role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Recipient Policies role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Remote and	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Accepted Domains role						
Reset Password role		X	Organization	Organization	OrganizationConfig	OrganizationConfig
Retention Management role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Role Management role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Security Group Creation and Membership role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Send Connectors role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Support Diagnostics role		X	Organization	Organization	OrganizationConfig	OrganizationConfig
TeamMailboxLifecycleApplication role		X	Self	Self	OrganizationConfig	OrganizationConfig
Transport Agents role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Transport Hygiene role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Transport Queues role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Transport Rules role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
UM Mailboxes role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
UM Prompts role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Unscoped Role Management role		X	Organization	Organization	OrganizationConfig	OrganizationConfig
Unified Messaging role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
UserApplication role		X	Organization	Organization	OrganizationConfig	OrganizationConfig
User Options role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
View-Only Audit Logs role	X	X	Organization	None	OrganizationConfig	None
View-Only Configuration role	X	X	Organization	None	OrganizationConfig	None

View-Only Recipients role	X	X	Organization	None	OrganizationConfig	None
WorkloadManagement role	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
My Custom Apps role		X	Self	Self	OrganizationConfig	OrganizationConfig
My Marketplace Apps role		X	Self	Self	OrganizationConfig	OrganizationConfig
MyBaseOptions role		X	Self	Self	OrganizationConfig	OrganizationConfig
MyContactInformation role		X	Self	Self	OrganizationConfig	OrganizationConfig
MyDiagnostics role		X	Self	Self	OrganizationConfig	OrganizationConfig
MyDistributionGroupMembership role		X	MyGAL	MyGAL	None	None
MyDistributionGroups role		X	MyGAL	MyDistributionGroups	OrganizationConfig	None
MyProfileInformation role		X	Self	Self	OrganizationConfig	OrganizationConfig

MyRetentionPolicies role		X	Self	Self	OrganizationConfig	OrganizationConfig
MyTeamMailboxes role		X	Organization	Organization	OrganizationConfig	OrganizationConfig
MyTextMessaging role		X	Self	Self	OrganizationConfig	OrganizationConfig
MyVoiceMail role		X	Self	Self	OrganizationConfig	OrganizationConfig

View-only Organization Management

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The View-Only Organization Management management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Administrators who are members of the View-Only Organization Management role group can view the properties of any object in the Exchange organization.

This role is equivalent to the Exchange View-Only Administrators role in Microsoft Exchange Server 2007.

For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or universal security groups (USGs) that are members of this role group.

```
Get-RoleGroupMember "View-Only Organization Management"
```

For more information about the members of a role group, see View the members of a role group in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the “Create a role group” section in Manage role groups.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.
- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.

- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Monitoring role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
View-Only Configuration role	X		Organization	None	OrganizationConfig	None
View-Only Recipients role	X		Organization	None	OrganizationConfig	None

Recipient Management

Permissions > Understanding Role Based Access Control > Built-in role groups >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Recipient Management management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Administrators who are members of the Recipient Management role group have administrative

access to create or modify Exchange 2013 recipients within the Exchange 2013 organization.

This role group is equivalent to the Exchange Recipient Administrators role in Exchange Server 2007.

For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or universal security groups (USGs) that are members of this role group.

```
Get-RoleGroupMember "Recipient Management"
```

For more information about the members of a role group, see View the members of a role group in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the "Create a role group" section in Manage role groups.

If the creation of security principals in your organization, such as user accounts, is controlled by a specific group other than the Exchange administrators, you can create a role group and move the Mail Recipient Creation role and the Security Group Creation and Membership role to the new role group. Doing so prevents members of the Recipient Management role group from creating Active Directory objects. They can, however, continue to mail-enable the new Active Directory objects. For more information about split permissions, see Understanding split permissions.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.
- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Distribution Groups role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Mail Recipient Creation role	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Mail Recipients role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Message Tracking role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Migration role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Move Mailboxes role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Recipient Policies role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Team Mailboxes role	X		Organization	Organization	OrganizationConfig	OrganizationConfig

UM Management

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The UM Management management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Administrators who are members of the UM Management role group can manage features in the Exchange organization such as Unified Messaging (UM) service configuration, UM properties on mailboxes, UM prompts, and UM auto attendant configuration.

For more information about Unified Messaging, see Unified Messaging.

For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or universal security groups (USGs) that are members of this role group.

```
Get-RoleGroupMember "UM Management"
```

For more information about the members of a role group, see the “View the members of a role group” section in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the “Create a role group” section in Manage role groups.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.
- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
UM Mailboxes role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
UM Prompts role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Unified Messaging role	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Help Desk

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Help Desk management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Users who are members of the Help Desk role group can perform limited recipient management of Exchange 2013 recipients.

The Help Desk role group, by default, enables members to view and modify the Outlook Web App options of any user in the organization. These options might include modifying the user's display name, address, phone number, and so on. They don't include options that aren't available in Outlook Web App options, such as modifying the size of a mailbox or configuring the mailbox database on which a mailbox is located.

The members of this role group can only modify the Outlook Web App options that the user can modify. This means that if a user can modify his or her display name, a member of the Help Desk role group can also modify that user's display name. However, if another user isn't allowed to modify his or her display name, a member of the Help Desk role group can't modify that user's display name.

Caution:

The limitations on which Outlook Web App options a member of the Help Desk role group can modify are enforced by the Exchange Administration Center (EAC). If a member of the Help Desk role group has access to the Exchange Management Shell, he or she can modify any Outlook Web App option for any user. You should carefully consider who you make a member of the Help Desk role group and whether they should also be given access to the Shell.

The Help Desk role group doesn't enable any other tasks because there are so many different types of organizations. Instead, you can add management roles to this role group to create a Help Desk role group that matches the needs of your organization. For example, if you want members of the Help Desk role group to be able to manage mailboxes, mail contacts, and mail-enabled users, assign the Mail Recipients management role to this role group. For more information about how to add management roles to this role group, see the "Role Group Customization" section later in this topic.

For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group

members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or universal security groups (USGs) that are members of this role group.

```
Get-RoleGroupMember "Help Desk"
```

For more information about the members of a role group, see View the members of a role group in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the “Create a role group” section in Manage role groups.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.

- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
User Options role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
View-Only Recipients role	X		Organization	None	OrganizationConfig	None

Hygiene Management

Permissions > Understanding Role Based Access Control > Built-in role groups >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Hygiene Management management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Users who are members of the Hygiene Management role group can configure the anti-spam and

anti-malware features of Exchange 2013. Third-party programs that integrate with Exchange 2013 can add service accounts to this role group to grant those programs access to the cmdlets required to retrieve and configure the Exchange configuration.

For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or universal security groups (USGs) that are members of this role group.

```
Get-RoleGroupMember "Hygiene Management"
```

For more information about the members of a role group, see the “View the members of a role group” section in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the "Create a role group" section in Manage role groups.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.
- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
ApplicationImpersonation role	X		Organization	Organization	None	None
Receive Connectors role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Transport Agents role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Transport Hygiene role	X		Organization	Organization	OrganizationConfig	OrganizationConfig

View-Only Configuration role	X		Organization	None	OrganizationConfig	None
View-Only Recipients role	X		Organization	None	OrganizationConfig	None

Compliance Management

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Compliance Management management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Users who are members of the Compliance Management role group can configure and manage Exchange compliance configuration in accordance with their policies.

For more information about compliance features, see Messaging policy and compliance. For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or universal security groups (USGs) that are members of this role group.

```
Get-RoleGroupMember "Compliance Management"
```

For more information about the members of a role group, see View the members of a role group in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the “Create a role group” section in Manage role groups.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.
- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the

role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Data Loss Prevention role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Information Rights Management role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Retention Management role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
View-Only Audit Logs role	X		Organization	None	OrganizationConfig	None
View-Only Configuration role	X		Organization	None	OrganizationConfig	None
View-Only Recipients role	X		Organization	None	OrganizationConfig	None

Records Management

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Records Management management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Users who are members of the Records Management role group can configure compliance features, such as retention policy tags, message classifications, and transport rules.

For more information about compliance features, see Messaging policy and compliance. For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or universal security groups (USGs) that are members of this role group.

```
Get-RoleGroupMember "Records Management"
```

For more information about the members of a role group, see View the members of a role group in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the "Create a role group" section in Manage role groups.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.
- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Audit Logs role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Journaling	X		Organization	Organization	OrganizationConfig	OrganizationConfig

role						
Message Tracking role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Retention Management role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Transport Rules role	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Discovery Management

Permissions > Understanding Role Based Access Control > Built-in role groups >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Discovery Management management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Administrators or users who are members of the Discovery Management role group can perform searches of mailboxes in the Exchange organization for data that meets specific criteria and can also configure litigation holds on mailboxes. For more information, see In-Place eDiscovery.

Important:
The Organization Management role group doesn't, by default, enable the discovery search feature for users or universal security groups (USGs) that are members of that role group. Members of the Organization Management role group must either be made members of this role group, or the Mailbox Search role listed later in this topic must be manually assigned to the Organization Management role group. For information about how to assign a role to a role group, see Manage role groups.

For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or USGs that are members of this role group.

```
Get-RoleGroupMember "Discovery Management"
```

For more information about the members of a role group, see View the members of a role group in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the “Create a role group” section in Manage role groups.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries

made available by the associated management role.

- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Legal Hold role	X		Organization	Organization	OrganizationConfig	None
Mailbox Search role	X		Organization	Organization	None	None

Public Folder Management

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Public Folder Management management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Administrators who are members of the Public Folder Management role group can manage public

folders on servers running Exchange 2013.

For more information about public folders, see Public folders. For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or universal security groups (USGs) that are members of this role group.

```
Get-RoleGroupMember "Public Folder Management"
```

For more information about the members of a role group, see View the members of a role group in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group

to the new role group, see the "Create a role group" section in Manage role groups.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.
- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Mail Enabled Public Folders role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Public Folders role	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Server Management

Permissions > Understanding Role Based Access Control > Built-in role groups >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Server Management management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Administrators who are members of this role group can configure server-specific configuration of transport client access, and mailbox features such as database copies, certificates, transport queues and Send connectors, virtual directories, and client access protocols.

This role group is similar to the Exchange Server Administrators role in Microsoft Exchange Server 2007. It grants access to manage the configuration of physical servers. However, unlike the Exchange Server Administrators role in Exchange 2007, which provided access only to a local server running Exchange 2007, the Server Management role group enables access to view and configure all Exchange Server 2010 servers in the organization.

If you want to allow administrators to manage only specific servers in your organization, you can change the management scopes that are applied to this role group. Alternatively, you can create a role group, based on the Server Management role group, and customize the management scopes on the new role group. For more information, see the "Role Group Customization" section later in this topic.

For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or universal security groups (USGs) that are members of this role group.

```
Get-RoleGroupMember "Server Management"
```

For more information about the members of a role group, see View the members of a role group in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in

the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the “Create a role group” section in Manage role groups.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.
- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Database Copies role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Databases role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Connectors role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Server Certificates role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Servers role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Exchange Virtual Directories role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Monitoring role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
POP3 and IMAP4 Protocols role	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Receive Connectors role	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Transport Queues role	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Delegated Setup

Permissions > Understanding Role Based Access Control > Built-in role groups >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Delegated Setup management role group is one of several built-in role groups that make up the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Role groups are assigned one or more management roles that contain the permissions required to perform a given set of tasks. The members of a role group are granted access to the management roles assigned to the role group. For more information about role groups, see Understanding management role groups.

Administrators who are members of the Delegated Setup role group can deploy servers running Exchange 2013 that have been previously provisioned by a member of the Organization Management role group.

Members of the Delegated Setup role group can only deploy Exchange 2013 servers. They can't manage the server after it's been deployed. To manage a server after it's been deployed, a user must be a member of the Server Management role group.

For more information about RBAC, see Understanding Role Based Access Control.

Role group membership

If you want to add or remove members to or from this role group, see Manage role group members.

By default, only members of the Organization Management role group can add or remove members from this role group. For more information about how to add additional role group delegates, see the "Add or remove a role group delegate" section in Manage role groups.

You can use the following command to view a list of users or universal security groups (USGs) that are members of this role group.

```
Get-RoleGroupMember "Delegated Setup"
```

For more information about the members of a role group, see View the members of a role group in Manage role group members.

Role group customization

This role group is assigned management roles by default. The roles that are included are listed in the "Management Roles Assigned to this Role Group" section. You can add or remove role assignments to or from this role group to match the needs of your organization.

The role groups provided with Exchange 2013 are designed to match more granular tasks. By assigning roles to a role group, you enable the members of that role group to perform the tasks associated with the role. For example, the Journaling role enables the management of the Journaling agent and journaling rules. For more information about how roles are assigned to role groups, see Understanding management role assignments.

The roles assigned to this role group are given default management scopes. Management scopes determine what Exchange objects can be viewed or modified by the members of a role group. You can change the scopes associated with assignments between roles and role groups. For example, you might want to do this if you only want members of a role group to be able to change recipients that are under a specific organizational unit or in a specific location. For more information about management scopes, see Understanding management role scopes.

For more information about how to customize this role group, see the following topics:

- Manage role groups
- Manage role group members

If you want to create a role group and assign some of the roles that are assigned to this role group to the new role group, see the “Create a role group” section in Manage role groups.

Additional permissions

The permissions granted to members of the Delegated Setup role group are primarily determined by the management roles assigned to the role group. However, not all tasks that you need to perform are covered by management roles. This is because some tasks occur outside of the Exchange management tools and therefore the RBAC permissions model doesn't apply. For these tasks, permissions are provided by adding the Delegated Setup role group to the access control lists (ACLs) of certain Active Directory objects.

The following task is granted permissions by way of ACLs on Active Directory objects and not by management roles assigned to the Delegated Setup role group:

- Deployment of servers that have been previously provisioned by a member of the Organization Management role group.

Management roles assigned to this role group

The following table lists all the management roles that are assigned to this role group and the following attributes of each role assignment:

- **Regular assignment** Enables members of the role group to access the management role entries made available by the associated management role.
- **Delegating assignment** Gives members of the role group the ability to assign the specified role to other role groups, role assignment policies, users, or USGs.
- **Recipient read scope** Determines what recipient objects members of the role group are allowed to read from Active Directory.
- **Recipient write scope** Determines what recipient objects members of the role group are allowed to modify in Active Directory.
- **Configuration read scope** Determines what configuration and server objects members of the role group are allowed to read from Active Directory.
- **Configuration write scope** Determines what organizational and server objects members of the role group are allowed to modify in Active Directory.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Management roles assigned to this role group

Management role	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
View-Only Configuration role	X		Organization	None	OrganizationConfig	None

Built-in management roles

Exchange Server 2013 > Permissions > Understanding Role Based Access Control >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-08-21*

Microsoft Exchange Server 2013 includes many management roles by default. The following roles are assigned to management role groups or management role assignment policies in various combinations that grant permissions to manage and use the features provided by Exchange 2013. For more information about roles, see Understanding management roles.

Active Directory Permissions role

Address Lists role

ApplicationImpersonation role

ArchiveApplication role

Audit Logs role

Cmdlet Extension Agents role

Data Loss Prevention role

Database Availability Groups role

Database Copies role

Databases role

Disaster Recovery role

Distribution Groups role

Edge Subscriptions role

E-Mail Address Policies role

Exchange Connectors role

Exchange Server Certificates role

Exchange Servers role

Exchange Virtual Directories role

Federated Sharing role

Information Rights Management role

Journaling role

Legal Hold role

LegalHoldApplication role

Mail Enabled Public Folders role

Mail Recipient Creation role

Mail Recipients role

Mail Tips role

Mailbox Import Export role

Mailbox Search role

MailboxSearchApplication role

Message Tracking role

Migration role

Monitoring role

Move Mailboxes role

My Custom Apps role

My Marketplace Apps role

MyAddressInformation role

MyBaseOptions role

MyContactInformation role

MyDiagnostics role

MyDisplayName role

MyDistributionGroupMembership role

MyDistributionGroups role

MyMobileInformation role

MyName role

MyPersonalInformation role

MyProfileInformation role

MyRetentionPolicies role

MyTeamMailboxes role

MyTextMessaging role

MyVoiceMail role

OfficeExtensionApplication role

Org Custom Apps role

Org Marketplace Apps role

Organization Client Access role

Organization Configuration role

Organization Transport Settings role

POP3 and IMAP4 Protocols role

Public Folders role

Receive Connectors role

Recipient Policies role

Remote and Accepted Domains role

Reset Password role

Retention Management rolet

Role Management role

Security Group Creation and Membership role

Send Connectors role

Support Diagnostics role

Team Mailboxes role

TeamMailboxLifecycleApplication role

Transport Agents role

Transport Hygiene role

Transport Queues role

Transport Rules role

UM Mailboxes role

UM Prompts role

Unified Messaging role

Unscoped Role Management role

User Options role

UserApplication role

View-Only Audit Logs role

View-Only Configuration role

View-Only Recipients role

Active Directory Permissions role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Active Directory Permissions` management role enables administrators to configure Active Directory permissions in an organization. Some features that use Active Directory permissions or an access control list (ACL) include transport Receive and Send connectors, and Send As and Send on behalf of permissions for mailboxes.

Note:

Permissions set directly on Active Directory objects may not be enforced through Role Based Access Control (RBAC).

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide

access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:
The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Address Lists role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Address Lists` management role enables administrators to create, modify, view, and remove address lists, global address lists (GALs), address book policies, and offline address lists (OABs) in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC)

permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding

management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.

- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and

from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

ApplicationImpersonation role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-17*

The `ApplicationImpersonation` management role enables applications to impersonate users in an

organization to perform tasks on behalf of the user.

Important:

A process or application that's a member of the `ApplicationImpersonation` role can access the contents of a user's mailbox and act on behalf of that user, even if the user's account is disabled. This might let users access their mailboxes if you have applications, like Blackberry Enterprise Server, that use the `ApplicationImpersonation` role. Third-party products that don't use the `ApplicationImpersonation` role and instead use Exchange ActiveSync can't access a mailbox after its user account has been disabled.

To prevent an application that uses the `ApplicationImpersonation` role from accessing a mailbox or performing tasks on its behalf after its user account has been disabled, do one or more of the following:

- Disable or remove the user in the third-party application.
- Delete the mailbox.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and

server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Hygiene Management	X		Organization	Organization	None	None
Organization Management		X	Organization	Organization	None	None

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG

4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

ArchiveApplication role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `ArchiveApplication` management role enables partner applications to archive items in user mailboxes.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope

allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.

- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups

- Remove a role from a user or USG

4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Audit Logs role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Audit Logs` management role enables administrators to configure the administrator audit log in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can

be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last

assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.

- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Records Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see

the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Cmdlet Extension Agents role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Cmdlet Extension Agents` management role enables administrators to enable, disable, and set the priority of cmdlet extension agents in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group,

user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs.

However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether

the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.

2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Data Loss Prevention role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Data Loss Prevention management role enables administrators to create and manage data loss prevention (DLP) policies and the rules within them, which can affect mail delivery for an entire organization. Furthermore, this role provides administrators with the ability to configure Policy Tips that appear in email clients and manage DLP policy violation reports. This DLP role also enables access to Microsoft Exchange transport rules. For more information about transport rules in Exchange, see the following topics:

- Transport Rules role
- What's new for transport rules

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a

management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is

assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Compliance Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and

from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Database Availability Groups role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Database Availability Groups` management role enables administrators to manage database

availability groups in an organization. Administrators assigned this role either directly or indirectly are the highest level administrators responsible for the high availability configuration in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment

- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Database Copies role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Database Copies` management role enables administrators to add, remove, suspend, resume, view, and update database copies on individual servers.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more

information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Additional scope considerations

The **Set-MailboxDatabaseCopy** and **Remove-MailboxDatabaseCopy** cmdlets, which are included with this role, require that the database you want to configure or remove must be within the database scope and the database must reside on a server that's within the server scope. For more information about scopes, see Understanding management role scopes.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Server Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the

following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Databases role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Databases` management role enables administrators to create, manage, mount, and dismount mailbox databases on individual servers.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role

assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups,

users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Additional scope considerations

The **Move-DatabasePath** cmdlet, which is included with this role, requires that the database you want to configure must be within the database scope and the database must reside on a server that's within the server scope.

Also, the **Remove-MailboxDatabase** cmdlet, which is also included with this role, requires that the database you want to remove must either be within the database scope or the database must reside on a server that's within the server scope. This means you control who can remove mailbox databases using either database or server scopes.

For more information, see Understanding management role scopes.

Default Management Role Assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Server Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Disaster Recovery role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Disaster Recovery` management role enables administrators to restore mailboxes and mailbox databases, create mailbox databases, and perform datacenter switchovers and switchbacks for database availability groups in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide

access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:
The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Distribution Groups role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Distribution Groups` management role enables administrators to create, modify, view, and remove distribution groups, and add or remove distribution group members in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to

one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role

assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Recipient Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters

to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Edge Subscriptions role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Edge Subscriptions` management role enables administrators to manage edge synchronization and subscription configuration between Microsoft Exchange Server 2010 Edge Transport servers and Microsoft Exchange Server 2013 Mailbox servers in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more

information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization	X	X	`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management						

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

E-Mail Address Policies role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `E-Mail Address Policies` management role enables administrators to manage email address policies in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and

role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular	Delegating	Recipient	Recipient	Configurati	Configurati

	assignment	assignment	read scope	write scope	on read scope	on write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee.

For more information, see Delegate role assignments.

Exchange Connectors role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-11-19*

The `Exchange Connectors` management role enables administrators to create, modify, view, and remove delivery agent connectors.

This role can't be used to manage Send and Receive connectors. To manage Send and Receive connectors, use the Send Connectors and Receive Connectors roles. For more information, see:

- Send Connectors role
- Receive Connectors role

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The

permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.

- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Server Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role

assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Exchange Server Certificates role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Exchange Server Certificates` management role enables administrators to create, import, export, and manage Exchange server certificates on individual servers.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:

There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign

this role to role groups, users, or USGs.

- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Server Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role

because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Exchange Servers role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Exchange Servers` management role enables administrators to do the following on individual servers:

- Add and remove database availability groups and configure database copies
- Enable, disable and configure Unified Messaging services
- Modify transport configuration on Mailbox and Client Access servers
- Enable and disable Microsoft Outlook Anywhere on Client Access servers
- Modify Mailbox and Client Access server configuration
- Modify Outlook Anywhere configuration on Client Access servers
- Modify content filtering configuration on Mailbox servers
- Modify general Exchange server configuration
- Modify server monitoring configuration
- View the configuration for each server role

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide

access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Server Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and

from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Exchange Virtual Directories role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Exchange Virtual Directories` management role enables administrators to manage Microsoft

Office Outlook Web App, Microsoft ActiveSync, offline address books (OABs), Autodiscover, Windows PowerShell, and Web administration interface virtual directories on individual servers.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments.

Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment**

cmdlet. For more information, see Change a role assignment.

- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Server Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Federated Sharing role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Federated Sharing` management role enables administrators to manage cross-forest and cross-organization sharing in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization	X	X	`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management						

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Information Rights Management role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Information Rights Management` management role enables administrators to manage the Information Rights Management (IRM) features of Exchange in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and

role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular	Delegating	Recipient	Recipient	Configurati	Configurati

	assignment	assignment	read scope	write scope	on read scope	on write scope
Compliance Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Journaling role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Journaling` management role enables administrators to create, modify, enable, disable, view, and remove journal rules in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG

- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.

- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Records Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups

- Remove a role from a user or USG

4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Legal Hold role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Legal Hold` management role enables administrators to configure whether data within a mailbox should be retained for litigation purposes in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can

be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last

assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Additional scope considerations

In addition to recipient scopes, the **Enable-Mailbox** cmdlet, which is included with this role, is also scoped using database configuration scopes. Database configuration scopes control which databases the cmdlet can create new mailboxes on. The database where you want to create a mailbox must be within the database scope. This applies both when you specify a database using the *Database* parameter on the **Enable-Mailbox** cmdlet, or if you allow automatic mailbox distribution to select the database for you. For more information, see Understanding management role scopes.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Discovery Management	X		Organization	Organization	OrganizationConfig	None
Organization Management	X	X	Organization	Organization	OrganizationConfig	None

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

LegalHoldApplication role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `LegalHoldApplication` management role enables partner applications to set and view the legal hold status of a mailbox.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox

databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:
The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Mail Enabled Public Folders role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Mail Enabled Public Folders` management role enables administrators to configure whether individual public folders are mail-enabled or mail-disabled in an organization.

This role enables you to manage only the e-mail properties of public folders. It doesn't enable you to manage public folder properties that aren't related to e-mail. To manage public folder

properties that aren't related to e-mail, use the `Public Folders` role. For more information, see Public Folders role.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments.

Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment**

cmdlet. For more information, see Change a role assignment.

- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Public Folder Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Mail Recipient Creation role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Mail Recipient Creation` management role enables administrators to create mailboxes, mail users, mail contacts, distribution groups, and dynamic distribution groups in an organization. This role can be combined with the `Mail Recipients` role to enable the creation and management of recipients. For more information, see Mail Recipients role.

This role doesn't enable you to mail-enable public folders. To mail-enable public folders, the `Mail Enabled Public Folders` role must be used. For more information, see Mail Enabled Public Folders role.

If your organization maintains a Role Based Access Control (RBAC) split permissions model where recipient creation is performed by a different group than those who perform recipient management, assign the `Mail Recipient Creation` role to the management role group that performs recipient creation, and the `Mail Recipients` role to the role group that performs recipient management.

If your organization has enabled Active Directory split permissions, all non-delegating management role assignments to this management role were removed. When Active Directory split permissions is enabled, only Active Directory administrators using Active Directory management tools can create new security principals such as users and security groups.

For more information about RBAC and Active Directory split permissions, see Understanding split permissions.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is

assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps

prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Additional scope considerations

In addition to recipient scopes, the **New-Mailbox** cmdlet, which is included with this role, is also scoped using database configuration scopes. Database configuration scopes control which databases the cmdlet can create new mailboxes on. The database where you want to create a

mailbox must be within the database scope. This condition applies when you specify a database using the *Database* parameter on the **New-Mailbox** cmdlet or if you allow automatic mailbox distribution to select the database for you. For more information, see Understanding management role scopes.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Recipient Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Mail Recipients role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Mail Recipients` management role enables administrators to manage existing mailboxes, mail users, and mail contacts in an organization. This role can't create these recipients. Use the `Mail Recipient Creation` role to create them.

This role type doesn't enable you to manage mail-enabled public folders or distribution groups.

Use the following roles to manage these objects:

- Mail Enabled Public Folders role
- Distribution Groups role

If your organization has a split permissions model where recipient creation and management are performed by different groups, assign the `Mail Recipient Creation` role to the group that performs recipient creation and the `Mail Recipients` role to the group that performs recipient management. For more information, see the following topics:

- Mail Recipient Creation role
- Mail Recipients role
- Understanding split permissions

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and

role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Additional scope considerations

In addition to recipient scopes, the **Connect-Mailbox** and **Enable-Mailbox** cmdlets, which are included with this role, are also scoped using database configuration scopes. Database configuration scopes control which databases the cmdlets can create new mailboxes on. The database where you want to create a mailbox must be within the database scope. This condition applies when you specify a database using the *Database* parameter on either cmdlet or if you allow automatic mailbox distribution to select the database for you. For more information, see Understanding management role scopes.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.

- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Recipient Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see

the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Mail Tips role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Mail Tips management role enables administrators to manage mail tips in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role

assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a

role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to

each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and

Remove-ManagementRoleEntry cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Mailbox Import Export role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Mailbox Import Export` management role enables administrators to import and export mailbox content and to purge unwanted content from a mailbox.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more

information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Mailbox Search role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Mailbox Search` management role enables administrators to search the content of one or more mailboxes in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions

without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Discovery Management	X		Organization	Organization	None	None
Organization Management		X	Organization	Organization	None	None

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MailboxSearchApplication role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MailboxSearchApplication` management role enables partner applications to search mailboxes.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox

databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Message Tracking role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Message Tracking` management role enables administrators to track messages in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to

one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role

assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Recipient Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Records Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Migration role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Migration` management role enables administrators to migrate mailboxes and mailbox content into or out of a server.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization	X	X	`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management						
Recipient Management	X		`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee.

For more information, see Delegate role assignments.

Monitoring role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Monitoring management role enables administrators to monitor Exchange services and component availability in an organization. This role can also be used with service accounts used by monitoring applications to collect information about the state of servers running Exchange.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and

server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
View-only Organization Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Server Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role

3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Move Mailboxes role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Move Mailboxes` management role enables administrators to move mailboxes between servers in an organization and between servers in the local organization and another organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions

granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Additional scope considerations

In addition to recipient scopes, the **New-MoveRequest** cmdlet, which is included with this role, is also scoped using database configuration scopes. Database configuration scopes control which databases the cmdlet can move mailboxes to. The database where you want to move a mailbox must be within the database scope. This condition applies when you specify a database using the

TargetDatabase parameter on the **New-MoveRequest** cmdlet or if you allow automatic mailbox distribution to select the database for you. For more information, see Understanding management role scopes.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Recipient Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

My Custom Apps role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-08*

The My Custom Apps role enables individual users to add apps from a file or a URL.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal

security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.

- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	`Self`	`Self`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG

4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

My Marketplace Apps role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-08*

The My Marketplace Apps management role enables individual users to add apps from the Microsoft Office Store.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Default role assignment policy. For more information, see Understanding	X		`Self`	`Self`	`OrganizationConfig`	`OrganizationConfig`

management role assignment policies.						
Organization Management		X	Self	Self	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups

- Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyAddressInformation role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyAddressInformation` management role enables individual users to view and modify their street address and work telephone and fax numbers. This is a custom role created from the MyContactInformation role parent role.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role.

This role is a specific type of role called a custom role. A custom role is one that's created, or derived, from a parent role. It contains a subset of management role entries that exist on the parent role. This role is provided to enable you to control, with a greater level of granularity, the information you allow end users to modify on their own mailboxes. Unlike other built-in roles, custom roles, including this one, can be deleted. If you won't use this role, it can be deleted.

For more information about built-in and custom management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee

is granted the combination of all the permissions granted by all the assigned roles.

Note:
You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, its parent role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- “Add or remove a role to or from a role group” section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role doesn't have any default role assignments. It's provided in case you want control, at a more granular level, of what end-user information you allow your users to modify. For more information about assigning this role to a role assignment policy, see the "Adding or Removing Role Assignments" section.

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry

- Remove a role entry from a role

3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyBaseOptions role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyBaseOptions` management role enables individual users to view and modify the basic configuration of their own mailbox and associated settings.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee

is granted the combination of all the permissions granted by all the assigned roles.

Note:
You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Default role assignment policy. For more	X		Self	Self	OrganizationConfig	OrganizationConfig

information, see Understanding management role assignment policies.						
Organization Management		X	Self	Self	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Remove a role from a user or USG

4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyContactInformation role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyContactInformation` management role enables individual users to modify their contact information, including address and phone numbers.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However

user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups

- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Default role assignment policy. For more information, see Understandi	X		`Self`	`Self`	`OrganizationConfig`	`OrganizationConfig`

ng management role assignment policies.						
Organization Management		X	Self	Self	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyDiagnostics role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyDiagnostics` management role enables individual users to perform basic diagnostics on their mailbox such as retrieving calendar diagnostic information.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups,

USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	Self	Self	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more

information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyDisplayName role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyDisplayName` management role enables individual users to view and modify their display name. This is a custom role created from the MyProfileInformation role parent role.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role.

This role is a specific type of role called a custom role. A custom role is one that's created, or derived, from a parent role. It contains a subset of management role entries that exist on the parent role. This role is provided to enable you to control, with a greater level of granularity, the information you allow end users to modify on their own mailboxes. Unlike other built-in roles, custom roles, including this one, can be deleted. If you won't use this role, it can be deleted.

For more information about built-in and custom management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, its parent role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- “Add or remove a role to or from a role group” section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role doesn't have any default role assignments. It's provided in case you want control, at a more granular level, of what end-user information you allow your users to modify. For more information about assigning this role to a role assignment policy, see the "Adding or Removing Role Assignments" section.

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyDistributionGroupMembership role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyDistributionGroupMembership` management role enables individual users to view and modify their membership in distribution groups in an organization, provided that those distribution groups allow manipulation of group membership.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to

each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Default role assignment policy. For more information, see Understanding management role assignment policies.	X		MyGAL	MyGAL	None	None
Organization Management		X	MyGAL	MyGAL	None	None

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters

to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyDistributionGroups role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyDistributionGroups` management role enables individual users to create, modify, and view distribution groups, and to modify, view, remove, and add members to distribution groups they own.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:
You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign

this role to role groups, users, or USGs.

- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	MyGAL	MyDistributionGroups	OrganizationConfig	None

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry

- Remove a role entry from a role

3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyMobileInformation role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyMobileInformation` management role enables individual users to view and modify their mobile telephone and pager numbers. This is a custom role created from the MyContactInformation role parent role.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role.

This role is a specific type of role called a custom role. A custom role is one that's created, or derived, from a parent role. It contains a subset of management role entries that exist on the parent role. This role is provided to enable you to control, with a greater level of granularity, the information you allow end users to modify on their own mailboxes. Unlike other built-in roles, custom roles, including this one, can be deleted. If you won't use this role, it can be deleted.

For more information about built-in and custom management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:
You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, its parent role must be assigned to a role group you're a

member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- “Add or remove a role to or from a role group” section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role doesn't have any default role assignments. It's provided in case you want control, at a more granular level, of what end-user information you allow your users to modify. For more information about assigning this role to a role assignment policy, see the "Adding or Removing Role Assignments" section.

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:
The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyName role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyName` management role enables individual users to view and modify their full name and their notes field. This is a custom role created from the MyProfileInformation role parent role.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role.

This role is a specific type of role called a custom role. A custom role is one that's created, or

derived, from a parent role. It contains a subset of management role entries that exist on the parent role. This role is provided to enable you to control, with a greater level of granularity, the information you allow end users to modify on their own mailboxes. Unlike other built-in roles, custom roles, including this one, can be deleted. If you won't use this role, it can be deleted.

For more information about built-in and custom management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, its parent role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- “Add or remove a role to or from a role group” section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role doesn't have any default role assignments. It's provided in case you want control, at a more granular level, of what end-user information you allow your users to modify. For more information about assigning this role to a role assignment policy, see the "Adding or Removing Role Assignments" section.

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:
The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyPersonalInformation role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyPersonalInformation` management role enables individual users to view and modify their website, address, and home telephone number. This is a custom role created from the MyContactInformation role parent role.

This management role is one of several built-in roles in the Role Based Access Control (RBAC)

permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role.

This role is a specific type of role called a custom role. A custom role is one that's created, or derived, from a parent role. It contains a subset of management role entries that exist on the parent role. This role is provided to enable you to control, with a greater level of granularity, the information you allow end users to modify on their own mailboxes. Unlike other built-in roles, custom roles, including this one, can be deleted. If you won't use this role, it can be deleted.

For more information about built-in and custom management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, its parent role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- “Add or remove a role to or from a role group” section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role doesn't have any default role assignments. It's provided in case you want control, at a more granular level, of what end-user information you allow your users to modify. For more information about assigning this role to a role assignment policy, see the "Adding or Removing Role Assignments" section.

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyProfileInformation role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyProfileInformation` management role enables individual users to modify their name.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	Self	Self	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role

because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyRetentionPolicies role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyRetentionPolicies` management role enables individual users to view their retention tags, and to view and modify their retention tag settings and defaults.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and

role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	`Self`	`Self`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyTeamMailboxes role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyTeamMailboxes` management role enables individual users to create site mailboxes and connect them to Microsoft SharePoint sites.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Default role assignment policy. For more information, see Understanding management role assignment policies.	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Management		X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and

from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyTextMessaging role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyTextMessaging` management role enables individual users to create, view, and modify their

text messaging settings.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee.

Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role

assignee is allowed to read from Active Directory.

- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Default Role Assignment Policy For more information, see Understanding management role assignment policies.	X		`Self`	`Self`	`OrganizationConfig`	`OrganizationConfig`
Organization Management		X	`Self`	`Self`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a

child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

MyVoiceMail role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `MyVoiceMail` management role enables individual users to view and modify their voice mail settings.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal

security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, such as a role assignment policy. This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

Note:

You can also assign this management role to a role group, USG, or directly to a user. However user-focused roles are most effective when used with role assignment policies.

This user-focused role has implicit scopes that can't be modified. Therefore, you shouldn't add custom scopes to role assignments that assign this role to role assignment policies, role groups, USGs, or users.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role may be assigned to one or more role assignment policies by default. For more information, see the "Default Management Role Assignments" section.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role assignment policies, or you can create role assignment policies and assign this role to them.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from the default role assignment policy, role assignment policies and role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.

- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group or assignment policy	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Default role assignment policy. For more information, see Understanding management role assignment policies.	X		Self	Self	OrganizationConfig	OrganizationConfig
Organization Management		X	Self	Self	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:
The following information enables you to perform advanced management of permissions.

Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

OfficeExtensionApplication role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `OfficeExtensionApplication` management role enables Microsoft Office extension applications to access user mailboxes.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a

management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is

assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	`Self`	`Self`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:
The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Org Custom Apps role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-08*

The Org Custom Apps management role enables administrators to view and modify their organization's apps, and to add custom apps from a file or URL.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to

one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role

assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize

individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Org Marketplace Apps role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-15*

The Org Marketplace Apps management role enables administrators to view and modify their organization's apps, and to add apps from the Microsoft Office Store.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role

assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For

more information, see Change a role assignment.

- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also

been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Organization Client Access role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Organization Client Access` management role enables administrators to manage Client Access server settings in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments.

Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment**

cmdlet. For more information, see Change a role assignment.

- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Organization Configuration role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Organization Configuration` management role enables administrators to manage organization-wide settings. Organization configuration that can be controlled with this role includes the following and more:

- Whether MailTips are enabled or disabled for the organization
- URL for the managed folder home page
- Microsoft Exchange recipient SMTP address and alternate email addresses
- Resource mailbox property schema configuration
- Help URLs for the Exchange Administration Center and Microsoft Office Outlook Web App

This role type doesn't include the permissions included in the `Organization Client Access` or `Organization Transport Settings` roles.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and

server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Organization Transport Settings role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Organization Transport Settings` management role enables administrators to manage organization-wide transport settings, such as system messages, Active Directory site configuration, and transport configuration settings for the whole Exchange organization.

This role doesn't enable you to create or manage Send or Receive connectors, queues, hygiene, agents, remote and accepted domains, or rules. To create or manage each of the transport features, you must be assigned one or more of the following roles:

- Receive Connectors role
- Send Connectors role
- Transport Queues role
- Transport Hygiene role
- Transport Agents role
- Remote and Accepted Domains role
- Transport Rules role

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group,

user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs.

However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether

the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.

2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

POP3 and IMAP4 Protocols role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `POP3 and IMAP4 Protocols` management role enables administrators to manage POP3 and IMAP4 configuration, such as authentication and connection settings, on individual servers.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more

information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Server Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Public Folders role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Public Folders` management role enables administrators to manage public folders in an organization.

This role doesn't enable you to manage whether public folders are mail-enabled. To mail-enable or disable a public folder, you must be assigned a role associated with the `Mail Enabled Public Folders` role. For more information, see Mail Enabled Public Folders role.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to

one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role

assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Public Folder Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters

to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Receive Connectors role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Receive Connectors` management role enables administrators to manage transport Receive connector configuration, such as size limits on an individual server.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment

- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Hygiene Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Organization	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management						
Server Management	X		`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee.

For more information, see Delegate role assignments.

Recipient Policies role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Recipient Policies` management role enables administrators to manage recipient policies, such as throttling policies, Microsoft Office Outlook Web App mailbox policies, and mobile device policies, in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and

server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Recipient Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG

4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Remote and Accepted Domains role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Remote and Accepted Domains` management role enables administrators to manage remote and accepted domains in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope

allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.

- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups

- Remove a role from a user or USG

4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Reset Password role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The Reset Password management role enables individual users to reset their own passwords and administrators to reset users' passwords in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can

be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last

assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.

- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role

assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Retention Management role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Retention Management` management role enables administrators to manage retention policies in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:

There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign

this role to role groups, users, or USGs.

- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Compliance Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Records Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Role Management role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Role Management` management role enables administrators to manage management role groups; role assignment policies and management roles; and role entries, assignments, and scopes in an organization.

Users assigned this role can override the role group managed by property, configure any role group, and add or remove members to or from any role group.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role

entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign

this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in

effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Security Group Creation and Membership role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Security Group Creation and Membership` management role enables administrators to create and manage universal security groups (USGs) and their memberships in an organization.

If your organization maintains a Role Based Access Control (RBAC) split permissions model where USG creation and management is performed by a different group other than those who manage servers running Exchange, assign this role to that group.

If your organization has enabled Active Directory split permissions, all non-delegating management role assignments to this management role were removed. When Active Directory split permissions is enabled, only Active Directory administrators using Active Directory management tools can create new security principals such as users and security groups.

For more information, see Understanding split permissions.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more

information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization	X	X	`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management						

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Send Connectors role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Send Connectors` management role enables administrators to manage transport Send connectors in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and

role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular	Delegating	Recipient	Recipient	Configurati	Configurati

	assignment	assignment	read scope	write scope	on read scope	on write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee.

For more information, see Delegate role assignments.

Support Diagnostics role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Support Diagnostics` management role enables administrators to perform advanced diagnostics under the direction of Microsoft Customer Service and Support in an organization.

Caution:

This role grants permissions to cmdlets and scripts that should only be used under the direction of Customer Service and Support.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role

assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups

- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and

server objects the role assignee is allowed to read from Active Directory.

- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:

- "Add or remove a role to or from a role group" section in Manage role groups
- Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Team Mailboxes role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Team Mailboxes` management role enables administrators to define one or more site mailbox provisioning policies and manage site mailboxes in the organization. Administrators assigned this role can manage site mailboxes they don't own.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role

assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups,

users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role

assignee is allowed to modify in Active Directory.

- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Recipient Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role

3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

TeamMailboxLifecycleApplication role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `TeamMailboxLifecycleApplication` management role enables partner applications to update site mailbox lifecycle states.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions

granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.

- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	`Self`	`Self`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry

- Remove a role entry from a role

3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Transport Agents role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Transport Agents` management role enables administrators to manage transport agents in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is

assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps

prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Hygiene Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Transport Hygiene role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Transport Hygiene` management role enables administrators to manage anti-spam and anti-malware features in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more

information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Hygiene Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Transport Queues role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Transport Queues` management role enables administrators to manage message queues on an individual transport server.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox

databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Server Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and

from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Transport Rules role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Transport Rules` management role enables administrators to manage transport rules in an

organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee.

Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.

- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
Records Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

UM Mailboxes role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `UM Mailboxes` management role enables administrators to manage the Unified Messaging configuration of mailboxes and other recipients in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization	X	X	`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management						
UM Management	X		`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee.

For more information, see Delegate role assignments.

UM Prompts role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `UM Prompts` management role enables administrators to create and manage custom Unified Messaging voice prompts in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and

server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
UM Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG

4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Unified Messaging role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Unified Messaging` role enables administrators to manage Unified Messaging (UM) services in an organization.

This role doesn't enable you to manage UM-specific mailbox configuration or UM prompts. To manage UM-specific mailbox configuration, use roles associated with the `UM Mailboxes` role. To manage UM prompts, use the roles associated with the `UM Prompts` role. For more information, see:

- UM Mailboxes role
- UM Prompts role

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is

assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps

prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig
UM Management	X		Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Unscoped Role Management role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `Unscoped Role Management` management role enables administrators to create and manage unscoped top-level management roles in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more

information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

User Options role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `User Options` management role enables administrators to view the Outlook Web App options of a user in an organization. This role can be used to help diagnose configuration problems.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions

without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Help Desk	X		Organization	Organization	OrganizationConfig	OrganizationConfig
Organization Management	X	X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

UserApplication role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `UserApplication` management role enables partner applications to act on behalf of end users.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a

management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is

assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization Management		X	Organization	Organization	OrganizationConfig	OrganizationConfig

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:
The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

View-Only Audit Logs role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `View-Only Audit Logs` management role enables administrators and specialist users to search the administrator audit logs in an organization.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to

one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role

assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Compliance Management	X		Organization	None	OrganizationConfig	None
Organization Management	X	X	Organization	None	OrganizationConfig	None

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters

to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

View-Only Configuration role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `View-Only Configuration` management role enables administrators to view all the non-recipient Exchange configuration settings in an organization. Examples of configuration that are viewable are server configuration, transport configuration, database configuration, and organization-wide configuration.

This role can be combined with roles associated with the `View-Only Recipients` role to create a role group that can view every object in an organization. For more information, see View-Only Recipients role.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the

scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope

Compliance Management	X		Organization	None	OrganizationConfig	None
Delegated Setup	X		Organization	None	OrganizationConfig	None
Hygiene Management	X		Organization	None	OrganizationConfig	None
Organization Management	X	X	Organization	None	OrganizationConfig	None
View-only Organization Management	X		Organization	None	OrganizationConfig	None

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry

- Remove a role entry from a role

3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:
If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

View-Only Recipients role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `View-Only Recipients` management role enables administrators to view the configuration of recipients, such as mailboxes, mail users, mail contacts, distribution groups, and dynamic distribution groups.

This role can be combined with roles associated with the `View-Only Configuration` role to create a role group that can view every object in the organization. For more information, see View-Only Configuration role.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more

information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Compliance Management	X		Organization	None	OrganizationConfig	None
Help Desk	X		Organization	None	OrganizationConfig	None
Hygiene Management	X		Organization	None	OrganizationConfig	None
Organization Management	X	X	Organization	None	OrganizationConfig	None
View-only Organization Management	X		Organization	None	OrganizationConfig	None

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters

to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - "Add or remove a role to or from a role group" section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

WorkloadManagement role

Permissions > Understanding Role Based Access Control > Built-in management roles >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

The `WorkloadManagement` management role enables administrators to manage workload management policies. Administrators can configure resource health definitions, workload classifications, and enable or disable workload management. Changes should only be made under the direction of Microsoft Customer Service and Support.

This management role is one of several built-in roles in the Role Based Access Control (RBAC) permissions model in Microsoft Exchange Server 2013. Management roles, which are assigned to one or more management role groups, management role assignment policies, users, or universal security groups (USG), act as a logical grouping of cmdlets or scripts that are combined to provide access to view or modify the configuration of Exchange 2013 components, such as mailbox databases, transport rules, and recipients. If a cmdlet or script and its parameters, together called a management role entry, are included on a role, that cmdlet or script and its parameters can be run by those assigned the role. For more information about management roles and management role entries, see Understanding management roles.

For more information about management roles, management role groups, and other RBAC components, see Understanding Role Based Access Control.

Management role assignments

For this role to grant permissions, it must be assigned to a role assignee, which can be a role group, user, or universal security group (USG). This assignment is done using management role assignments. Role assignments link role assignees and roles together. If more than one role is assigned to a role assignee, the role assignee is granted the combination of all the permissions granted by all the assigned roles.

In addition to linking role assignees to roles, role assignments can also apply custom or built-in management scopes. Management scopes control which recipient, server and database objects can be modified by role assignees. If this role is assigned to a role assignee, but a management scope allows the role assignee only to manage certain objects based on a defined scope, the role assignee can only use the permissions granted by this role on those specific objects. The permissions provided by this role can't be applied to objects outside the scope defined on the role assignment. For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

This role is assigned to one or more role groups by default. For more information, see the "Default Management Role Assignments" section later in this topic.

If you want to view a list of role groups, users, or USGs assigned to this role, use the following command.

```
Get-ManagementRoleAssignment -Role "<role name>"
```

Regular and delegating role assignments

This role can be assigned to role assignees using either regular or delegating role assignments. Regular role assignments grant the permissions provided by the role to the role assignee. Delegating role assignments grant the role assignee the ability to assign the role to other role assignees. For more information about regular and delegating role assignments, see Understanding management role assignments.

Adding or removing role assignments

You can change which role assignees are assigned this role. By changing which role assignee is assigned this role, you change who is granted its permissions. You can assign this role to other built-in role groups, or you can create role groups and assign this role to them. You can also assign this role to users or USGs. However, we recommend that you limit assignment of roles to users and USGs because such assignments can greatly increase the complexity of your permissions model.

To assign this role to role assignees, the role must be assigned to a role group you're a member of, directly to you, or to a USG you're a member of, using a delegating role assignment. For more information about delegating role assignments, see the "Regular and Delegating Role Assignments" section.

You can also remove this role from built-in role groups, role groups you create, users, and USGs. However, there must always be at least one delegating role assignment between this role and a role group or USG. You can't delete the last delegating role assignment. This limitation helps prevent you from locking yourself out of the system.

Important:
There must be at least one delegating role assignment between this role and a role group or USG. You can't remove the last delegating role assignment associated with this role if the last assignment is to a user.

For more information about how to add or remove assignments between this role and role groups, users, and USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG
- Remove a role from a user or USG

Changing the management scopes on role assignments

You can also change the management scopes on existing role assignments between this role and role assignees. By changing the scopes on role assignments, you control what objects can be managed using the permissions provided by this role. You have several choices when changing the scope on a role assignment. You can do one of the following:

- Add a new custom scope using the **Set-ManagementRoleAssignment** cmdlet. For more

information, see the following topics:
  - Create a regular or exclusive scope
  - Change a role assignment
- Add or change an organizational unit scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Add or change a predefined scope using the **Set-ManagementRoleAssignment** cmdlet. For more information, see Change a role assignment.
- Change the recipient, server, or database scope on a custom scope associated with a role assignment using the **Set-ManagementScope** cmdlet. For more information, see Change a role scope.

Enabling or disabling role assignments

By enabling or disabling a role assignment, you control whether that role assignment should be in effect. If a role assignment is disabled, the permissions granted by the associated role aren't applied to the role assignee. This is convenient if you want to temporarily remove permissions without deleting a role assignment. For more information, see Change a role assignment.

Default management role assignments

This role has role assignments to one or more role assignees. The following table indicates whether the role assignment is regular or delegating, and also indicates the management scopes applied to each assignment. The following list describes each column:

- **Regular assignment** Regular role assignments enable the role assignee to access the permissions provided by the management role entries on this role.
- **Delegating assignment** Delegating role assignments give the role assignee the ability to assign this role to role groups, users, or USGs.
- **Recipient read scope** The recipient read scope determines what recipient objects the role assignee is allowed to read from Active Directory.
- **Recipient write scope** The recipient write scope determines what recipient objects the role assignee is allowed to modify in Active Directory.
- **Configuration read scope** The configuration read scope determines what configuration and server objects the role assignee is allowed to read from Active Directory.
- **Configuration write scope** The configuration write scope determines what organizational and server objects the role assignee is allowed to modify in Active Directory.

Default management role assignments for this role

Role group	Regular assignment	Delegating assignment	Recipient read scope	Recipient write scope	Configuration read scope	Configuration write scope
Organization	X	X	`Organization`	`Organization`	`OrganizationConfig`	`OrganizationConfig`

Management						

Management role customization

This role has been configured to provide a role assignee with all necessary cmdlets and parameters to manage the features and components listed in the beginning of this topic. Other roles have also been provided to enable management of other features. By adding and removing roles to and from role groups, you can create a customized permissions model without the need to customize individual management roles. For a complete list of roles, see Built-in management roles. For more information about customizing role groups, see Manage role groups.

If you decide that you need to create a customized version of this role, you must create a role as a child of this role, and customize the new role.

Caution:

The following information enables you to perform advanced management of permissions. Customizing management roles can significantly increase the complexity of your permissions model. You could cause certain features to stop functioning if you replace a built-in management role with an incorrectly configured custom role.

The following are the most common steps to create a customized role and assign it to a role assignee:

1. Create a copy of this role. For more information, see Create a role.
2. Change or remove the role entries on the new role using the **Set-ManagementRoleEntry** and **Remove-ManagementRoleEntry** cmdlets. You can't add additional role entries to the new role because it can only contain the role entries on the parent built-in role. For more information, see the following topics:
   - Change a role entry
   - Remove a role entry from a role
3. If you want to replace the built-in role with this new customized role, remove any role assignments associated with the built-in role. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Remove a role from a user or USG
4. Add the new customized role to the required role assignees. For more information, see the following topics:
   - “Add or remove a role to or from a role group” section in Manage role groups
   - Add a role to a user or USG

Important:

If you want other users, in addition to the user that created the role, to be able to assign the new customized role, be sure to add a delegating role assignment to at least one role assignee. For more information, see Delegate role assignments.

Understanding multiple-forest permissions

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-05*

Many organizations deploy multiple forests to create security boundaries within their organizations. Using multiple forests helps administrators to define these security boundaries to better match their requirements, whether that's ensuring the fewest number of people have access to resources, or segmenting divisions within an organization.

Microsoft Exchange Server 2013 supports two types of multiple forest topologies:

- **Cross-forest** Cross-forest topologies can have multiple forests, each with their own installation of Exchange.
- **Resource forest** Resource forest topologies have an Exchange forest and one or more accounts forests.

For the purposes of this topic, the forest that contains the universal security groups (USGs) and users outside of the forest where Exchange 2013 is installed, whether it's an accounts forest or other resource forest, is called a foreign forest.

Configuration of permissions in a multiple forest topology relies on the correct configuration of forest trusts and global address list (GAL) synchronization for the creation of linked mailboxes. The Exchange 2013 forest must trust the foreign forest that contains the USGs associated with linked role groups and users associated with linked mailboxes.

Exchange 2013 uses a Role Based Access Control (RBAC) permissions model. The management role groups that administrators are members of, and the management role assignment policies that end users are assigned, determine what each administrator and end user can do. To understand multiple-forest permissions, you need to be familiar with RBAC. For more information about RBAC, role groups, and role assignment policies, see the following topics:

- Understanding Role Based Access Control
- Understanding management role groups
- Understanding management role assignment policies

Looking for management tasks related to managing permissions? See Permissions.

Contents

Permissions in a multiple forest topology

RBAC applies permissions to all Exchange objects within a single forest and the RBAC configuration in each forest is configured independently of all other forests. When you create a role group in one forest, that role group doesn't exist in any other forest and the permissions granted by that role group apply only to the forest in which it was created. For example, a member of a role group that grants permissions to create a mailbox can create a mailbox only in the forest that contains that role group.

If you have multiple Exchange forests and want to configure permissions identically within each forest, you must apply the same configuration explicitly in each forest. For example, if you have two Exchange 2013 forests and want to create a Compliance Management role group to manage permissions for your legal department, you must do the following:

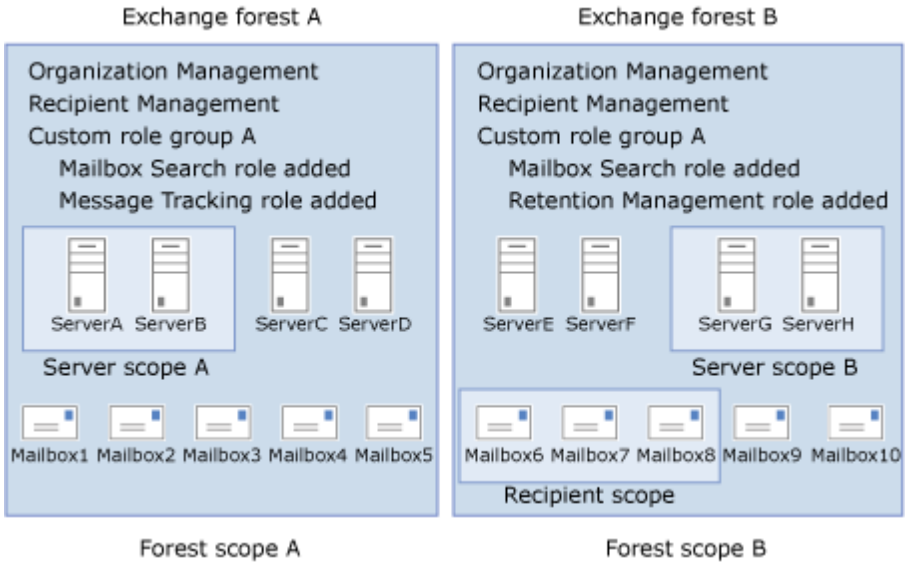
- In each forest, create a role group named Compliance Management. If your administrators are in a separate foreign forest from either Exchange forest, create both role groups as linked role groups. For more information about role groups, see the Cross-Boundary Permissions section.
- In each forest, create role assignments between the new role groups and the roles that you want to use.
- As part of the new role assignments, optionally add management scopes that encompass the server and recipient objects within each forest.
- If you created the role groups as linked role groups, add members to the associated USG in the foreign forest.

The following figure shows how the role groups configured within Exchange 2013 forests are bound to their respective forests. The Organization Management role group in Exchange 2013 forest A grants permissions only to manage the mailboxes and servers that are within that forest. Likewise, the role groups in Exchange 2013 forest B grant permissions only to the mailboxes and servers within that forest.

The figure also shows that the Custom role group A is created in each forest. Even though they were created with the same name, each is its own separate entity. In fact, as the figure shows, each can be assigned different management roles in their respective forests. Custom role group A in Exchange 2013 forest A is assigned the Mailbox Search and Message Tracking roles while Custom role group A in Exchange 2013 forest B is assigned the Mailbox Search and Retention Management roles.

Finally, management scopes created in each forest are also bound by the forest. Server scopes created in each forest can only contain the servers that are members of that forest. Server scope A can contain only servers within Exchange 2013 forest A while Server scope B can contain only servers that are within Exchange 2013 forest B. Similarly, the recipient scope in Exchange 2013 forest B can only contain mailboxes that are within Exchange 2013 forest B.

RBAC and forest scope boundary relationship



Cross-boundary permissions

The permissions granted by RBAC only allow users to view or modify Exchange objects within a specific forest. However, you can grant permissions to view and modify Exchange objects in a forest to users outside of that forest. By using cross-boundary permissions, you can centralize Exchange management accounts in a single forest rather than having to authenticate against each individual forest to perform tasks.

Note:

The permissions that are granted to a user outside of an Exchange forest still apply only to that specific Exchange forest. For example, if a user in a foreign forest is a member of the Organization Management linked role group that's located in ForestA, the user can manage only the Exchange objects contained within ForestA. A user must be made a member of linked role groups in each Exchange forest to be granted permissions to manage each forest.

Cross-boundary permissions also enable you to apply role assignment policies to the mailboxes of users who have mailboxes in an Exchange forest, but have user accounts that reside in an accounts forest. Exchange 2013 supports cross-boundary permission using linked role groups and linked mailboxes, which are discussed in the following sections.

Administrative permissions

Administrative permissions are granted cross forest boundaries by the use of linked role groups and linked mailboxes.

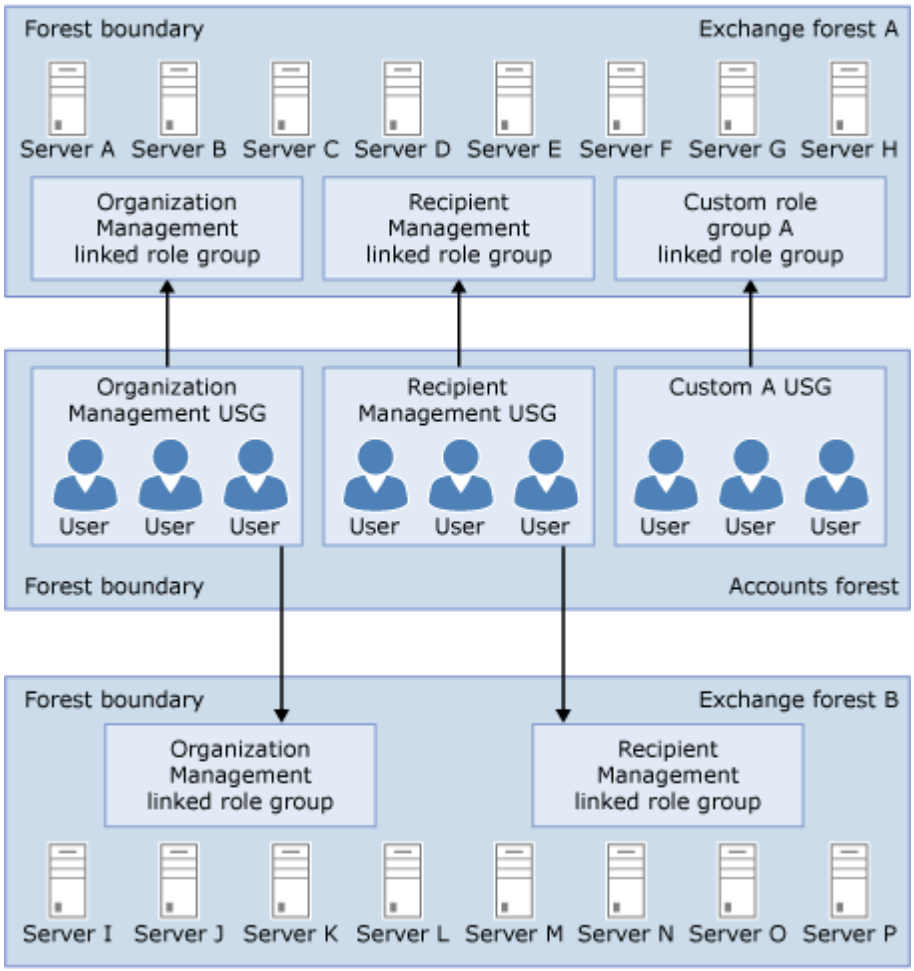
A linked role group is created in the Exchange 2013 organization and is linked to a USG across the forest boundary in the foreign forest. The USG the linked role group is linked to can be any of the following:

- A dedicated USG for the specific use of the linked role group
- A USG that's linked to by linked role groups in multiple Exchange 2013 forests
- A role group USG in another Exchange 2013 forest
- A USG associated with an Exchange Server 2007 administrative role or Exchange 2010 role group

The USG that a linked role group is linked to must be in another forest. You can't link a linked role group to a USG in the same forest.

The following figure shows that USGs in an accounts forest can be associated with role groups in one or more Exchange 2013 resource forests. The members of the USGs in the accounts forest effectively become members of the role groups through the USGs.

Linked role groups associated with USGs in a separate forest



When you create a linked role group, you assign roles to the linked role group in the Exchange 2013 forest. The assignments that associate the roles to the linked role group can include management scopes, if necessary. These scopes are confined to the forest in which the linked role group is created.

Membership of the linked role group is managed by adding and removing members to and from the USG in the foreign forest. When you add members to this USG, they are granted the permissions assigned to the linked role group in the Exchange 2013 forest. If you've linked multiple linked role groups with the same USG, the members of that USG are granted the permissions assigned to each linked role group in each Exchange 2013 forest.

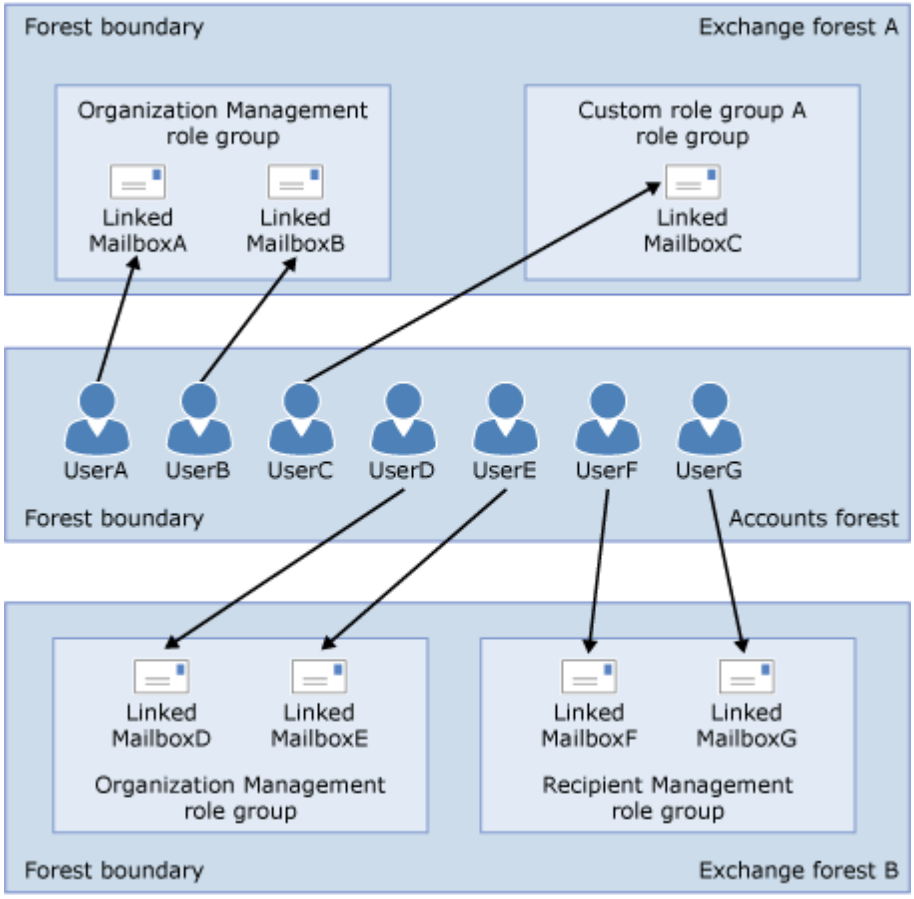
You can't manage the membership of a linked role group from the Exchange 2013 forest.

A second method to assign administrative permissions across forest boundaries is through the use of linked mailboxes. For users in an accounts forest to use an Exchange 2013 deployment in a separate Exchange 2013 resource forest, you must configure linked mailboxes for each user. Linked

mailboxes can be added as members to role groups within the Exchange 2013 forest. When a linked mailbox becomes a member of a role group, that linked mailbox, and in turn the user in the accounts forest associated with the linked mailbox, is granted the permissions provided by the role group.

The following figure shows the relationship between users in an accounts forest, the linked mailboxes associated with them, and the role groups in which they're members.

Users in an accounts forest associated with linked mailboxes that are members of role groups



Linked role groups and linked mailboxes both have advantages and disadvantages when used to assign administrative permission across forest boundaries. The following table describes some of them.

Linked role group and linked mailbox advantages and disadvantages

Linked role groups or linked mailboxes	Advantage	Disadvantage
Linked role groups	You can associate multiple linked role groups from multiple Exchange 2013 forests to a single USG in an accounts forest or other Exchange resource forest. This enables	A regular role group can't be converted to a linked role group. You must manually create linked role groups to replace each regular role group that has the permissions you

	you to administer complex Exchange forest topologies through a small set of USGs in a single forest.	want to grant across a forest boundary. For more information, see Configure cross-boundary permissions.
Linked mailboxes	Linked mailboxes allow you to use the existing role groups within the Exchange forest. Linked mailboxes are added as members to the existing role groups just like regular mailboxes, USGs, and users in the same Exchange forest.	If you grant permissions in multiple Exchange 2013 forests using linked mailboxes linked to a single user in an accounts forest, you must modify the role group membership in each Exchange 2013 forest if you want to modify the permissions granted to the user.

We recommend that you use linked role groups to grant permission across forest boundaries if you plan on having multiple Exchange resource forests.

End-user permissions

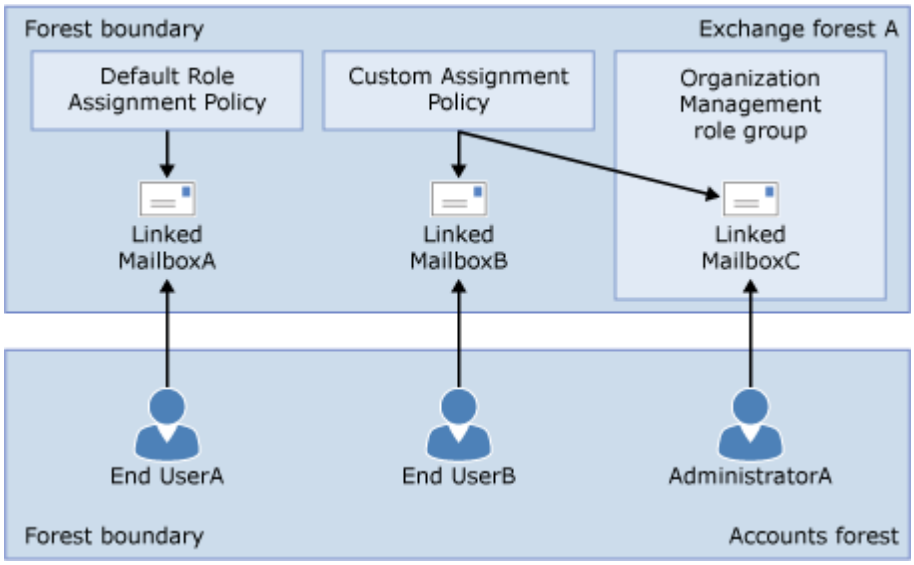
End-user permissions are assigned to individual mailboxes using role assignment policies. When Exchange 2013 is installed in a resource forest, linked mailboxes are created in the resource forest and are associated with user accounts in the accounts forest.

When a linked mailbox is created, it's assigned to a default role assignment policy just like a regular mailbox. The role assignment policy determines which end-user permissions are granted to the mailbox. These permissions enable users to view and modify settings related to the following, and other, features:

- End-user profile information
- End-user voicemail
- End-user distribution membership and ownership

When a role assignment policy is assigned to a linked mailbox, the user in the accounts forest associated with the linked mailbox is granted permissions to manage the features available to that user. The permissions apply to only the resources in the Exchange forest where the linked mailbox is located. The following figure shows the relationship between the end user in the accounts forest, its associated linked mailbox, and the role assignment policy assigned to the linked mailbox. Additionally, a linked mailbox associated with an administrative user in the accounts forest can be associated with multiple role groups in addition to a role assignment policy.

Users in an accounts forest associated with linked mailboxes that are each assigned a role assignment policy



Return to top

Configure cross-boundary permissions

To configure cross-boundary permissions in a multiple-forest topology, you must create linked role groups for each of the role groups you want to link to USGs in a foreign forest. This means that you must create a linked role group for each built-in role group. You need to:

1. Create a USG in the foreign forest for each linked role group to be created. Add members to this USG that you want to grant permissions to.
2. Create a linked role group for each built-in role group. The following happens when the linked role group is created:
   - The same roles that are assigned to the built-in role group are assigned to the new linked role group.
   - The linked role group is associated with the USG in the foreign forest.
3. Create linked role groups for any custom role groups you created.
4. Optionally assign custom scopes to the new linked role groups.

For detailed information about how to perform these steps, see the following topics:

- Create linked role groups that mirror built-in role groups
- Manage linked role groups
- Manage role groups

If you need to change the USG that a linked role group is associated with, see Manage linked role groups.

When a linked mailbox is created, it's automatically assigned to a role assignment policy. You can change the role assignment policy that's assigned to the linked mailbox or change the role assignment policy that's assigned to mailboxes by default when they're created. For more information, see the following topics:

- Change the assignment policy on a mailbox

- Manage role assignment policies

Understanding split permissions

Exchange Server 2013 > Permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-05*

Organizations that separate the management of Microsoft Exchange Server 2013 objects and Active Directory objects use what's called a *split permissions* model. Split permissions enable organizations to assign specific permissions and related tasks to specific groups within the organization. This separation of work helps to maintain standards and workflows, and helps to control change in the organization.

The highest level of split permissions is the separation of Exchange management and Active Directory management. Many organizations have two groups: administrators that manage the organization's Exchange infrastructure, including servers and recipients, and administrators that manage the Active Directory infrastructure. This is an important separation for many organizations because the Active Directory infrastructure often spans many locations, domains, services, applications, and even Active Directory forests. Active Directory administrators must ensure that changes made to Active Directory don't negatively impact any other services. As a result, typically only a small group of administrators is allowed to manage that infrastructure.

At the same time, the infrastructure for Exchange, including servers and recipients, can also be complex and require specialized knowledge. Additionally, Exchange stores extremely confidential information about the business of the organization. Exchange administrators can potentially access this information. By limiting the number of Exchange administrators, the organization limits who can make changes to Exchange configuration and who can access sensitive information.

Split permissions typically make a distinction between the creation of security principals in Active Directory, such as users and security groups, and the subsequent configuration of those objects. This helps to reduce the chance of unauthorized access to the network by controlling who can create objects that grant access to it. Most often only Active Directory administrators can create security principals while other administrators, such as Exchange administrators, can manage specific attributes on existing Active Directory objects.

To support the varying needs to separate the management of Exchange and Active Directory, Exchange 2013 lets you choose whether you want a shared permissions model or a split permissions model. Exchange 2013 offers two types of split permissions models: RBAC and Active Directory. Exchange 2013 defaults to a shared permissions model.

Contents

Shared permissions

Split permissions

RBAC split permissions

Active Directory split permissions

Explanation of Role Based Access Control and Active Directory

To understand split permissions, you need to understand how the Role Based Access Control (RBAC) permissions model in Exchange 2013 works with Active Directory. The RBAC model controls who can perform what actions, and on which objects those actions can be performed. For more information about the various components of RBAC that are discussed in this topic, see Understanding Role Based Access Control.

In Exchange 2013, all tasks that are performed on Exchange objects must be done through the Exchange Management Shell or the Exchange Administration Center (EAC) interface. Both of these management tools use RBAC to authorize all tasks that are performed.

RBAC is a component that exists on every server running Exchange 2013. RBAC checks whether the user performing an action is authorized to do so:

- If the user isn't authorized to perform the action, RBAC doesn't allow the action to proceed.
- If the user is authorized to perform the action, RBAC checks whether the user is authorized to perform the action against the specific object being requested:
  - If the user is authorized, RBAC allows the action to proceed.
  - If the user isn't authorized, RBAC doesn't allow the action to proceed.

If RBAC allows an action to proceed, the action is performed in the context of the Exchange Trusted Subsystem and not the user's context. The Exchange Trusted Subsystem is a highly privileged universal security group (USG) that has read/write access to every Exchange-related object in the Exchange organization. It's also a member of the Administrators local security group and the Exchange Windows Permissions USG, which enables Exchange to create and manage Active Directory objects.

Caution:
Don't make any manual changes to the membership of the Exchange Trusted Subsystem security group. Also, don't add it to or remove it from object access control lists (ACLs). By making changes to the Exchange Trusted Subsystem USG yourself, you could cause irreparable damage to your Exchange organization.

It's important to understand that it doesn't matter what Active Directory permissions a user has

when using the Exchange management tools. If the user is authorized, via RBAC, to perform an action in the Exchange management tools, the user can perform the action regardless of his or her Active Directory permissions. Conversely, if a user is an Enterprise Admin in Active Directory but isn't authorized to perform an action, such as creating a mailbox, in the Exchange management tools, the action won't succeed because the user doesn't have the required permissions according to RBAC.

Important:
Although the RBAC permissions model doesn't apply to the Active Directory Users and Computers management tool, Active Directory Users and Computers can't manage the Exchange configuration. So although a user may have access to modify some attributes on Active Directory objects, such as the display name of a user, the user must use the Exchange management tools, and therefore must be authorized by RBAC, to manage Exchange attributes.

Return to top

Shared permissions

The shared permissions model is the default model for Exchange 2013. You don't need to change anything if this is the permissions model you want to use. This model doesn't separate the management of Exchange and Active Directory objects from within the Exchange management tools. It allows administrators using the Exchange management tools to create security principals in Active Directory.

The following table shows the roles that enable the creation of security principals in Exchange and the management role groups they're assigned to by default.

Security principal management roles

Management role	Role group
Mail Recipient Creation role	Organization Management Recipient Management
Security Group Creation and Membership role	Organization Management

Only role groups, users, or USGs that are assigned the Mail Recipient Creation role can create security principals such as Active Directory users. By default, the Organization Management and Recipient Management role groups are assigned this role. Therefore members of these role groups can create security principals.

Only role groups, users, or USGs that are assigned the Security Group Creation and Membership role can create security groups or manage their memberships. By default, only the Organization Management role group is assigned this role. Therefore only members of the Organization Management role group can create or manage the membership of security groups.

You can assign the Mail Recipient Creation role and the Security Group Creation and Membership

role to other role groups, users, or USGs if you want other users to be able to create security principals.

To enable the management of existing security principals in Exchange 2013, the Mail Recipients role is assigned to the Organization Management and Recipient Management role groups by default. Only role groups, users, or USGs that are assigned the Mail Recipients role can manage existing security principals. If you want other role groups, users, or USGs to be able to manage existing security principals, you must assign the Mail Recipients role to them.

For more information about how to add roles to role groups, users, or USGs, see the following topics:

- Manage role groups
- Add a role to a user or USG

If you switched to a split permissions model and want to change back to a shared permissions model, see Configure Exchange 2013 for shared permissions.

Return to top

Split permissions

If your organization separates Exchange management and Active Directory management, you need to configure Exchange to support the split permissions model. When configured correctly, only the administrators who you want to create security principals, such as Active Directory administrators, will be able to do so and only Exchange administrators will be able to modify the Exchange attributes on existing security principals. This splitting of permissions also falls roughly along the lines of the domain and configuration partitions in Active Directory. Partitions are also called naming contexts. The domain partition stores the users, groups, and other objects for a specific domain. The configuration partition stores the forest-wide configuration information for the services that used Active Directory, such as Exchange. Data that's stored in the domain partition is typically managed by Active Directory administrators, although objects may contain Exchange-specific attributes that can be managed by Exchange administrators. Data that's stored in the configuration partition is managed by the administrators for each respective service that stores data in this partition. For Exchange, this is typically Exchange administrators.

Exchange 2013 supports the two following types of split permissions:

- **RBAC split permissions** Permissions to create security principals in the Active Directory domain partition are controlled by RBAC. Only Exchange servers, services, and those who are members of the appropriate role groups can create security principals.
- **Active Directory split permissions** Permissions to create security principals in the Active Directory domain partition are completely removed from any Exchange user, service, or server. No option is provided in RBAC to create security principals. Creation of security principals in Active Directory must be performed using Active Directory management tools.

Important:

Although Active Directory split permissions can be enabled or disabled by running Setup on a computer that has Exchange 2013 installed, Active Directory split permissions configuration applies to both Exchange 2013 and Exchange 2010 servers. It doesn't, however, have any impact on Microsoft Exchange Server 2007 servers.

If your organization chooses to use a split permissions model instead of shared permissions, we recommend that you use the RBAC split permissions model. The RBAC split permissions model provides significantly more flexibility while providing the nearly same administration separation as Active Directory split permissions, with the exception that Exchange servers and services can create security principals in the RBAC split permissions model.

You're asked whether you want to enable Active Directory split permissions during Setup. If you choose to enable Active Directory split permissions, you can only change to shared permissions or RBAC split permissions by rerunning Setup and disabling Active Directory split permissions. This choice applies to all Exchange 2010 and Exchange 2013 servers in the organization.

The following sections describe RBAC and Active Directory split permissions in more detail.

Return to top

RBAC split permissions

The RBAC security model modifies the default management role assignments to separate who can create security principals in the Active Directory domain partition from those who administer the Exchange organization data in the Active Directory configuration partition. Security principals, such as users with mailboxes and distribution groups, can be created by administrators who are members of the Mail Recipient Creation and Security Group Creation and Membership roles. These permissions remain separate from the permissions required to create security principals outside of the Exchange management tools. Exchange administrators who aren't assigned the Mail Recipient Creation or Security Group Creation and Membership roles can still modify Exchange-related attributes on security principals. Active Directory administrators also have the option of using the Exchange management tools to create Active Directory security principals.

Exchange servers and the Exchange Trusted Subsystem also have permissions to create security principals in Active Directory on behalf of users and third-party programs that integrate with RBAC.

RBAC split permissions is a good choice for your organization if the following are true:

- Your organization doesn't require that security principal creation be performed using only Active Directory management tools and only by users who are assigned specific Active Directory permissions.
- Your organization allows services, such as Exchange servers, to create security principals.
- You want to simplify the process required to create mailboxes, mail-enabled users, distribution groups, and role groups by allowing their creation from within the Exchange management tools.
- You want to manage the membership of distribution groups and role groups within the Exchange management tools.
- You have third-party programs that require that Exchange servers be able to create security

principals on their behalf.

If your organization requires a complete separation of Exchange and Active Directory administration where no Active Directory administration can be performed using Exchange management tools or by Exchange services, see the Active Directory Split Permissions section later in this topic.

Switching from shared permissions to RBAC split permissions is a manual process where you remove the permissions required to create security principals from the role groups that are granted them by default. The following table shows the roles that enable the creation of security principals in Exchange and the management role groups they're assigned to by default.

Security principal management roles

Management role	Role group
Mail Recipient Creation role	Organization Management Recipient Management
Security Group Creation and Membership role	Organization Management

By default, members of the Organization Management and Recipient Management role groups can create security principals. You must transfer the ability to create security principals from the built-in role groups to a new role group that you create.

To configure RBAC split permissions, you must do the following:

1. Disable Active Directory split permissions if it's enabled.
2. Create a role group, which will contain the Active Directory administrators that will be able to create security principals.
3. Create regular and delegating role assignments between the Mail Recipient Creation role and the new role group.
4. Create regular and delegating role assignments between the Security Group Creation and Membership role and the new role group.
5. Remove the regular and delegating management role assignments between the Mail Recipient Creation role and the Organization Management and Recipient Management role groups.
6. Remove the regular and delegating role assignments between the Security Group Creation and Membership role and the Organization Management role group.

After doing this, only members of the new role group that you create will be able to create security principals, such as mailboxes. The new group will only be able to create the objects. It won't be able to configure the Exchange attributes on the new object. An Active Directory administrator, who is a member of the new group, will need to create the object, and then an Exchange administrator will need to configure the Exchange attributes on the object. Exchange administrators won't be able to use the following cmdlets:

- **New-Mailbox**
- **New-MailContact**

- **New-MailUser**
- **New-RemoteMailbox**
- **Remove-Mailbox**
- **Remove-MailContact**
- **Remove-MailUser**
- **Remove-RemoteMailbox**

Exchange administrators will, however, be able to create and manage Exchange-specific objects, such as transport rules, distribution groups, and so on and manage Exchange-related attributes on any object.

Additionally, the associated features in the EAC and Outlook Web App, such as the New Mailbox Wizard, will also no longer be available or will generate an error if you try to use them.

If you want the new role group to also be able to manage the Exchange attributes on the new object, the Mail Recipients role also needs to be assigned to the new role group.

For more information about configuring a split permissions model, see Configure Exchange 2013 for split permissions.

Return to top

Active Directory split permissions

With Active Directory split permissions, the creation of security principals in the Active Directory domain partition, such as mailboxes and distribution groups, must be performed using Active Directory management tools. Several changes are made to the permissions granted to the Exchange Trusted Subsystem and Exchange servers to limit what Exchange administrators and servers can do. The following changes in functionality occur when you enable Active Directory split permissions:

- Creation of mailboxes, mail-enabled users, distribution groups, and other security principals is removed from the Exchange management tools.
- Adding and removing distribution group members can't be done from the Exchange management tools.
- All permissions granted to the Exchange Trusted Subsystem and Exchange servers to create security principals are removed.
- Exchange servers and the Exchange management tools can only modify the Exchange attributes of existing security principals in Active Directory.

For example, to create a mailbox with Active Directory split permissions enabled, a user must first be created using Active Directory tools by a user with the required Active Directory permissions. Then, the user can be mailbox-enabled using the Exchange management tools. Only the Exchange-related attributes of the mailbox can be modified by Exchange administrators using the Exchange management tools.

Active Directory split permissions is a good choice for your organization if the following are true:

- Your organization requires that security principals be created using only the Active Directory

management tools or only by users who are granted specific permissions in Active Directory.

- You want to completely separate the ability to create security principals from those who manage the Exchange organization.
- You want to perform all distribution group management, including creation of distribution groups and adding and removing members of those groups, using Active Directory management tools.
- You don't want Exchange servers, or third-party programs that use Exchange on their behalf, to create security principals.

Important:

Switching to Active Directory split permissions is a choice that you can make when you install Exchange 2013 either by using the Setup wizard or by using the *ActiveDirectorySplitPermissions* parameter while running `setup.exe` from the command line. You can also enable or disable Active Directory split permissions after you've installed Exchange 2013 by rerunning `setup.exe` from the command line. To enable Active Directory split permissions, set the *ActiveDirectorySplitPermissions* parameter to `true`. To disable it, set it to `false`. You must always specify the *PrepareAD* switch along with the *ActiveDirectorySplitPermissions* parameter.
If you have multiple domains within the same forest, you must also either specify the *PrepareAllDomains* switch when you apply Active Directory split permissions or run setup with the *PrepareDomain* switch in each domain. If you choose to run setup with the *PrepareDomain* switch in each domain rather than use the *PrepareAllDomains* switch, you must prepare every domain that contains Exchange servers, mail-enabled objects, or global catalog servers that could be accessed by an Exchange server.

Important:

You can't enable Active Directory split permissions if you've installed Exchange 2010 or Exchange 2013 on a domain controller.
After you enable or disable Active Directory split permissions, we recommend that you restart the Exchange 2010 and Exchange 2013 servers in your organization to force them to pick up the new Active Directory access token with the updated permissions.

Exchange 2013 achieves Active Directory split permissions by removing permissions and membership from the Exchange Windows Permissions security group. This security group, in shared permissions and RBAC split permissions, is given permissions to many non-Exchange objects and attributes throughout Active Directory. By removing the permissions and membership to this security group, Exchange administrators and services are prevented from creating or modifying those non-Exchange Active Directory objects.

For a list of changes that occur to the Exchange Windows Permissions security group and other Exchange components when you enable or disable Active Directory split permissions, see the following table.

Note:

Role assignments to role groups that enable Exchange administrators to create security principals are removed when Active Directory split permissions is enabled. This is done to remove access to cmdlets that would otherwise generate an error when they're run because they don't have permissions to create the associated Active Directory object.

Active Directory split permissions changes

Action	Changes made by Exchange
Enable Active Directory split permissions during first Exchange 2013 server installation	The following happens when you enable Active Directory split permissions either through the Setup wizard or by running `setup.exe` with the `/PrepareAD` and `/ActiveDirectorySplitPermissions:true` parameters: • An organizational unit (OU) called **Microsoft Exchange Protected Groups** is created. • The **Exchange Windows Permissions** security group is created in the **Microsoft Exchange Protected Groups** OU. • The **Exchange Trusted Subsystem** security group isn't added to the **Exchange Windows Permissions** security group. • Creation of non-delegating management role assignments to management roles with the following management role types is skipped: ○ `MailRecipientCreation` ○ `SecurityGroupCreationandMembership` • Access control entries (ACEs) that would have been assigned to the **Exchange Windows Permissions** security group aren't added to the Active Directory domain object. If you run setup with the *PrepareAllDomains* or *PrepareDomain* switch, the following happens in each child domain that's prepared: • All ACEs assigned to the **Exchange Windows Permissions** security group are removed from the domain object. • ACEs are set in each domain with the exception of any ACEs assigned to the **Exchange Windows Permissions** security group.

Switch from shared permissions or RBAC split permissions to Active Directory split permissions	The following happens when you run the `setup.exe` command with the `/PrepareAD` and `/ActiveDirectorySplitPermissions:true` parameters: • An OU called **Microsoft Exchange Protected Groups** is created. • The **Exchange Windows Permissions** security group is moved to the **Microsoft Exchange Protected** Groups OU. • The **Exchange Trusted Subsystem** security group is removed from the **Exchange Windows Permissions** security group. • Any non-delegating role assignments to management roles with the following role types are removed: ○ `MailRecipientCreation` ○ `SecurityGroupCreationandMembership` • All ACEs assigned to the **Exchange Windows Permissions** security group are removed from the domain object. If you run setup with either the *PrepareAllDomains* or *PrepareDomain* switch, the following happens in each child domain that's prepared: • All ACEs assigned to the **Exchange Windows Permissions** security group are removed from the domain object. • ACEs are set in each domain with the exception of any ACEs assigned to the **Exchange Windows Permissions** security group.
Switch from Active Directory split permissions to shared permissions or RBAC split permissions	The following happens when you run the `setup.exe` command with the `/PrepareAD` and `/ActiveDirectorySplitPermissions:false` parameters:

	• The **Exchange Windows Permissions** security group is moved to the **Microsoft Exchange Security Groups** OU. • The **Microsoft Exchange Protected Groups** OU is removed. • The **Exchange Trusted Subsystems** security group is added to the **Exchange Windows Permissions** security group. • ACEs are added to the domain object for the **Exchange Windows Permissions** security group. If you run setup with either the *PrepareAllDomains* or *PrepareDomain* switch, the following happens in each child domain that's prepared: • ACEs are added to the domain object for the **Exchange Windows Permissions** security group. • ACEs are set in each domain including ACEs assigned to the **Exchange Windows Permissions** security group. Role assignments to the Mail Recipient Creation and Security Group Creation and Membership roles aren't automatically created when switching from Active Directory split to shared permissions. If delegating role assignments were customized prior to Active Directory split permissions being enabled, those customizations are left intact. To create role assignments between these roles and the Organization Management role group, see Configure Exchange 2013 for shared permissions.

After you enable Active Directory split permissions, the following cmdlets are no longer available:

- **New-Mailbox**

- **New-MailContact**
- **New-MailUser**
- **New-RemoteMailbox**
- **Remove-Mailbox**
- **Remove-MailContact**
- **Remove-MailUser**
- **Remove-RemoteMailbox**

After you enable Active Directory split permissions, the following cmdlets are accessible but you can't use them to create distribution groups or modify distribution group membership:

- **Add-DistributionGroupMember**
- **New-DistributionGroup**
- **Remove-DistributionGroup**
- **Remove-DistributionGroupMember**
- **Update-DistributionGroupMember**

Some cmdlets, although still available, may offer only limited functionality when used with Active Directory split permissions. This is because they may allow you to configure recipient objects that are in the domain Active Directory partition and Exchange configuration objects that are in the configuration Active Directory partition. They may also allow you to configure Exchange-related attributes on objects stored in the domain partition. Attempts to use the cmdlets to create objects, or modify non-Exchange-related attributes on objects, in the domain partition will result in an error. For example, the **Add-ADPermission** cmdlet will return an error if you attempt to add permissions to a mailbox. However, the **Add-ADPermission** cmdlet will succeed if you configure permissions on a Receive connector. This is because a mailbox is stored in the domain partition while Receive connectors are stored in the configuration partition.

Additionally, the associated features in the Exchange Administration Center and Outlook Web App, such as the New Mailbox wizard, will also no longer be available or will generate an error if you try to use them.

Exchange administrators will, however, be able to create and manage Exchange-specific objects, such as transport rules, and so on.

For more information about configuring an Active Directory split permissions model, see Configure Exchange 2013 for split permissions.

Return to top

Understanding permissions coexistence with Exchange 2007 and Exchange 2010

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-05-21*

When you install Microsoft Exchange Server 2013 into an existing Microsoft Exchange Server 2010 or Microsoft Exchange Server 2007 organization, you need to understand how permissions will work in the new organization. Read the section below that applies to your organization.

- Installing Exchange 2013 into an existing Exchange 2010 organization
- Installing Exchange 2013 into an existing Exchange 2007 organization

Installing Exchange 2013 into an existing Exchange 2010 organization

Exchange 2013 uses the same Role Based Access Control (RBAC) permissions model that's used in Exchange 2010. When you install Exchange 2013 into an existing Exchange 2010 organization, the same management role groups, management roles, and management scopes apply to both Exchange 2013 and Exchange 2010 servers and recipients. Members of role groups, or users assigned to roles, can administer any Exchange 2013 or Exchange 2013 server or recipient that's included in the scope of the role group or role. If you don't use scopes in your organization and you want the members of your existing role groups to manage Exchange 2010 and Exchange 2013 servers and recipients, you don't have to do anything else. Those administrators will be able to manage Exchange 2013 servers and recipients that are added to the organization. If you need a reminder on how permissions work in Exchange 2010 and Exchange 2013, see Permissions.

In the new organization, you might want to separate the administration of Exchange 2010 and Exchange 2013 servers and recipients. The group of administrators responsible for administering Exchange 2010 servers and recipients may not be allowed to administer Exchange 2013 servers and recipients, and vice versa. In this case, you can use management scopes to define the servers and recipients each group of administrators should be allowed to manage. After you create the scopes you want, you can then copy existing role groups, add the administrators who should be a member of each, and then add the scopes to those role groups. When you're done, the members of each role group will only be able to administer the servers and recipients that match their respective scopes.

For more information about role groups, scopes, copying role groups, and adding scopes to role groups, see the following topics:

- Manage role groups
- Create a regular or exclusive scope
- Understanding management role groups
- Understanding management role scopes

Installing Exchange 2013 into an existing Exchange 2007

organization

Exchange 2013 includes Role Based Access Control (RBAC) permissions that replace the Active Directory access control entry (ACE)-based authorization model used in Microsoft Exchange Server 2007. RBAC is the authorization mechanism used for most of the management of Exchange 2013. This mechanism includes the following management areas:

- Exchange Management Shell
- Exchange admin center (EAC)
- Exchange Web Services

For more information about how to plan coexistence between Exchange 2013 and Exchange 2007, see Upgrade from Exchange 2007 to Exchange 2013.

Looking for management tasks related to permissions? Check out Permissions.

Exchange 2013 permissions

The Exchange 2013 RBAC permissions model consists of management role groups assigned one of several management roles. Management roles contain permissions that enable administrators to perform tasks in the Exchange organization. Administrators are added as members of the role groups and are granted all the permissions that the roles provide. The following table provides an example of the role groups, some of the roles assigned to role groups, and a description of the kind of user who might be a member of the role group.

Examples of role groups and roles in Exchange 2013

Management role group	Management roles	Members of this role group
Organization Management	The following roles are some of the roles assigned to this role group: • Address Lists • Exchange Servers • Journaling • Mail Recipients • Public Folders	Users who manage the entire Exchange 2013 organization should be members of this role group. With some exceptions, members of this role group can manage nearly any aspect of the Exchange 2013 organization. By default, the user account used to prepare Active Directory for Exchange 2013 is a member of this role group.

		For more information about this role group and for a complete list of roles assigned to this role group, see Organization Management.
View Only Organization Management	The following roles are assigned to this role group: • Monitoring • View-Only Configuration • View-Only Recipients	Users who view the configuration of the entire Exchange 2013 organization should be members of this role group. These users must be able to view server configuration and recipient information, and perform monitoring functions without the ability to change organization or recipient configuration. For more information about this role group, see View-only Organization Management.
Recipient Management	The following roles are assigned to this role group: • Distribution Groups • Mail Enabled Public Folders • Mail Recipient Creation • Mail Recipients • Message Tracking • Migration • Move Mailboxes • Recipient Policies	Users who manage recipients such as mailboxes, contacts, and distribution groups in the Exchange 2013 organization should be members of this role group. These users can create recipients, modify or delete existing recipients, or move mailboxes. For more information about this role group and for a

		complete list of roles assigned to this role group, see Recipient Management.
Server Management	The following roles are some of the roles assigned to this role group: • Databases • Exchange Connectors • Exchange Servers • Receive Connectors • Transport Queues	Users who manage Exchange server configuration such as Receive connectors, certificates, databases, and virtual directories should be members of this role group. These users can modify Exchange server configuration, create databases, and restart and manipulate transport queues. For more information about this role group and for a complete list of roles assigned to this role group, see Server Management.
Discovery Management	The following roles are assigned to this role group: • Legal Hold • Mailbox Search	Users who perform searches of mailboxes to support legal proceedings or to configure legal holds should be members of this role group. This is an example of a role group that might contain non-Exchange administrators, such as personnel in the legal department. Legal personnel can perform their tasks without intervention from Exchange administrators.

		For more information about this role group and for a complete list of roles assigned to this role group, see Discovery Management.

This table shows that Exchange 2013 provides a granular level of control over the permissions that you grant to your administrators. You can choose among 12 role groups in Exchange 2013. For a complete list of role groups and the permissions that they provide, see Built-in role groups.

Because Exchange 2013 provides many role groups and because further customization is possible by creating role groups that have different role combinations, the manipulation of access control lists (ACLs) on Active Directory objects is no longer necessary and has no effect. ACLs are no longer used to apply permissions to individual administrators or groups in Exchange 2013. All tasks, such as an administrator creating a mailbox or a user accessing a mailbox, are managed by RBAC. RBAC authorizes the task, and then Exchange performs the task on behalf of the user if allowed. Exchange performs the task in the Exchange Trusted Subsystem universal security group (USG). With some exceptions, all the ACLs on objects in Active Directory that Exchange 2010 has to access are granted to the Exchange Trusted Subsystem USG. This is a fundamental change from how permissions are handled in Exchange 2007.

The permissions granted to a user in Active Directory are separate from the permissions granted to the user by RBAC when that user is using the Exchange 2013 management tools.

For more information about RBAC, see Understanding Role Based Access Control.

Exchange 2007 permissions

The Exchange 2007 administrative model leverages Active Directory forests to define security boundaries. There is no isolation of security permissions within a specific forest. Forest owners and enterprise administrators can always gain access to all resources in any domain. In Exchange 2007, you may have to grant enterprise administrator rights and top-level domain administrator rights on a temporary basis only.

Exchange 2007 provides the following predefined administrator roles:

- **Exchange Organization Administrator role** This role grants permissions to control all aspects of the Exchange 2007 organization. Additionally, an administrator who has this role can grant permissions to other Exchange administrators. This role is granted to the account that you use to install Exchange 2007.
- **Exchange View-Only Administrator role** This role grants permissions to view Exchange configuration. However, an administrator who has this role can't modify objects in the Exchange 2007 organization.
- **Exchange Recipient Administrator role** This role grants permissions to manage e-mail

recipients. An administrator who has this role can modify Exchange-related items for users, groups, contacts, and distribution groups.

- **Exchange Server Administrator role** This role grants permissions to manage a specific server. However, this role doesn't grant permissions to perform actions that have a global impact on the Exchange 2007 organization.
- **Exchange Public Folder Administrator role** This role was added in Exchange 2007 Service Pack 1**.** This role grants permissions to manage public folders in the Exchange 2007 organization.

This permissions model uses USGs for all roles except for the Exchange Server Administrator role. When you run the Exchange 2007 **Setup /PrepareAD** command, the Setup program creates the USGs in the root domain and gives a forest-wide scope to the USGs. The USGs are assigned ACLs to manage Exchange objects throughout Active Directory.

In Exchange 2007, you can separate administrators by assigning them various roles. The permissions are assigned directly either to the user or to the USG of which the user is a member. Any actions performed by the user are performed in the context of that user's Active Directory account. The following table lists the Exchange 2007 administrator roles together with their Exchange-related permissions.

Exchange 2007 administrator roles

Administrator role	Members	Member of	Exchange permissions
Exchange Organization Administrator	The Administrator account or the account used to install the first Exchange 2007 server	Exchange Recipient Administrator Administrators local group of <*server name*>	Full control of the Microsoft Exchange container in Active Directory
Exchange View-Only Administrator	Exchange Recipient Administrators Exchange Server Administrators (<*server name*>)	Exchange Recipient Administrators Exchange Server Administrators	Read access to the Microsoft Exchange container in Active Directory Read access to all the Windows domains that have Exchange recipients
Exchange Recipient Administrator	Exchange Organization Administrators	Exchange View-Only Administrators	Full control of Exchange properties on

			Active Directory user objects
Exchange Server Administrator	Exchange Organization Administrators	Exchange View-Only Administrators Administrators local group of *<server name>*	Full control of Exchange *<server name>*
Exchange Server	Each Exchange 2007 computer account	Exchange View-Only Administrators	Special
Exchange Public Folder Administrator	Exchange Organization Administrators	Exchange View-Only Administrators	Full control to manage all public folders (granted the Create top level public folder extended right)

If you need to make more granular permission assignments, you can modify the ACLs on individual Exchange 2007 objects, such as address lists or databases. You must add the user or security group of which the user is a member directly to the ACL. Then, the actions are performed in the context of the particular user.

For more information about how to manage permissions in Exchange 2007, see Configuring Permissions in Exchange Server 2007.

Exchange 2013 and Exchange 2007 coexistence permissions

Because the permissions models for Exchange 2013 and for Exchange 2007 differ, Exchange 2013 permission assignments are separate from Exchange 2007 permission assignments. This is true even if both versions of Exchange are installed in the same forest. Without additional configuration, Exchange 2013 administrators don't have the required permissions to manage Exchange 2007-based servers, and Exchange 2007 administrators don't have the required permissions to manage Exchange 2013-based servers. You should consider the following questions:

- Do you want to grant Exchange 2013 administrators access to manage Exchange 2007 servers?
- Do you want to grant Exchange 2007 administrators access to manage Exchange 2013 servers?
- Do you want to customize Exchange 2013 permissions so that they match any customizations that have been made to Exchange 2007 ACLs?

Granting Exchange 2013 permissions to Exchange 2007 administrators

If you want Exchange 2007 administrators to administer Exchange 2013 servers, the Exchange 2007 administrators must be added as members of one or more Exchange 2013 role groups. You can add either users or USGs to role groups. The permissions granted to the role groups are then applied to the users or the USGs that you add as members.

Important:
If you use domain local or global Active Directory security groups, you must change them to USGs if you want to add them as members of an Exchange 2013 role group. Exchange 2013 supports only USGs.

The following table describes the mapping between Exchange 2007 administrator roles and Exchange 2013 role groups.

Exchange 2007 administrator roles and Exchange 2010 role groups

Exchange 2007 administrator role	Exchange 2013 role group
Exchange Organization Administrator	Organization Management
Exchange Recipient Administrator	Recipient Management
Exchange Server Administrator	Server Management
Exchange View-Only Administrator	View Only Organization Management
Exchange Server	No equivalent role group in Exchange 2013 (For more information about how to create custom role groups, see Manage role groups.)
Exchange Public Folder Administrator	Public Folder Management

If all your Exchange 2007 administrators are members of one of the Exchange 2007 administrative roles, you can add the members of each of the administrative groups to their equivalent Exchange 2013 role group. For example, if you want to give all Exchange 2007 organization administrators full access to Exchange 2013 objects, add the Exchange Organization Administrators USG to the Organization Management role group.

For more information about how to add users and USGs to role groups, see Manage role group members.

If you modify ACLs on Exchange 2007 objects to grant more granular permissions to Exchange 2007 administrators, and if you want to assign similar permissions to Exchange 2013 servers to those administrators, follow these steps:

1. Review the ACL customizations that have been made to the Exchange 2007 objects, and locate

the administrators who have been granted permissions to each object.
2. Categorize each Exchange 2007 object. For example, determine whether the object is a database, server, or recipient object.
3. Map the objects to the corresponding Exchange 2013 role group. For a list of built-in role groups, see Built-in role groups.
4. Add the USGs or users for each kind of object to the corresponding Exchange 2013 role groups. For more information about how to add users and USGs to role groups, see Manage role group members.

After you complete these steps, the Exchange 2007 administrators will be members of the specific role group that's mapped to the appropriate Exchange 2013 objects. The Exchange 2007 administrators can use the Exchange 2013 management tools to manage the Exchange 2013 servers and recipients.

Important:
In general, Exchange 2007 servers and recipients must be managed by using Exchange 2007 management tools, and Exchange 2013 servers and recipients must be managed by using Exchange 2013 management tools.

If the built-in role groups don't provide the specific set of permissions that you want to grant to some administrators, you can create custom role groups. When you create a custom role group, you can select which roles to add to it. You can define the specific features you want members of the role group to manage. For example, if you want administrators to manage only distribution groups, you can create a custom role group, and then select only the Distribution Groups role. After you do this, members of that custom role group can manage only distribution groups. For more information about how to create custom role groups, see Manage role groups.

If you assign selective permissions to certain Exchange 2007 objects (for example, you allow administrators to administer only specific databases), and if you want to apply the same configuration to your Exchange 2013 servers, see "Re-Creating Exchange 2007 ACL Customization Using Management Scopes in Exchange 2013" later in this topic.

Granting Exchange 2007 permissions to Exchange 2013 administrators

If you want Exchange 2013 administrators to administer Exchange 2007 servers, add the Exchange 2013 administrators to the USGs or the security group that corresponds to the particular Exchange 2007 administrator role. Alternatively, if you have customized ACL settings, add the administrators to the appropriate ACLs. Role groups are USGs, so role groups can be added directly to Exchange 2007 administrator role USGs.

After you finish, the Exchange 2013 administrators will be members of the appropriate Exchange 2007 administrator role or roles. The Exchange 2013 administrators can use the Exchange 2007 management tools to manage Exchange 2007 servers and recipients.

Re-Creating Exchange 2007 ACL customization using management scopes in Exchange 2013

In Exchange 2007, when you want to restrict who can administer a specific mailbox store, administer specific users, or control which mailbox store mailboxes are created on, you must modify the ACLs on the objects you want to restrict. Exchange 2013 provides the same capabilities, but without having to modify any ACLs. It does this by using management scopes, which are a component of RBAC.

Management scopes provide built-in scopes and custom scopes to define the objects that administrators can manage. By applying management scopes, you can define which recipients can be administered, which mailbox databases mailboxes can be created on, and which recipients or servers should be administered by a small group of administrators and by no one else.

You can create the following types of management scopes:

- **Predefined relative** Predefined relative scopes are included in Exchange 2013. You can control what a user sees and what a user modifies. For example, predefined relative scopes can control whether users see only information about themselves or information about the entire organization.
- **Recipient** Recipient scopes control which recipients an administrator can create, modify, or delete. These selections can be based on an organizational unit (OU), a recipient filter, or both. Recipient filters specify the criteria that a recipient must match to be included in the scope. For example, you might create a recipient filter scope that includes all users in a certain location or in a specific department. You can even combine OUs and recipient filters to match only users who are within a specific OU and who report to a specific manager.
- **Server** Server scopes control which servers an administrator can manage. You can specify either server lists or server filters. For server lists, you define a static list of servers that can be managed. Server filters work in the same manner as recipient filters in that you can specify the criteria that have to be matched. For example, you might create a server scope that matches all servers within a particular Active Directory site.
- **Database** Database scopes control which databases an administrator can manage. They can also control which databases mailboxes can be created on or which databases mailboxes can be moved to. Like server scopes, they can be defined as lists or as filters. For example, you might create a list or filter that allows administrators to create mailboxes on or move mailboxes to specific mailbox databases managed by a specific subsidiary.
- **Exclusive** Recipient, server, and database scopes can also be created as exclusive scopes. Exclusive scopes work in the same manner as deny access ACEs in ACLs. If anything matches an exclusive scope, only the administrators assigned an exclusive scope can manage that object. This is true even if another scope that isn't exclusive matches the same object. This is especially useful when you might want only a few, highly trusted individuals to be able to manage an executive's mailbox. Even if another regular recipient scope is broader and includes the executive's mailbox in the scope, the administrators assigned the broader, regular recipient scope won't be able to manage that executive's mailbox unless they are also assigned the exclusive scope.

Management scopes are used with management roles, management role assignments, and management role groups to control who can manage what objects and in what manner they can manage those objects. For more information, see the following topics:

- Understanding management role scopes
- Understanding exclusive scopes
- Understanding management role assignments
- Understanding management role groups
- Understanding management roles

To create the same permissions model in Exchange 2013 using management scopes that you might have defined using customized ACLs, you must inventory the ACLs that you've customized, and then create management scopes that match them. You can use the filterable properties available on recipient, server, and database objects to create management scopes that include the objects to which you want each management scope to control access. For more information about the properties that you can use with management scope filters, see Understanding management role scope filters.

For more information about how to create management scopes, see Create a regular or exclusive scope.

Manage role groups

Exchange Server 2013 > Permissions >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-08*

This topic shows you how to add, remove, copy, and view management role groups in Microsoft Exchange Server 2013. It also shows you how to add, remove, and list management roles on role groups and how to change management scopes and delegates on role groups. For more information about role groups in Exchange 2013, see Understanding management role groups.

For additional management tasks related to role groups, see Permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 to 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a role group

If you want to customize the permissions that you can assign to a group of users, create a new custom management role group.

Use the EAC to create a role group

1. In the Exchange Administration Center (EAC), navigate to **Permissions** > **Admin Roles** and then click **Add +**.
2. In the **New role group** window, provide a name for the new role group.
3. You can either select the roles that you want to be assigned to the role group and the members you want to be added to the role group now, or you can do this at another time.
4. Select the write scope that you want to apply to the new role group.
5. Click **Save** to create the role group.

Use the Shell to create a role group

To create a role group, see the Examples section in New-RoleGroup.

How do you know this worked?

To verify that you have successfully created a role group, do the following:

1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Verify that the new role group appears in the role group list and then select it.
3. Verify that members, assigned roles, and scope that you specified on the new role group are listed in the role group details pane.

Copy a role group

Use the EAC to copy a role group

If you have a role group that contains the permissions you want to grant to users, but you want to apply a different management scope, or remove or add one or two management roles without having to add all the other roles manually, you can copy the existing role group.

Important:
You can't use the EAC to copy a role group if you've used the Exchange Management Shell to configure multiple management role scopes or exclusive scopes on the role group. If you've configured multiple scopes or exclusive scopes on the role group, you must use the Shell procedures later in this topic to copy the role group. For more information about management role scopes, see Understanding management role scopes.

1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Select the role group you want to copy and then click **Copy**.

3. In the **New role group** window, provide a name for the new role group.
4. Review the roles that have been copied to the new role group. Add or remove roles as necessary.
5. Review the write scope, and change it as necessary.
6. Review the members that have been copied to the new role group. Add or remove members as necessary.
7. Click **Save** to create the role group.

Use the Shell to copy a role group with no scope

1. Store the role group that you want to copy in a variable using the following syntax.

```
$RoleGroup = Get-RoleGroup <name of role group to copy>
```

2. Create the new role group, and also add members to the role group and specify who can delegate the new role group to other users, using the following syntax.

```
New-RoleGroup <name of new role group> -Roles
$RoleGroup.Roles -Members <member1, member2, member3...> -
ManagedBy <user1, user2, user3...>
```

For example, the following commands copy the Organization Management role group, and name the new role group "Limited Organization Management". It adds the members Isabelle, Carter, and Lukas and can be delegated by Jenny and Katie.

```
$RoleGroup = Get-RoleGroup "Organization Management"
New-RoleGroup "Limited Organization Management" -Roles
$RoleGroup.Roles -Members Isabelle, Carter, Lukas -
ManagedBy Jenny, Katie
```

After the new role group is created, you can add or remove roles, change the scope of role assignments on the role, and more.

For detailed syntax and parameter information, see Get-RoleGroup and New-RoleGroup.

Use the Shell to copy a role group with a custom scope

1. Store the role group that you want to copy in a variable using the following syntax.

```
$RoleGroup = Get-RoleGroup <name of role group to copy>
```

2. Create the new role group with a custom scope using the following syntax.

```
New-RoleGroup <name of new role group> -Roles
$RoleGroup.Roles -CustomRecipientWriteScope <recipient
scope name> -CustomConfigWriteScope <configuraiton scope
name>
```

For example, the following commands copy the Organization Management role group and create a

new role group called Vancouver Organization Management with the Vancouver Users recipient scope and Vancouver Servers configuration scope.

```
$RoleGroup = Get-RoleGroup "Organization Management"
New-RoleGroup "Vancouver Organization Management" -Roles
$RoleGroup.Roles -CustomRecipientWriteScope "Vancouver
Users" -CustomConfigWriteScope "Vancouver Servers"
```

You can also add members to the role group when you create it by using the *Members* parameter as shown in Use the Shell to copy a role group with no scope earlier in this topic. For more information about management scopes, see Understanding management role scopes.

After the new role group is created, you can add or remove roles, change the scope of role assignments on the role, and perform other tasks.

For detailed syntax and parameter information, see Get-RoleGroup and New-RoleGroup.

Use the Shell to copy a role group with an OU scope

1. Store the role group that you want to copy in a variable using the following syntax.

```
$RoleGroup = Get-RoleGroup <name of role group to copy>
```

2. Create the new role group with a custom scope using the following syntax.

```
New-RoleGroup <name of new role group> -Roles
$RoleGroup.Roles -RecipientOrganizationalUnitScope <OU
name>
```

For example, the following commands copy the Recipient Management role group and create a new role group called Toronto Recipient Management that allows management of only users in the Toronto Users OU.

```
$RoleGroup = Get-RoleGroup "Recipient Management"
New-RoleGroup "Toronto Recipient Management" -Roles
$RoleGroup.Roles -RecipientOrganizationalUnitScope
"contoso.com/Toronto Users"
```

You can also add members to the role group when you create it by using the *Members* parameter as shown in Use the Shell to copy a role group with no scope earlier in this topic. For more information about management scopes, see Understanding management role scopes.

After the new role group is created, you can add or remove roles, change the scope of role assignments on the role, and more.

For detailed syntax and parameter information, see Get-RoleGroup and New-RoleGroup.

How do you know this worked?

To verify that you have successfully copied a role group, do the following:

1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Verify that the copied role group appears in the role group list, and then select it.
3. Verify that members, assigned roles, and scope that you specified on the copied role group are listed in the role group details pane.

Remove a role group

If you no longer need a role group you created, you can remove it. When you remove a role group, the management role assignments between the role group and the management roles are deleted. The management roles aren't deleted. If a user depended on the role group for access to a feature, the user will no longer have access to the feature. You can't remove built-in role groups.

Use the EAC to remove a role group

1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Select the role group you want to remove and then click **Delete** 🗑.
3. Verify that you want to remove the selected role group, and if so, respond **Yes** to the warning.

Use the Shell to remove a role group

To remove a role group, see the Examples section in Remove-RoleGroup.

View role groups

You can view either a list of role groups or the detailed information about a specific role group that exists in your organization.

Use the EAC to view a list of role groups and role group details

1. In the EAC, navigate to **Permissions** > **Admin Roles**. All of the role groups in your organization are listed here.
2. Select a role group to view the members, assigned roles, and scope that are configured on the role group.

Use the Shell to view a list of role groups and role group details

To view a list of role groups, see the Examples section in Get-RoleGroup.

Add a role to a role group

Adding a management role to a role group is the best and simplest way to grant permissions to a group of administrators or specialist users. If you want to give users that are members of a role group the ability to manage a feature, you add the management role that manages the feature to the role group. After the role is added, the members of the role group are granted the permissions

provided by the role.

Use the EAC to add a management role to a role group

Important:
You can't use the EAC to add roles to a role group if you've used the Shell to configure multiple management role scopes or exclusive scopes on the role group. If you've configured multiple scopes or exclusive scopes on the role group, you must use the Shell procedures later in this topic to add roles to the role group. For more information about management role scopes, see Understanding management role scopes.

1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Select the role group you want to add a role to, and then click **Edit** .
3. In the **Roles** section, select the roles you want to add to the role group.
4. When you've finished adding roles to the role group, click **Save**.

Use the Shell to create a role assignment with no scope

You can create a role assignment with no scope between a role and a role group. When you do this, the implicit read and implicit write scopes of the role apply.

Use the following syntax to assign a role without any scope to a role group. A role assignment name is created automatically if you don't specify one.

```
New-ManagementRoleAssignment -SecurityGroup <role group
name> -Role <role name>
```

This example assigns the Transport Rules management role to the Seattle Compliance role group.

```
New-ManagementRoleAssignment -SecurityGroup "Seattle
Compliance" -Role "Transport Rules"
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Use the Shell to create a role assignment with a predefined scope

If a predefined scope meets your business requirements, you can apply that scope to the role assignment rather than create a new one. For a list of predefined scopes and their descriptions, see Understanding management role scopes.

For more information about role assignments, see Understanding management role assignments.

Use the following syntax to assign a role to a role group with a predefined scope. A role assignment name is created automatically if you don't specify one.

```
New-ManagementRoleAssignment -SecurityGroup <role group
name> -Role <role name> -RecipientRelativeWriteScope <
MyGAL | MyDistributionGroups | Organization | Self >
```

This example assigns the Message Tracking role to the Enterprise Support role group and applies the Organization predefined scope.

```
New-ManagementRoleAssignment -SecurityGroup "Enterprise Support" -Role "Message Tracking" -RecipientRelativeWriteScope Organization
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Use the Shell to create a role assignment with a recipient filter-based scope

If you created a recipient filter-based scope, you need to include the scope in the command used to assign the role to a role group by using the *CustomRecipientWriteScope* parameter.

You can also include a configuration write scope when you create a role assignment that has a recipient write scope.

For more information about role assignments and scopes, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Use the following syntax to assign a role to a role group with a recipient filter-based scope. A role assignment name is created automatically if you don't specify one.

```
New-ManagementRoleAssignment -SecurityGroup <role group name> -Role <role name> -CustomRecipientWriteScope <role scope name>
```

This example assigns the Message Tracking role to the Seattle Recipient Admins role group and applies the Seattle Recipients scope.

```
New-ManagementRoleAssignment -SecurityGroup "Seattle Recipient Admins" -Role "Message Tracking" -CustomRecipientWriteScope "Seattle Recipients"
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Use the Shell to create a role assignment with a configuration scope

If you created a server or database configuration filter or list-based scope, you need to include the scope in the command used to assign the role to a role group by using the *CustomConfigWriteScope* parameter.

You can also include a recipient write scope when you create a role assignment that has a configuration write scope.

For more information about role assignments and management scopes, see the following topics:

- Understanding management role assignments

- Understanding management role scopes

Use the following syntax to assign a role to a role group with a configuration scope. A role assignment name is created automatically if you don't specify one.

```
New-ManagementRoleAssignment -SecurityGroup <role group name> -Role <role name> -CustomConfigWriteScope <role scope name>
```

This example assigns the Databases role to the Seattle Server Admins role group and applies the Seattle Servers scope.

```
New-ManagementRoleAssignment -SecurityGroup "Seattle Server Admins" -Role "Databases" -CustomConfigWriteScope "Seattle Servers"
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Use the Shell to create a role assignment with an OU scope

If you want to scope a role's write scope to an OU, you can specify the OU in the *RecipientOrganizationalUnitScope* parameter directly.

For more information about role assignments and management scopes, see the following topics:
- Understanding management role assignments
- Understanding management role scopes

Use the following command to assign a role to a role group and restrict the write scope of a role to a specific OU. A role assignment name is created automatically if you don't specify one.

```
New-ManagementRoleAssignment -SecurityGroup <role group name> -Role <role name> -RecipientOrganizationalUnitScope <OU>
```

This example assigns the Mail Recipients role to the Seattle Recipient Admins role group and scopes the assignment to the Sales\Users OU in the Contoso.com domain.

```
New-ManagementRoleAssignment -SecurityGroup "Seattle Recipient Admins" -Role "Mail Recipients" -RecipientOrganizationalUnitScope contoso.com/sales/users
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

How do you know this worked?

To verify that you have successfully added roles to a role group, do the following:
1. In the EAC, navigate to **Permissions** > **Admin Roles**.

2. Select the role group you added roles to. In the role group details pane, verify that the roles that you added are listed.

Remove a role from a role group

Removing a role from a management role group is the best and simplest way to revoke permissions granted to a group of administrators or specialist users. If you don't want administrators or specialist users to have permissions to manage a feature, you remove the management role from the management role group that manages the permissions. After the role is removed, the members of the role group will no longer have permissions to manage the feature.

Note:

Some role groups, such as the Organization Management role group, restrict what roles can be removed from a role group. For more information, see Understanding management role groups.

If an administrator is a member of another role group that contains management roles that grants permissions to manage the feature, you need to either remove the administrator from the other role groups, or remove the role that grants permissions to manage the feature from the other role groups.

Use the EAC to remove a management role from a role group

Important:

You can't use the EAC to remove roles from a role group if you've used the Shell to configure multiple scopes or exclusive scopes on the role group. If you've configured multiple scopes or exclusive scopes on the role group, you must use the Shell procedures later in this topic to remove roles from the role group. For more information about management role scopes, see Understanding management role scopes.

1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Select the role group you want to remove a role from, and then click **Edit** .
3. In the **Roles** section, select the roles you want to remove from the role group.
4. When you've finished removing roles from the role group, click **Save**.

Use the Shell to remove a role from a role group

You can remove roles from role groups by retrieving the associated management role assignment using the **Get-ManagementRoleAssignment** cmdlet and then piping the role assignment returned to the **Remove-ManagementRoleAssignment** cmdlet. Unless you want to remove both delegating and regular role assignments at the same time, specify the *Delegating* parameter to specify whether you want to remove regular or delegating role assignments.

For more information about regular and delegating role assignments, see Understanding management role assignments.

This procedure uses pipelining. For more information about pipelining, see Pipelining.

To remove a role from a role group, use the following syntax.

```
Get-ManagementRoleAssignment -RoleAssignee <role group
name> -Role <role name> -Delegating <$true | $false> |
Remove-ManagementRoleAssignment
```

This example removes the Distribution Groups role, which enables administrators to manage distribution groups, from the Seattle Recipient Administrators role group. Because we want to remove the role assignment that provides permissions to manage distribution groups, the *Delegating* parameter is set to `$False`, which returns only regular role assignments.

```
Get-ManagementRoleAssignment -RoleAssignee "Seattle
Recipient Administrators" -Role "Distribution Groups" -
Delegating $false | Remove-ManagementRoleAssignment
```

For detailed syntax and parameter information, see Remove-ManagementRoleAssignment.

How do you know this worked?

To verify that you have successfully removed roles from a role group, do the following:

1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Select the role group you removed roles from. In the role group details pane, verify that the roles that you removed are no longer listed.

Change a role group's scope

The management role assignments between a role group and a role contain management scopes, which determine what objects are made available to members of that role group. By changing the write scope on a role group, you can change what objects are made available to role group members to create, change, or remove. You can't change the read scope on a role group.

Exchange 2013 includes scopes that are applied by default to role assignments when no custom scopes are created. If you want to use a custom scope with a role assignment on a role group, you must create one first. For more information about creating custom scopes, which is an advanced task, see Create a regular or exclusive scope.

For more information about management role scopes and assignments in Exchange 2013, see the following topics:

- Understanding management role scopes
- Understanding management role assignments

Use the EAC to change the scope on a role group

When you use the EAC to change the scope on a role group, you're actually changing the scope on all the role assignments between the role group and each of the management roles assigned to the role group. If you want to change the scope on specific role assignments, you must use the Shell procedures later in this topic.

Important:

You can't use the EAC to manage scopes on role assignments between roles and a role group if you've used the Shell to configure multiple scopes or exclusive scopes on those role assignments. If you've configured multiple scopes or exclusive scopes on those role assignments, you must use the Shell procedures later in this topic to manage scopes. For more information about management role scopes, see Understanding management role scopes.

1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Select the role group you want to change the scope on, and then click **Edit** .
3. Select one of the two following **Write scope** options:
   - A write scope from the drop-down box, where you can select either the default write scope or a custom write scope.
   - **Organizational unit** Select this option and provide an organizational unit (OU) if you want to scope this role group to an OU.
4. Click **Save** to save the changes to the role group.

Use the Shell to change the scope of all role assignments on a role group at the same time

Role assignments between the role group and the roles assigned to it can use the implicit scope obtained from the roles themselves, the same custom scope, or different custom scopes. For more information about role assignments, see Understanding management role assignments.

The scopes on the role assignments are managed using the **Set-ManagementRoleAssignment** cmdlet. You can't manage scopes using the **Set-RoleGroup** cmdlet.

To change the scope of all the role assignments between a role group and a set of management roles at the same time, you need to first retrieve the role assignments on the role group, and then set the new scope on each of the assignments. You can do this by using the **Get-ManagementRoleAssignment** cmdlet to retrieve the role assignments, and then pipe them to the **Set-ManagementRoleAssignment** cmdlet.

This procedure uses the concepts of pipelining and the *WhatIf* switch. For more information, see the following topics:

- Pipelining
- WhatIf, Confirm, and ValidateOnly switches

To set the scope on all of the role assignments on a role group at the same time, use the following syntax.

```
Get-ManagementRoleAssignment -RoleAssignee <name of role
group> | Set-ManagementRoleAssignment -
CustomRecipientWriteScope <recipient scope name> -
CustomConfigWriteScope <configuration scope name> -
RecipientRelativeScopeWriteScope < MyDistributionGroups |
Organization | Self> -ExclusiveRecipientWriteScope
<exclusive recipient scope name> -ExclusiveConfigWriteScope
```

```
<exclusive configuration scope name> -
RecipientOrganizationalUnitScope <organizational unit>
```

You use only the parameters you need to configure the scope you want to use. For example, if you want to change the recipient scope for all role assignments on the Sales Recipient Management role group to Direct Sales Employees, use the following command.

```
Get-ManagementRoleAssignment -RoleAssignee "Sales Recipient
Management" | Set-ManagementRoleAssignment -
CustomRecipientWriteScope "Direct Sales Employees"
```

Note:

You can use the *WhatIf* switch to verify that only the role assignments you want to change are changed. Run the preceding command with the *WhatIf* switch to verify the results, and then remove the *WhatIf* switch to apply the changes.

For more information about changing management role assignments, see Change a role assignment.

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

Use the Shell to change the scope of individual role assignments on a role group

Role assignments between the role group and the roles assigned to it can use the implicit scope obtained from the roles themselves, the same custom scope, or different custom scopes. For more information about role assignments, see Understanding management role assignments.

The scopes on the role assignments are managed using the **Set-ManagementRoleAssignment** cmdlet. You can't manage scopes using the **Set-RoleGroup** cmdlet.

This procedure uses the concepts of pipelining and the **Format-List** cmdlet. For more information, see the following topics:

- Pipelining
- Working with command output

To change the scope on a role assignment between a role group and a management role, you first find the name of the role assignment, and then set the scope on the role assignment.

1. To find the names of all the role assignments on a role group, use the following command. By piping the management role assignments to the **Format-List** cmdlet, you can view the full name of the assignment.

```
Get-ManagementRoleAssignment -RoleAssignee <role group
name> | Format-List Name
```

2. Find the name of the role assignment you want to change. Use the name of the role assignment in the next step.
3. To set the scope on an individual assignment, use the following syntax.

```
Set-ManagementRoleAssignment <role assignment name> -
CustomRecipientWriteScope <recipient scope name> -
CustomConfigWriteScope <configuration scope name> -
RecipientRelativeScopeWriteScope < MyDistributionGroups |
Organization | Self> -ExclusiveRecipientWriteScope
<exclusive recipient scope name> -ExclusiveConfigWriteScope
<exclusive configuration scope name> -
RecipientOrganizationalUnitScope <organizational unit>
```

You use only the parameters you need to configure the scope you want to use. For example, if you want to change the recipient scope for the Mail Recipients_Sales Recipient Management role assignment to All Sales Employees, use the following command.

```
Set-ManagementRoleAssignment "Mail Recipients_Sales
Recipient Management" -CustomRecipientWriteScope "All Sales
Employees"
```

For more information about changing management role assignments, see Change a role assignment.

For detailed syntax and parameter information, see Set-ManagementRoleAssignment.

How do you know this worked?

To verify that you have successfully changed the scope of a role assignment on a role group, do the following:

- If you used the EAC to configure the scope on the role group, do the following:
  1. In the EAC, navigate to **Permissions**> **Admin Roles**. All the role groups in your organization are listed here.
  2. Select a role group to view the scope that's configured on the role group.
- If you used the Shell to configure the scope on the role group, do the following:
  1. Run the following command in the Shell.

```
Get-ManagementRoleAssignment -RoleAssignee <role group
name> | Format-Table *WriteScope
```

  2. Verify that the write scope on the role assignments has been changed to the scope you specified.

Add or remove a role group delegate

Role group delegates are users or universal security groups (USGs) that can add or remove members from a role group or change the properties of a role group. By adding or removing role group delegates, you can control who is allowed to manage a role group.

Important:

After you add a delegate to a role group, the role group can only be managed by the delegates on the role group, or by users who are assigned, either directly or indirectly, the Role Management management role.
If a user is assigned, either directly or indirectly, the Role Management role and isn't added as a delegate of the role group, the user must use the *BypassSecurityGroupManagerCheck* switch on the **Add-RoleGroupMember**, **Remove-RoleGroupMember**, **Update-RoleGroupMember**, and **Set-RoleGroup** cmdlets to manage a role group.

Note:

You can't use the EAC to add a delegate to a role group.

Use the Shell to add a delegate to a role group

To change the list of delegates on a role group, you use the *ManagedBy* parameter on the **Set-RoleGroup** cmdlet. The *ManagedBy* parameter overwrites the entire delegate list on the role group. If you want to add delegates to the role group rather than replace the entire list of delegates, use the following steps:

1. Store the role group in a variable using the following command.

```
$RoleGroup = Get-RoleGroup <role group name>
```

2. Add the delegate to the role group stored in the variable using the following command.

```
$RoleGroup.ManagedBy += (Get-User <user to add>).Identity
```

Note:

Use the **Get-Group** cmdlet if you want to add a USG.

3. Repeat Step 2 for each delegate you want to add.
4. Apply the new list of delegates to the actual role group using the following command.

```
Set-RoleGroup <role group name> -ManagedBy
$RoleGroup.ManagedBy
```

This example adds the user David Strome as a delegate on the Organization Management role group.

```
$RoleGroup = Get-RoleGroup "Organization Management"
$RoleGroup.ManagedBy += (Get-User "David Strome").Identity
Set-RoleGroup "Organization Management" -ManagedBy
$RoleGroup.ManagedBy
```

For detailed syntax and parameter information, see Set-RoleGroup.

Use the Shell to remove a delegate from a role group

To change the list of delegates on a role group, you use the *ManagedBy* parameter on the **Set-RoleGroup** cmdlet. The *ManagedBy* parameter overwrites the entire delegate list on the role group. If you want to remove delegates from the role group rather than replace the entire list of delegates, use the following steps:

1. Store the role group in a variable using the following command.

```
$RoleGroup = Get-RoleGroup <role group name>
```

2. Remove the delegate from the role group stored in the variable using the following command.

```
$RoleGroup.ManagedBy -= (Get-User <user to
remove>).Identity
```

Note:
Use the **Get-Group** cmdlet if you want to remove a USG.

3. Repeat Step 2 for each delegate you want to remove.
4. Apply the new list of delegates to the actual role group using the following command.

```
Set-RoleGroup <role group name> -ManagedBy
$RoleGroup.ManagedBy
```

This example removes the user David Strome as a delegate on the Organization Management role group.

```
$RoleGroup = Get-RoleGroup "Organization Management"
$RoleGroup.ManagedBy -= (Get-User "David Strome").Identity
Set-RoleGroup "Organization Management" -ManagedBy
$RoleGroup.ManagedBy
```

For detailed syntax and parameter information, see Set-RoleGroup.

How do you know this worked?

To verify that you have successfully changed the delegate list on a role group, do the following:

1. In the Shell, run the following command.

```
Get-RoleGroup <role group name> | Format-List ManagedBy
```

2. Verify that the delegates listed on the *ManagedBy* property include only the delegates that should be able to manage the role group.

Manage role group members

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-08*

This topic shows you how to add, remove and view members of a management role group in Microsoft Exchange Server 2013. For more information about role groups in Exchange 2013, see Understanding management role groups.

For additional management tasks related to role groups, see Permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Add members to a role group

To give a user the permissions that are granted by a role group, you need to add the user, or a universal security group (USG), or another role group that the user is a member of, as a member of the role group.

Use the EAC to add members to a role group

1. In the Exchange Administration Center (EAC), navigate to **Permissions** > **Admin Roles**.
2. Select the role group you want to add members to, and then click **Edit** .
3. In the **Members** section, click **Add +**.
4. Select the users, USGs, or other role groups you want to add to the role group, click **Add**, and then click **OK**.
5. Click **Save** to save the changes to the role group.

Use the Shell to add members to a role group

To add a role group member, see the Examples section in Add-RoleGroupMember.

To add multiple role group members or to replace the role group membership entirely, see the Examples section in Update-RoleGroupMember.

How do you know this worked?

To verify that you have successfully added one or more members to a role group, do the following:
1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Select the role group you added members to.
3. In the role group details pane, verify that the members you added are listed.

Remove members from a role group

To remove the permissions granted by a role group from a user, you need to remove the user, or the universal security group (USG) the user is a member of, from the role group's membership.

Use the EAC to remove members from a role group

1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Select the role group you want to remove members from, and then click **Edit** .
3. In the **Members** section, select the members you want to remove, click **Remove** —, and then click **Save**.

Use the Shell to remove members from a role group

To remove a role group member, see the Examples section in Remove-RoleGroupMember.

To remove multiple role group members or to replace the role group membership entirely, see the Examples section in Update-RoleGroupMember.

How do you know this worked?

To verify that you have successfully removed one or more members to a role group, do the following:
1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Select the role group you removed members from.
3. In the role group details pane, verify that the members you removed are no longer listed.

View the members of a role group

The members of a role group are granted the permissions provided by the management roles assigned to the role group. You can view the members of a role group to see which users, universal security groups (USG), or other role groups are granted permissions by the role group you specify.

Use the EAC to view the members of a role group

1. In the EAC, navigate to **Permissions** > **Admin Roles**.
2. Select the role group you want to view the members of.
3. In the role group details pane, view the members in the role group details pane.

Use the Shell to view the members of a role group

To view the members of a role group, see the “Examples” section in Get-RoleGroupMember.

Manage linked role groups

Exchange Server 2013 > Permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-09*

You can use a linked management role group to enable members of a universal security group (USG) in a foreign Active Directory forest to manage a Microsoft Exchange Server 2013 organization in a resource Active Directory forest. By associating a USG in a foreign forest with a linked role group, the members of that USG are granted the permissions provided by the management roles assigned to the linked role group. For more information about linked role groups, see Understanding management role groups.

To create and configure linked role groups, you need to use the **New-RoleGroup** and **Set-RoleGroup** cmdlets. For detailed syntax and parameter information, see the following topics:

- New-RoleGroup
- Set-RoleGroup

For additional management tasks related to role groups, see Permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 to 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.
- You can’t use the Exchange Administration Center (EAC) to create or configure linked role groups. You must use the Exchange Management Shell.
- At a minimum, configuring a linked role group requires that a one-way trust is established between the resource Active Directory forest in which the linked role group will reside, and the foreign Active Directory forest where the users or USGs reside. The resource forest must trust the foreign forest.
- You must have the following information about the foreign Active Directory forest:
  - **Credentials** You must have a user name and password that can access the foreign Active Directory forest. This information is used with the *LinkedCredential* parameter on the **New-RoleGroup** and **Set-RoleGroup** cmdlets.
  - **Domain controller** You must have the fully qualified domain name (FQDN) of an Active Directory domain controller in the foreign Active Directory forest. This information is used with the *LinkedDomainController* parameter on the **New-RoleGroup** and **Set-RoleGroup** cmdlets.

- **Foreign USG** You must have the full name of a USG in the foreign Active Directory forest that contains the members you want to associate with the linked role group. This information is used with the *LinkedForeignGroup* parameter on the **New-RoleGroup** and **Set-RoleGroup** cmdlet.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a linked role group

Use the Shell to create a linked role group with no scope

To create a linked role group and assign management roles to the linked role group, do the following:

1. Store the foreign Active Directory forest credentials in a variable.

```
$ForeignCredential = Get-Credential
```

2. Create the linked role group using the following syntax.

```
New-RoleGroup <role group name> -LinkedForeignGroup <name
of foreign USG> -LinkedDomainController <FQDN of foreign
Active Directory domain controller> -LinkedCredential
$ForeignCredential -Roles <role1, role2, role3...>
```

3. Add or remove members to or from the foreign USG using Active Directory Users and Computers on a computer in the foreign Active Directory forest.

This example does the following:

- Retrieves the credentials for the users.contoso.com foreign Active Directory forest. These credentials are used to connect to the DC01.users.contoso.com domain controller in the foreign forest.
- Creates a linked role group called Compliance Role Group in the resource forest where Exchange 2013 is installed.
- Links the new role group to the Compliance Administrators USG in the users.contoso.com foreign Active Directory forest.
- Assigns the Transport Rules and Journaling management roles to the new linked role group.

```
$ForeignCredential = Get-Credential
New-RoleGroup "Compliance Role Group" -LinkedForeignGroup
```

```
"Compliance Administrators" -LinkedDomainController
DC01.users.contoso.com -LinkedCredential $ForeignCredential
-Roles "Transport Rules", "Journaling"
```

Use the Shell to create a linked role group with a custom management scope

You can create linked role groups with custom recipient management scopes, custom configuration management scopes, or both. To create a linked role group and assign management roles with custom scopes to it, do the following:

1. Store the foreign Active Directory forest credentials in a variable.

```
$ForeignCredential = Get-Credential
```

2. Create the linked role group using the following syntax.

```
New-RoleGroup <role group name> -LinkedForeignGroup <name
of foreign USG> -LinkedDomainController <FQDN of foreign
Active Directory domain controller> -CustomConfigWriteScope
<name of configuration scope> -CustomRecipientWriteScope
<name of recipient scope> -LinkedCredential
$ForeignCredential -Roles <role1, role2, role3...>
```

3. Add or remove members to or from the foreign USG using Active Directory Users and Computers on a computer in the foreign Active Directory forest.

This example does the following:

- Retrieves the credentials for the users.contoso.com foreign Active Directory forest. These credentials are used to connect to the DC01.users.contoso.com domain controller in the foreign forest.
- Creates a linked role group called Seattle Compliance Role Group in the resource forest where Exchange 2013 is installed.
- Links the new role group to the Seattle Compliance Administrators USG in the users.contoso.com foreign Active Directory forest.
- Assigns the Transport Rules and Journaling management roles to the new linked role group with the Seattle Recipients custom recipient scope.

```
$ForeignCredential = Get-Credential
New-RoleGroup "Seattle Compliance Role Group" -
LinkedForeignGroup "Seattle Compliance Administrators" -
LinkedDomainController DC01.users.contoso.com -
LinkedCredential $ForeignCredential -
CustomRecipientWriteScope "Seattle Recipients" -Roles
"Transport Rules", "Journaling"
```

For more information about management scopes, see Understanding management role scopes.

Use the Shell to create a linked role group with an OU scope

You can create linked role groups that use an OU recipient scope. To create a linked role group and assign management roles to it with an OU scope, do the following:

1. Store the foreign Active Directory forest credentials in a variable.

```
$ForeignCredential = Get-Credential
```

2. Create the linked role group using the following syntax.

```
New-RoleGroup <role group name> -LinkedForeignGroup <name
of foreign USG> -LinkedDomainController <FQDN of foreign
Active Directory domain controller> -LinkedCredential
$ForeignCredential -RecipientOrganizationalUnitScope <OU
name> -Roles <role1, role2, role3...>
```

3. Add or remove members to or from the foreign USG using Active Directory Users and Computers on a computer in the foreign Active Directory forest.

This example does the following:

- Retrieves the credentials for the users.contoso.com foreign Active Directory forest. These credentials are used to connect to the DC01.users.contoso.com domain controller in the foreign forest.
- Creates a linked role group called Executives Compliance Role Group in the resource forest where Exchange 2013 is installed.
- Links the new role group to the Executives Compliance Administrators USG in the users.contoso.com foreign Active Directory forest.
- Assigns the Transport Rules and Journaling management roles to the new linked role group with the OU recipient scope Executives OU.

```
$ForeignCredential = Get-Credential
New-RoleGroup "Executives Compliance Role Group" -
LinkedForeignGroup "Executives Compliance Administrators" -
LinkedDomainController DC01.users.contoso.com -
LinkedCredential $ForeignCredential -
RecipientOrganizationalUnitScope "Executives OU" -Roles
"Transport Rules", "Journaling"
```

For more information about management scopes, see Understanding management role scopes.

Change the foreign USG on a linked role group

Use the Shell to change the foreign USG on a linked role group

To change the foreign USG associated with a linked role group, do the following:
1. Store the foreign Active Directory forest credentials in a variable.

```
$ForeignCredential = Get-Credential
```

2. Change the foreign USG on the existing linked role group using the following syntax.

```
Set-RoleGroup <role group name> -LinkedForeignGroup <name
of foreign USG> -LinkedDomainController <FQDN of foreign
Active Directory domain controller> -LinkedCredential
$ForeignCredential
```

This example does the following:

- Retrieves the credentials for the users.contoso.com foreign Active Directory forest. These credentials are used to connect to the DC01.users.contoso.com domain controller in the foreign forest.
- Changes the foreign USG on the Compliance Role Group role group to Regulatory Compliance Officers.

```
$ForeignCredential = Get-Credential
Set-RoleGroup "Compliance Role Group" -LinkedForeignGroup
"Regulatory Compliance Officers" -LinkedDomainController
DC01.users.contoso.com -LinkedCredential $ForeignCredential
```

Manage role assignment policies

Exchange Server 2013 > Permissions >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-09*

If you want to customize the permissions that you assign to a group of end users, create a new custom management role assignment policy. The assignment policy you create can be customized to suit your end user's specific requirements. For more information about assignment policies in Microsoft Exchange Server 2013, see Understanding management role assignment policies.

Looking for other management tasks related to managing permissions? Check out Permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Assignment policies" entry in the Role management permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Add an assignment policy

After you've created the new assignment policy, you assign users to it. For more information, see Change the assignment policy on a mailbox.

Use the EAC to create a new assignment policy

Note:
You can only create explicit assignment policies using the Exchange Administration Center (EAC). If you want to create a new default assignment policy, you must use the Exchange Management Shell. For more information, see the "Use the Shell to create a default assignment policy" section later in this topic.

1. In the EAC, navigate to **Permissions** > **User Roles** and then click **Add +**.
2. In the role assignment policy window, provide a name for the new assignment policy.
3. Select the check box next to the role or roles you want to add to the assignment policy. You can select multiple roles, including end-user roles you've added. If you select a role that has child roles, the child roles are automatically selected.
4. Click **Save** to save the changes to the assignment policy.

Use the Shell to create an explicit assignment policy

To create an explicit assignment policy that can be manually assigned to mailboxes, use the following syntax.

```
New-RoleAssignmentPolicy <assignment policy name> -Roles
<roles to assign>
```

This example creates the explicit assignment policy Limited Mailbox Configuration and assigns the `MyBaseOptions`, `MyAddressInformation`, and `MyDisplayName` roles to it.

```
New-RoleAssignmentPolicy "Limited Mailbox Configuration" -
Roles MyBaseOptions, MyAddressInformation, MyDisplayName
```

For detailed syntax and parameter information, see New-RoleAssignmentPolicy.

Use the Shell to create a default assignment policy

To create a default assignment policy assigned to new mailboxes, use the following syntax.

```
New-RoleAssignmentPolicy <assignment policy name> -Roles
<roles to assign> -IsDefault
```

This example creates the default assignment policy Limited Mailbox Configuration and assigns the `MyBaseOptions`, `MyAddressInformation`, and `MyDisplayName` roles to it.

```
New-RoleAssignmentPolicy "Limited Mailbox Configuration" -
Roles MyBaseOptions, MyAddressInformation, MyDisplayName -
IsDefault
```

For detailed syntax and parameter information, see New-RoleAssignmentPolicy.

Remove an assignment policy

If you no longer need a management role assignment policy, you can remove it.

What do you need to know before you begin?

- All users assigned the assignment policy must be changed to another assignment policy. For more information about how to change an assignment policy on a mailbox, see Change the assignment policy on a mailbox.
- All the management role assignments between the assignment policy and the assigned management roles must be removed. For more information about how to remove a role assignment from an assignment policy, see the Use the Shell to remove a role from an assignment policy section later in this topic.
- If you want to remove a default assignment policy, it must be the last assignment policy in the Exchange 2013 organization.

Use the EAC to remove an assignment policy

1. In the EAC, navigate to **Permissions** > **User Roles**.
2. Select the assignment policy you want to remove, and then click **Delete** 🗑.

Use the Shell to remove an assignment policy

To remove an assignment policy, use the following syntax.

```
Remove-RoleAssignmentPolicy <role assignment policy>
```

This example removes the New York Temporary Users assignment policy.

```
Remove-RoleAssignmentPolicy "New York Temporary Users"
```

For detailed syntax and parameter information, see Remove-RoleAssignmentPolicy.

View a list of assignment policies or assignment policy details

You can view management role assignment policies in a variety of ways, depending on the information you want and whether you're using the EAC or the Shell.

In the EAC, you can view the list of assignment policies and the roles assigned to them. In the Shell, you can view all the assignment policies in your organization, list the mailboxes assigned a specific policy, and more.

Use the EAC to view a list of assignment policies

1. In the EAC, navigate to **Permissions** > **User Roles**. All of the assignment policies in the organization are listed here.
2. To view the details of a specific assignment policy, select the assignment policy you want to view. The description and the roles assigned to the assignment policy are displayed in the details pane.

Use the Shell to view a list of assignment policies

You can view a list of all the assignment policies in your organization by not specifying any assignment policies when you run the **Get-RoleAssignmentPolicy** cmdlet.

This procedure makes use of pipelining and the **Format-Table** cmdlet. For more information about these concepts, see the following topics:

- Pipelining
- Working with command output

To return a list of all assignment policies in your organization, use the following command.

```
Get-RoleAssignmentPolicy
```

To return a list of specific properties for all the assignment policies in your organization, you can pipe the results to the **Format-Table** cmdlet and specify the properties you want in the list of results. Use the following syntax.

```
Get-RoleAssignmentPolicy | Format-Table <property 1>,
<property 2...>
```

This example returns a list of all the assignment policies in your organization and includes the **Name** and **IsDefault** properties.

```
Get-RoleAssignmentPolicy | Format-Table Name, IsDefault
```

For detailed syntax and parameter information, see Get-Mailbox or Get-RoleAssignmentPolicy.

Use the Shell to view the details of a single assignment policy

You can view the details of a specific assignment policy by using the **Get-RoleAssignmentPolicy** cmdlet and piping the output to the **Format-List** cmdlet.

This procedure makes use of pipelining and the **Format-List** cmdlet. For more information about these concepts, see the following topics:

- Pipelining
- Working with command output

To view the details of a specific assignment policy, use the following syntax.

```
Get-RoleAssignmentPolicy <assignment policy name> | Format-List
```

This example views the details about the Redmond Users - no Text Messaging assignment policy.

```
Get-RoleAssignmentPolicy "Redmond Users - no Text Messaging" | Format-List
```

For detailed syntax and parameter information, see Get-Mailbox or Get-RoleAssignmentPolicy.

Use the Shell to find the default assignment policy

You can find the default assignment policy by piping the output of the **Get-RoleAssignmentPolicy** cmdlet to the **Where** cmdlet. With the **Where** cmdlet, filter the data returned to display only the assignment policy that has its *IsDefault* property set to `$True`.

This procedure makes use of pipelining and the **Where** cmdlet. For more information about these concepts, see the following topics:

- Pipelining
- Working with command output

This example returns the default assignment policy.

```
Get-RoleAssignmentPolicy | Where { $_.IsDefault -eq $True }
```

For detailed syntax and parameter information, see Get-Mailbox or Get-RoleAssignmentPolicy.

Use the Shell to view mailboxes that are assigned a specific policy

You can find all the mailboxes assigned a specific assignment policy by piping the output of the **Get-Mailbox** cmdlet to the **Where** cmdlet. With the **Where** cmdlet, filter the data returned to display only the mailboxes that have their *RoleAssignmentPolicy* property set to the assignment

policy name you specify.

This procedure makes use of pipelining and the **Where** cmdlet. For more information about these concepts, see the following topics:

- Pipelining
- Working with command output

Use the following syntax.

```
Get-Mailbox | Where { $_.RoleAssignmentPolicy -Eq "<role assignment policy>" }
```

This example finds all the mailboxes assigned the policy Vancouver End Users.

```
Get-Mailbox | Where { $_.RoleAssignmentPolicy -Eq "Vancouver End Users" }
```

For detailed syntax and parameter information, see Get-Mailbox or Get-RoleAssignmentPolicy.

Change the default assignment policy

You can change the management role assignment policy assigned to new mailboxes that are created. Changing the default role assignment policy doesn't change the assignment policy assigned to existing mailboxes. To change the assignment policy assigned to existing mailboxes, see Change the assignment policy on a mailbox.

Note:
You can't use the EAC to change the default assignment policy. You need to use the Shell.

Use the Shell to change the default assignment policy

To change the default assignment policy, use the following syntax.

```
Set-RoleAssignmentPolicy <assignment policy name> -IsDefault
```

This example sets the Vancouver End Users assignment policy as the default assignment policy.

```
Set-RoleAssignmentPolicy "Vancouver End Users" -IsDefault
```

Important:
New mailboxes are assigned the default assignment policy even if the policy hasn't been assigned management roles. Mailboxes assigned assignment policies with no assigned management roles can't access any mailbox configuration features in Microsoft Outlook Web App.

For detailed syntax and parameter information, see Set-RoleAssignmentPolicy.

Add a role to an assignment policy

Use the EAC to add a role to an assignment policy

1. In the EAC, navigate to **Permissions** > **User Roles**.
2. Select the assignment policy you want to add one or more roles to, and then click **Edit** ✎.
3. Select the check box next to the role or roles you want to add to the assignment policy. You can select multiple roles, including end-user roles you've added. If you select a role that has child roles, the child roles are automatically selected.
4. Click **Save** to save the changes to the assignment policy.

Use the Shell to add a role to an assignment policy

To create a management role assignment between a role and an assignment policy, use the following syntax.

```
New-ManagementRoleAssignment -Name <role assignment name> -
Role <role name> -Policy <assignment policy name>
```

This example creates the role assignment Seattle Users - Voicemail between the MyVoicemail role and the Seattle Users assignment policy.

```
New-ManagementRoleAssignment -Name "Seattle Users -
Voicemail" -Role MyVoicemail -Policy "Seattle Users"
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Remove a role from an assignment policy

If you don't want end users to have permissions to manage certain features of their mailbox or distribution group, you can remove the management role that grants the permissions from the management role assignment policy to which the user is assigned. If other users are assigned the same assignment policy, they also lose the ability to manage that feature.

Use the EAC to remove a role from an assignment policy

1. In the EAC, navigate to **Permissions** > **User Roles**.
2. Select the assignment policy you want to remove one or more roles from, and then click **Edit** ✎.
3. Clear the check box next to the role or roles you want to remove from the assignment policy. If you clear the check box for a role that has child roles, the check boxes for the child roles are also cleared.
4. Click **Save** to save the changes to the assignment policy.

Use the Shell to remove a role from an assignment policy

You can remove roles from assignment policies by retrieving the associated management role

assignment using the **Get-ManagementRoleAssignment** cmdlet and then piping the role assignment returned to the **Remove-ManagementRoleAssignment** cmdlet.

For more information about regular and delegating role assignments, see Understanding management role assignments.

This procedure uses pipelining. For more information about pipelining, see Pipelining.

To remove a role from an assignment policy, use the following syntax.

```
Get-ManagementRoleAssignment -RoleAssignee <assignment policy name> -Role <role name> | Remove-ManagementRoleAssignment
```

This example removes the MyVoicemail management role, which enables users to manage their voice mail options, from the Seattle Users assignment policy.

```
Get-ManagementRoleAssignment -RoleAssignee "Seattle Users" -Role MyVoicemail | Remove-ManagementRoleAssignment
```

For detailed syntax and parameter information, see Remove-ManagementRoleAssignment.

Change the assignment policy on a mailbox

Exchange Server 2013 > Permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-08*

You can change the management role assignment policy assigned to a mailbox. When you change a mailbox's assignment policy, the change takes effect as soon as the user refreshes the connection, such as the next time they log into their mailbox or open the mailbox options page. For more information about assignment policies in Microsoft Exchange Server 2013, see Understanding management role assignment policies.

Looking for other management tasks related to permissions? Check out Permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role groups" entry in the Role management permissions

topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to change the assignment policy on a mailbox

1. In the Exchange Administration Center (EAC), navigate to **Recipients** > **Mailboxes**.
2. Select the user or resource mailbox you want to change the assignment policy on and then click **Edit** .
3. Select **Mailbox Features**.
4. In the **Role assignment policy** list, select the assignment policy you want to assign to the mailbox and then click **Save**.

Use the Shell to change the assignment policy on a mailbox

To change the assignment policy that's assigned to a mailbox, use the following syntax.

```
Set-Mailbox <mailbox alias or name> -RoleAssignmentPolicy <assignment policy>
```

This example sets the assignment policy to Unified Messaging Users on the mailbox Brian.

```
Set-Mailbox Brian -RoleAssignmentPolicy "Unified Messaging Users"
```

Use the Shell to change the assignment policy on a group of mailboxes assigned a specific assignment policy

Note:
You can't use the EAC to change the assignment policy on a group of mailboxes all at once.

This procedure makes use of pipelining, the **Where** cmdlet, and the *WhatIf* parameter. For more information about these concepts, see the following topics:

- Pipelining
- Working with command output
- WhatIf, Confirm, and ValidateOnly switches

If you want to change the assignment policy for a group of mailboxes that are assigned a specific policy, use the following syntax.

```
Get-Mailbox | Where { $_.RoleAssignmentPolicy -Eq
"<assignment policy to find>" } | Set-Mailbox -
RoleAssignmentPolicy <assignment policy to set>
```

This example finds all the mailboxes assigned to the Redmond Users - No Voicemail assignment policy and changes the assignment policy to Redmond Users - Voicemail Enabled.

```
Get-Mailbox | Where { $_.RoleAssignmentPolicy -Eq "Redmond
Users - No Voicemail" } | Set-Mailbox -RoleAssignmentPolicy
"Redmond Users - Voicemail Enabled"
```

This example includes the *WhatIf* parameter so that you can see all the mailboxes that would be changed without committing any changes.

```
Get-Mailbox | Where { $_.RoleAssignmentPolicy -Eq "Redmond
Users - No Voicemail" } | Set-Mailbox -RoleAssignmentPolicy
"Redmond Users - Voicemail Enabled" -WhatIf
```

For detailed syntax and parameter information, see Get-Mailbox or Set-Mailbox.

Create linked role groups that mirror built-in role groups

Exchange Server 2013 > Permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Using linked management role groups in Microsoft Exchange Server 2013, you can link a role group in an Exchange 2013 resource forest with a universal security group (USG) in a foreign user forest. This is useful when you want administrators with accounts in the user forest to manage the servers running Exchange in the resource forest. For more information about linked role groups, see Understanding management role groups.

By default, Exchange 2013 includes a number of built-in role groups that provide you with permissions to manage a variety of features and job functions. Each role group is tailored to provide specific permissions for each feature and job function. However, these role groups can't be linked to USGs in a foreign forest. They can only contain users and USGs from the local resource

forest. Fortunately, it's possible to replicate these built-in role groups using linked role groups.

You can re-create each built-in role group as a linked role group. All of the management roles and management scopes assigned to each role group are added to the new linked role group. For more information about management roles and scopes, see the following topics:

- Understanding management roles
- Understanding management role scopes

Looking for other management tasks related to role groups? Check out Permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- Configuring a linked role group requires a one-way trust between the resource Active Directory forest in which the linked role group will reside, and the foreign Active Directory forest where the users or USGs reside. The resource forest must trust the foreign forest.
- You must have the following information about the foreign Active Directory forest:
  - **Credentials** You must have a user name and password that can access the foreign Active Directory forest. This information is used with the *LinkedCredential* parameter on the **New-RoleGroup** cmdlet. This information is obtained by running the **Get-Credential** cmdlet. The format of the user name is *domain**username*.
  - **Domain controller** You must have the fully qualified domain name (FQDN) of an Active Directory domain controller in the foreign Active Directory forest. This information is used with the *LinkedDomainController* parameter on the **New-RoleGroup** cmdlet.
  - **Foreign USG** You must have the full name of a USG in the foreign Active Directory forest that contains the members you want to associate with the linked role group. This information is used with the *LinkedForeignGroup* parameter on the **New-RoleGroup** cmdlet.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to create linked role groups that replicate built-in role groups

Each of the following sections shows you how to re-create each role group as a linked role group. Complete the procedures in each section to re-create all of the built-in role groups as linked role

groups.

Create the Organization Management linked role group

To re-create the Organization Management role group as a linked role group, you perform a procedure that's different than the procedure used to re-create other built-in role groups. This is because the Organization Management role group has delegating role assignments between it and all of the management roles. Re-creating the delegating role assignments requires an additional step.

1. Create a USG in the foreign forest that will be linked to the Organization Management role group.
2. Store the foreign Active Directory forest credentials in a variable.

```
$ForeignCredential = Get-Credential
```

3. Store all of the roles assigned to the Organization Management role group in a variable.

```
$OrgMgmt  = Get-RoleGroup "Organization Management"
```

4. Create the Organization Management linked role group and add the roles assigned to the built-in Organization Management role group.

```
New-RoleGroup "Organization Management - Linked" -
LinkedForeignGroup <name of foreign USG> -
LinkedDomainController <FQDN of foreign Active Directory
domain controller> -LinkedCredential $ForeignCredential -
Roles $OrgMgmt.Roles
```

5. Remove all of the regular assignments between the new Organization Management linked role group and the My* end-user roles.

```
Get-ManagementRoleAssignment -RoleAssignee "Organization
Management - Linked" -Role My* | Remove-
ManagementRoleAssignment
```

6. Add delegating role assignments between the new Organization Management linked role group and all management roles.

```
Get-ManagementRole | New-ManagementRoleAssignment -
SecurityGroup "Organization Management - Linked" -
Delegating
```

This example assumes the following values are used for each parameter:

- **LinkedForeignGroup** `Organization Management Administrators`
- **LinkedDomainController** `DC01.users.contoso.com`

Using the preceding values, this example re-creates the Organization Management role group as a linked role group.

```
$ForeignCredential = Get-Credential
$OrgMgmt  = Get-RoleGroup "Organization Management"
New-RoleGroup "Organization Management - Linked" -
LinkedForeignGroup "Organization Management Administrators"
-LinkedDomainController DC01.users.contoso.com -
LinkedCredential $ForeignCredential -Roles $OrgMgmt.Roles
Get-ManagementRoleAssignment -RoleAssignee "Organization
Management - Linked" -Role My* | Remove-
ManagementRoleAssignment
Get-ManagementRole | New-ManagementRoleAssignment -
SecurityGroup "Organization Management - Linked" -
Delegating
```

Create all other linked role groups

To re-create the built-in role groups (other than the Organization Management role group) as linked role groups, use the following procedure for each group.

1. Create a USG in the foreign forest for each role group that will be linked to each new role group.
2. Store the foreign Active Directory forest credentials in a variable. You only need to do this once.

```
$ForeignCredential = Get-Credential
```

3. Retrieve a list of role groups using the following cmdlet.

```
Get-RoleGroup
```

4. For each role group, other than the Organization Management role group, do the following.

```
$RoleGroup = Get-RoleGroup <name of role group to re-
create>
New-RoleGroup "<role group name> - Linked" -
LinkedForeignGroup <name of foreign USG> -
LinkedDomainController <FQDN of foreign Active Directory
domain controller> -LinkedCredential $ForeignCredential -
Roles $RoleGroup.Roles
```

5. Repeat the preceding step for each built-in role group you want to re-create as a linked role group.

This example assumes the following values are used for each parameter:

- **LinkedDomainController** `DC01.users.contoso.com`

- **Built-in role groups to be re-created as linked role groups** `Recipient Management, Server Management`
- **Foreign group for Recipient Management linked role group** `Recipient Management Administrators`
- **Foreign group for Server Management linked role group** `Server Management Administrators`

Using the preceding values, this example re-creates the Recipient Management and Server Management role groups as linked role groups.

```
$ForeignCredential = Get-Credential
Get-RoleGroup
$RoleGroup = Get-RoleGroup "Recipient Management"
New-RoleGroup "Recipient Management - Linked" -
LinkedForeignGroup "Recipient Management Administrators" -
LinkedDomainController DC01.users.contoso.com -
LinkedCredential $ForeignCredential -Roles $RoleGroup.Roles
$RoleGroup = Get-RoleGroup "Server Management"
New-RoleGroup "Server Management - Linked" -
LinkedForeignGroup "Server Management Administrators" -
LinkedDomainController DC01.users.contoso.com -
LinkedCredential $ForeignCredential -Roles $RoleGroup.Roles
```

Other tasks

After you create linked role groups, you may also want to:

Add members to the foreign USGs using Active Directory Users and Computers in the foreign forest.

Remove members of built-in role groups. For more information, see Manage role group members.

Add, remove, or change the scope of roles on the new linked role groups. For more information, see Manage role groups.

Create additional linked role groups. For more information, see Manage linked role groups.

View effective permissions

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-09*

Permissions in Microsoft Exchange Server 2013 are granted using management roles that are assigned to management role groups, management role assignment policies, universal security

groups (USGs), or directly to users. Users are granted the permissions if they're members of the role groups or USGs, or are assigned role assignment policies.

Most permissions are granted based on role group membership or the assignment of assignment policies to end users. Although using role groups and assignment policies makes it easy to grant permissions to large numbers of users, you may not be aware of who is a member of a role group, or who has been assigned an assignment policy. This is where the *GetEffectiveUsers* switch on the **Get-ManagementRoleAssignment** cmdlet is useful. It shows you what users are granted the permissions given by a management role through the role groups, assignment policies, and USGs that are assigned to them.

The *GetEffectiveUsers* switch is used with the **Get-ManagementRoleAssignment** cmdlet when the *Role* parameter is used. By specifying this switch with a particular role, the **Get-ManagementRoleAssignment** cmdlet examines all the role assignees assigned to the role, such as role groups, assignment policies, and USGs, and lists the members of each.

Note:

The *GetEffectiveUser* switch doesn't list users that are members of a linked foreign role group. Instead of a list of users, if a linked role group is found, **All Linked Group Members** is displayed. For more information about permissions in multiple forests, see Understanding multiple-forest permissions.

For more information about management roles, role groups, and assignment policies, see Understanding Role Based Access Control. For more information about management role assignments, see Understanding management role assignments.

Looking for other management tasks related to managing permissions? Check out Permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role groups" or "Role assignment policy" entries in the Role management permissions topic.
- The procedures in this topic can only be performed in the Shell. You can't use the Exchange Administration Center (EAC) to view effective permissions.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to list all effective users

To list all the users that are granted the permissions provided by a management role, use the

following syntax.

```
Get-ManagementRoleAssignment -Role <role name> -
GetEffectiveUsers
```

This example lists all the users that are granted permissions provided by the Mail Recipients role.

```
Get-ManagementRoleAssignment -Role "Mail Recipients" -
GetEffectiveUsers
```

If you want to change what properties are returned in the list or export the list to a comma-separated value (.csv) file, see Use the Shell to customize output and display it later in this topic.

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

Use the Shell to find a specific user on a role

To find a specific user that's been granted permissions by a management role, you must use the **Get-ManagementRoleAssignment** cmdlet to retrieve a list of all effective users, and then pipe the output of the cmdlet to the **Where** cmdlet. The **Where** cmdlet filters the output and returns only the user you specified. Use the following syntax.

```
Get-ManagementRoleAssignment -Role <role name> -
GetEffectiveUsers | Where { $_.EffectiveUserName -Eq "<name
of user>" }
```

This example finds the user David Strome on the Journaling role.

```
Get-ManagementRoleAssignment -Role Journaling -
GetEffectiveUsers | Where { $_.EffectiveUserName -Eq "David
Strome" }
```

If you want to change what properties are returned in the list or export the list to a .csv file, see Use the Shell to customize output and display it later in this topic.

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

Use the Shell to find a specific user on all roles

To know every role that a user receives permissions from, you must use the **Get-ManagementRoleAssignment** cmdlet to retrieve all effective users on all management roles and then pipe the output of the cmdlet to the **Where** cmdlet. The **Where** cmdlet filters the output and returns only the role assignments that grant the user permissions.

```
Get-ManagementRoleAssignment -GetEffectiveUsers | Where
{ $_.EffectiveUserName -Eq "<name of user>" }
```

This example finds all the role assignments that grant permissions to the user Kim Akers.

```
Get-ManagementRoleAssignment -GetEffectiveUsers | Where
{ $_.EffectiveUserName -Eq "Kim Akers" }
```

If you want to change what properties are returned in the list or export the list to a CSV file, see Use the Shell to customize output and display it later in this topic.

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

Use the Shell to customize output and display it

The default output of the **Get-ManagementRoleAssignment** cmdlet might not have the information you want. The output of the cmdlet contains many more properties that you can access. The following are some of the properties that could be useful:

- **EffectiveUserName** Name of the user.
- **Role** Role that's granting the permissions.
- **RoleAssigneeName** Role group, assignment policy, or USG that's assigned to the role and contains the user in the `EffectiveUserName` property.
- **RoleAssigneeType** Indicates whether the role assignment is to a role group, assignment policy, USG, or user.
- **AssignmentMethod** Indicates whether the assignment between the role and the role assignee is direct or indirect.
- **CustomRecipientWriteScope** Indicates the custom recipient write scope, if any, that was applied to the role assignment when it was created. The scope specified in this property overrides the implicit recipient write scope specified in the `RecipientWriteScope` property.
- **CustomConfigWriteScope** Indicates the custom configuration write scope, if any, that was applied to the role assignment when it was created. The scope specified in this property overrides the implicit configuration write scope specified in the `ConfigWriteScope` property.
- **RecipientReadScope** Indicates the implicit recipient read scope that's applied to the role.
- **RecipientWriteScope** Indicates the implicit recipient write scope that's applied to the role.
- **ConfigReadScope** Indicates the implicit configuration read scope that's applied to the role.
- **ConfigWriteScope** Indicates the implicit configuration write scope that's applied to the role.

To select the properties you want to display in your list, you use commands similar to those used in the Use the Shell to list all effective users, Use the Shell to find a specific user on a role, and Use the Shell to find a specific user on all roles sections. The difference is that you pipe the results of those commands to the **Format-Table** or **Select-Object** cmdlets. The **Format-Table** cmdlet is useful to output the list of results to your screen. The **Select-Object** cmdlet is useful to output the list of your results to a .csv file.

Both cmdlets let you specify the properties you want to see and in the order you want to see them. The **Format-Table** cmdlet gives you more options when you list results to a screen while **Select-Object** doesn't modify the output in any way, which is useful when piping the list to a .csv file.

For more information about the **Format-Table** and **Select-Object** cmdlets, see Working with command output.

Output a customized list to your screen

1. Choose the information you want to see and find the associated command from one of the following procedures:
   - Use the Shell to list all effective users
   - Use the Shell to find a specific user a role
   - Use the Shell to find a specific user on all roles
2. Choose the properties you want to see in your list.
3. Use the following syntax to view the list.

```
<command to retrieve list > | Format-Table <property 1>,
<property 2>, <property ...>
```

This example finds the user David Strome on all roles, and displays the `EffectiveUserName`, `Role`, `CustomRecipientWriteScope`, and `CustomConfigWriteScope` properties.

```
Get-ManagementRoleAssignment -GetEffectiveUsers | Where
{ $_.EffectiveUserName -Eq "David Strome" } | Format-Table
EffectiveUserName, Role, CustomRecipientWriteScope,
CustomConfigWriteScope
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

Output a customized list to a .csv file

To export a list to a .csv file, you need to pipe the results of the **Get-ManagementRoleAssignment** command from the appropriate procedure listed previously to the **Select-Object** cmdlet. The output of the **Select-Object** cmdlet is then piped to the **Export-CSV** cmdlet, which saves the .csv output to a file name you specify.

1. Choose the information you want to see and find the associated command from one of the following procedures:
   - Use the Shell to list all effective users
   - Use the Shell to find a specific user a role
   - Use the Shell to find a specific user on all roles
2. Choose the properties you want to see in your list.
3. Use the following syntax to export the list to a .csv file.

```
<command to retrieve list > | Select-Object <property 1>,
<property 2>, <property ...> | Export-CSV <filename>
```

This example finds the user David Strome on all roles, and displays the `EffectiveUserName`, `Role`, `CustomRecipientWriteScope`, and `CustomConfigWriteScope` properties.

```
Get-ManagementRoleAssignment -GetEffectiveUsers | Where
{ $_.EffectiveUserName -Eq "David Strome" } | Select-Object
EffectiveUserName, Role, CustomRecipientWriteScope,
CustomConfigWriteScope | Export-CSV c:\output.csv
```

You can now view the .csv file in a viewer of your choice.

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

Feature permissions

Exchange Server 2013 > Permissions >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-12*

Permissions in Microsoft Exchange Server 2013 are managed using the Role Based Access Control (RBAC) permissions model. The following topics identify the management role groups required to administer the features associated with each functional area in Exchange 2013.

- Role management permissions
- Messaging policy and compliance permissions
- Anti-spam and anti-malware permissions
- Mail flow permissions
- Recipients Permissions
- Email address and address book permissions
- Sharing and collaboration permissions
- Clients and mobile devices permissions
- Unified Messaging permissions
- High availability and site resilience permissions
- Exchange and Shell infrastructure permissions
- Server health and performance permissions

Role management permissions

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-07-05*

The permissions required to perform tasks to configure management roles vary depending on the procedure being performed or the cmdlet you want to run. For more information about management roles, see Understanding management roles.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or the cmdlet you want to run.
2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role groups and management roles, see Understanding Role Based Access Control.
3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:
You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Role management permissions

You can use the features in the following table to manage the management role groups, roles, assignment policies, assignments, scopes that define the permissions you can apply to administrators, and end users. Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Management roles	Organization Management
Unscoped management roles	Unscoped Role Management role management role

Role groups	Organization Management
Assignment policies	Organization Management
Role assignments	Organization Management
Management scopes	Organization Management
Management role entries	Organization Management
Legacy permissions	Organization Management
Active Directory split permissions	Organization Management **Important:** To run the `setup.exe` command with the *PrepareAD* and *ActiveDirectorySplitPermissions* parameters, the account you use must be a member of the Schema Admins and Enterprise Administrators groups.

Messaging policy and compliance permissions

Exchange Server 2013 > Permissions > Feature permissions >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-01-23*

The permissions required to configure messaging policy and compliance vary depending on the procedure being performed or the cmdlet you want to run. For more information about messaging policy and compliance, see Messaging policy and compliance.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or the cmdlet you want to run.
2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role

groups and management roles, see Understanding Role Based Access Control.

3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:

You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Note:

Some features that you want to manage might exist on Edge Transport servers. To manage features on Edge Transport servers, you need to become a member of the Local Administrators group on the Edge Transport server you want to manage. Edge Transport servers don't use Role Based Access Control (RBAC). Features that can be managed on Edge Transport servers have Edge Transport Local Administrator in the "Permissions required" column in the table below.

Messaging policy and compliance permissions

You can use the features in the following table to configure messaging policy and compliance features. The role groups that are required to configure each feature are listed.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Data loss prevention (DLP)	If using Office 365: • Office 365 global admin, which automatically includes Exchange Organization Management • Office 365 service admin, plus the Organization Management admin role group in Exchange • Office 365 password admin If using Exchange Server 2013 or Exchange Online only: • Compliance Management
Delete mailbox content (using the Search-Mailbox cmdlet with the *DeleteContent*	Mailbox Import Export role **Note:**

switch)	By default, the Mailbox Import Export role isn't assigned to any role group. You can assign a management role to a built-in or custom role group, a user or a universal security group. Assigning a role to a role group is recommended. For more information, see Add a role to a user or USG.
Discovery mailboxes - Create	Organization Management Recipient Management
Journaling	Organization Management Records Management
Mailbox audit logging	Organization Management Records Management
Message classifications	Organization Management
Messaging records management	Compliance Management Organization Management Records Management
In-Place eDiscovery	Discovery Management **Note:** By default, the Discovery Management role group doesn't have any members. No users, including administrators, have the required permissions to search mailboxes. For more information, see Add a user to the Discovery Management role group.
In-Place Hold	Discovery Management Organization Management **Important:** To create a query-based In-Place Hold, a user requires the Mailbox Search and Litigation Hold roles to be assigned directly or via membership in a role group that has both roles assigned. To create an In-Place Hold without using a query,

	which places all mailbox items on hold, you must have the Litigation Hold role assigned. The Discovery Management role group is assigned both roles. The Organization Management role group is assigned the Litigation Hold role. Members of the Organization Management role group can place an In-Place Hold on all items in a mailbox, but can't create a query-based In-Place Hold.
In-Place Archive	Organization Management Recipient Management
In-Place Archive – Test connectivity	Organization Management Server Management
Information Rights Management (IRM) configuration	Compliance Management Organization Management
Retention policies – Apply	Organization Management Recipient Management Records Management
Retention policies - Create	See the entry for Messaging Records Management
Transport rules	Organization Management Records Management

Anti-spam and anti-malware permissions

Exchange Server 2013 > Permissions > Feature permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-11*

The permissions required to perform tasks related to anti-spam and anti-malware vary depending on the procedure being performed or the cmdlet you want to run. For more information about transport features, see Mail flow.

This topic lists the permissions required to manage the mail flow features in Microsoft Exchange Server 2013.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or the cmdlet you want to run.
2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role groups and management roles, see Understanding Role Based Access Control.
3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:

You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Note:

Some features that you want to manage might exist on Edge Transport servers. To manage features on Edge Transport servers, you need to become a member of the Local Administrators group on the Edge Transport server you want to manage. Edge Transport servers don't use Role Based Access Control (RBAC). Features that can be managed on Edge Transport servers have Edge Transport Local Administrator in the "Permissions required" column in the table below.

Note:

Some features may require that you have local administrator permissions on the server you want to manage. To manage these features, you must be a member of the Local Administrators group on that server.

Anti-spam and Anti-Malware Permissions

You can use the features in the following tables to configure anti-spam and anti-malware settings in your organization. The permissions that are required to configure each feature are listed.

Users who are assigned the View Only Management role group can view the configuration of the features shown in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Anti-malware	Organization Management Hygiene Management
Anti-spam features	Organization Management Hygiene Management
Anti-spam features - Edge Transport	Edge Transport Local Administrator

Mail flow permissions

Exchange Server 2013 > Permissions > Feature permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-07*

The permissions required to perform tasks related to mail flow vary depending on the procedure being performed or the cmdlet you want to run. For more information about transport features, see Mail flow.

This topic lists the permissions required to manage the mail flow features in Microsoft Exchange Server 2013. For information about how Office 365 permissions relate to Exchange permissions, see Permissions in Office 365.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or the cmdlet you want to run.
2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role groups and management roles, see Understanding Role Based Access Control.
3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:

You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Note:

Some features that you want to manage might exist on Edge Transport servers. To manage features on Edge Transport servers, you need to become a member of the Local Administrators group on the Edge Transport server you want to manage. Edge Transport servers don't use Role Based Access Control (RBAC). Features that can be managed on Edge Transport servers have Edge Transport Local Administrator in the "Permissions required" column in the table below.

Note:

Some features may require that you have local administrator permissions on the server you want to manage. To manage these features, you must be a member of the Local Administrators group on that server.

Mail flow permissions

You can use the features in the following tables to configure mail flow settings in the Front End Transport service on Client Access servers, in the Transport service on Mailbox servers, in the Mailbox Transport service on Mailbox servers, and on Edge Transport servers. The permissions that are required to configure each feature are listed.

Users who are assigned the View Only Management role group can view the configuration of the features shown in the following table. For more information, see View-only Organization Management.

Mailbox servers and Client Access servers

Feature	Permissions required
Accepted domains	Organization Management
Active Directory site and site link management	Organization Management
Anti-spam features	Organization Management Hygiene Management
Anti-spam updates	Organization Management Hygiene Management

Certificate management	Organization Management
Delivery Agent connectors	Organization Management Server Management
DSNs	Organization Management
EdgeSync	Organization Management
Foreign connectors	Organization Management
Front End Transport service	Organization Management Server Management Hygiene Management
Journaling	Organization Management Records Management
Mailbox access	Organization Management
Mailbox junk email configuration	Organization Management Records Management Recipient Management Help Desk
Mailbox Transport service	Organization Management Server Management Hygiene Management
MailTips	Organization Management
Message classifications	Organization Management Records Management
Message tracking	Organization Management Records Management

	Recipient Management
Moderated transport	Organization Management Recipient Management
Queues	Organization Management Server Management
Receive connectors	Organization Management Server Management Hygiene Management
Remote domains	Organization Management
SafeList aggregation	Organization Management Records Management
Send connectors	Organization Management
Shadow redundancy	Organization Management
Testing mail flow	Organization Management Server Management
Testing Transport rule processing	Organization Management
Transport agents	Organization Management Records Management
Transport configuration	Organization Management
Transport logs	Organization Management Server Management
Transport rules	Organization Management Records Management

Transport service	Organization Management Server Management Hygiene Management
X.400 domains	Organization Management

Edge Transport servers

Feature	Permissions required
Accepted domains - Edge Transport	Edge Transport Local Administrator
Address Rewriting - Edge Transport	Edge Transport Local Administrator
Edge Transport server	Edge Transport Local Administrator
EdgeSync - Edge Transport	Edge Transport Local Administrator
Queues - Edge Transport	Edge Transport Local Administrator
Receive connectors - Edge Transport	Edge Transport Local Administrator
Send connectors - Edge Transport	Edge Transport Local Administrator
Transport configuration - Edge Transport	Edge Transport Local Administrator
Transport logs - Edge Transport	Edge Transport Local Administrator
Transport rules - Edge Transport	Edge Transport Local Administrator

Recipients Permissions

Exchange Server 2013 > Permissions > Feature permissions >

Topic Last Modified: *2013-02-21*

The permissions required to perform tasks to manage recipients vary depending on the procedure being performed or the cmdlet you want to run.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or

the cmdlet you want to run.

2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role groups and management roles, see Understanding Role Based Access Control.
3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:
You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Mailbox server permissions

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Calendar repair, server configuration	Organization Management Server Management
Delegating Mailbox servers	Organization Management
Email address policies	Organization Management Server Management
Exchange Search	Organization Management View-only Organization Management Server Management
Exchange Search – diagnostics	Organization Management View-only Organization Management Support Diagnostics role

	Note: The Support Diagnostics role isn't assigned to a role group. For more information, see Add a role to a user or USG.
Group metrics	Organization Management Server Management
Import Export	Mailbox Import Export role **Note:** The Mailbox Import Export role isn't assigned to a role group. For more information, see Mailbox Import Export role.
Mailbox Assistants	Organization Management Server Management
Mailbox moves	Organization Management Recipient Management
Mailbox recovery	Organization Management
Mailbox repair request	Organization Management Server Management Recipient Management
Mailbox restore request	Organization Management
Mailbox server configuration	Organization Management Server Management
Manage Exchange Search Indexer service on a Mailbox server	Local Administrator on the Mailbox server
MAPI connectivity	Organization Management Server Management
OAB virtual directories	Organization Management

	Server Management
Remove store mailbox	Organization Management Server Management

Calendar and sharing permissions

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Calendar configuration	Organization Management Recipient Management Help Desk
Calendar diagnostics	Organization Management Records Management Hygiene Management Compliance Management Help Desk
Calendar processing	Organization Management Recipient Management Help Desk
Notifications	Organization Management Recipient Management
Organization relationships	Organization Management
Sharing policies	Organization Management

Resource mailbox configuration permissions

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Booking policies	Organization Management Recipient Management Help Desk
Delegation	Organization Management Recipient Management
Resource mailbox schema configuration	Organization Management

Mailbox database permissions

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Mailbox databases	Organization Management Server Management

Recipient provisioning permissions

This table contains the various permissions that are required to manage recipients.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Address list, GAL	Organization Management
Anti-spam	Organization Management Recipient Management
Apps for Outlook	Organization Management View-only Organization Management Help Desk
Applying sharing policies	Organization Management Recipient Management

Arbitration	Organization Management
Archive connectivity	Organization Management View-only Organization Management Server Management
Assigning offline address books	Organization Management Recipient Management
Automatic replies	Organization Management Recipient Management Help Desk
Calendar configuration	Organization Management Recipient Management
Calendar repair	Organization Management Recipient Management
Contact aggregation settings	Organization Management Recipient Management View-only Organization Management
Disconnected mailboxes	Organization Management Recipient Management Help Desk
Distribution groups	Organization Management Recipient Management
Dynamic distribution groups	Organization Management Recipient Management
Email addresses	Organization Management

	Recipient Management UM Management
Inbox rules	Organization Management Recipient Management Help Desk
Mail contacts	Organization Management Recipient Management
Mail tips	Organization Management Recipient Management
Mail user	Organization Management Recipient Management
Mailbox folder permissions	Organization Management Recipient Management Help Desk
Mailbox folders	Organization Management Recipient Management
MAPI connectivity	Organization Management
Message configuration	Organization Management Recipient Management Help Desk
Message quotas	Organization Management Recipient Management
Moderation	Organization Management Recipient Management

Permissions and delegation	Organization Management	
Archive mailboxes	Organization Management Recipient Management	
Recipient data properties	Organization Management Recipient Management	
Remote mailboxes	Organization Management Recipient Management	
Retention and legal holds	Organization Management Recipient Management Records Management	
Send As	Organization Management Recipient Management	
Spelling configuration	Organization Management Recipient Management Help Desk	
Unified Messaging	Organization Management UM Management	
User mailboxes	Organization Management Recipient Management	
User photos	Organization Management Recipient Management Help Desk	

Mailbox move and migration permissions

The table contains the permissions that are required to move on-premises mailboxes to different

domains or forests and to migrate on-premises mailboxes to and from your cloud-based organization.

Feature	Permissions required
Mailbox moves (local or cross-forest)	Organization Management Recipient Management
Mailbox moves (hybrid deployment)	Organization Management Recipient Management
Migration (on-boarding and off-boarding from the cloud)	Organization Management Recipient Management

Email address and address book permissions

Exchange Server 2013 > Permissions > Feature permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-11*

The permissions required to configure email address and address book features vary depending on the procedure being performed or the cmdlet you want to run. For more information about email addresses and address books, see Email addresses and address books.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or the cmdlet you want to run.
2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role groups and management roles, see Understanding Role Based Access Control.
3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:

You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Email address and address book permissions

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Address book policies	Organization Management
Address lists	Organization Management
Email address policies	Organization Management
Details templates	Organization Management
Global address lists	Organization Management
Offline address books	Organization Management
Offline address book connectivity	Organization Management

Sharing and collaboration permissions

Exchange Server 2013 > Permissions > Feature permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-14*

The permissions required to configure sharing and collaboration features vary depending on the procedure being performed or the cmdlet you want to run. For more information about sharing and collaboration, see Collaboration and Sharing.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or

the cmdlet you want to run.

2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role groups and management roles, see Understanding Role Based Access Control.
3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:
You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Sharing and collaboration feature permissions

You can use the features in the following table to configure sharing and collaboration features. The role groups that are required to configure each feature are listed.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Partner applications - configure	Organization Management
Public folders, mail-enabled	Organization Management Recipient Management
Public folders	Organization Management
Site mailboxes	Organization Management Recipient Management
Site mailbox provisioning policy	Organization Management Recipient Management

Clients and mobile devices permissions

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-31*

The permissions required to perform tasks for clients and mobile devices vary depending on the procedure being performed or the cmdlet you want to run. For more information about client and mobile device features, see Clients and mobile.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or the cmdlet you want to run.
2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role groups and management roles, see Understanding Role Based Access Control.
3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:
You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Note:
Some features may require that you have local administrator permissions on the server you want to manage. To manage these features, you must be a member of the Local Administrators group on that server.

Client Access server permissions

You can configure any of the following features for the Client Access server.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Client Access server array settings	Organization Management Server Management
Client Access server settings	Server Management
Client Access service email channel settings	Organization Management Server Management
Client Access user settings	Server Management
Client Access virtual directory settings	Organization Management Server Management
RPC Client Access settings	Organization Management Server Management View-only Organization Management
Push notification proxy settings	Organization Management Recipient Management
OAuth authentication redirection settings	Organization Management

Exchange ActiveSync permissions

You can configure any of the following for Exchange ActiveSync.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Exchange ActiveSync Autoblock settings	Organization Management
Exchange ActiveSync mailbox policy settings	Organization Management Server Management
Exchange ActiveSync server settings	Organization Management Server Management

Exchange ActiveSync settings	Organization Management Server Management
Exchange ActiveSync user settings	Recipient Management
Exchange ActiveSync virtual directory settings	Organization Management Server Management
Mobile device mailbox policy settings	Organization Management Server Management
Mobile device user settings	Organization Management Server Management Recipient Management

Autodiscover permissions

You can configure the following for the Autodiscover service.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Autodiscover service configuration settings	Organization Management Server Management View-only Organization Management Delegated Setup Hygiene Management
Autodiscover virtual directory settings	Organization Management Server Management

Availability service permissions

You can configure the following for the Availability service.

Users who are assigned the View-Only Management role group can view the configuration of the

features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Availability service address space settings	Organization Management View-only Organization Management
Availability service configuration settings	Organization Management Server Management View-only Organization Management

Client throttling permissions

You can configure the following for client throttling.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Client throttling settings	Organization Management View-only Organization Management

Exchange Web Services permissions

You can configure the following for Web Services virtual directories.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Exchange Web Services virtual directory settings	Organization Management Server Management
Test Exchange Web Services	Organization Management Server Management
Test Outlook Web Services	Organization Management

Outlook Anywhere permissions

You can configure and manage the following settings for Outlook Anywhere.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Outlook Anywhere configuration (enable, disable, change, view)	Organization Management Server Management View-only Organization Management Delegated Setup Hygiene Management
RPC over HTTP Proxy component	Local Server Administrator
Test Outlook Anywhere connectivity	Organization Management View-only Organization Management Server Management

Outlook Web App permissions

You can use the following features to view Outlook Web App settings, control security and user access to Outlook Web App, and test Outlook Web App connectivity.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Graphics editor	Local Server Administrator
IIS Manager	Local Server Administrator
ISA Server 2006	ISA Server Enterprise Administrator
Outlook Web App mailbox policies	Organization Management Recipient Management
Outlook Web App virtual directories	Organization Management Server Management
Registry Editor	Local Server Administrator

S/MIME configuration	Organization Management
Text editor	Local Server Administrator
View Outlook Web App mailbox policies	Organization Management Recipient Management View-only Organization Management Delegated Setup Hygiene Management

POP3 and IMAP4 permissions

You can configure the following for POP3 and IMAP4.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
IMAP4 settings	Organization Management Server Management View-only Organization Management
POP3 settings	Organization Management Server Management View-only Organization Management
Test IMAP4 settings	Organization Management Server Management View-only Organization Management
Test POP3 settings	Organization Management Server Management View-only Organization Management

Windows PowerShell virtual directory permissions

You can configure the following for Windows PowerShell.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Test Windows PowerShell	Organization Management
Windows PowerShell settings	Organization Management

Text Messaging permissions

You can configure the following for text messaging.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Text messaging notification settings	Recipient Management
Text messaging settings	Recipient Management
Text messaging user settings	Recipient Management

Unified Messaging permissions

Exchange Server 2013 > Permissions > Feature permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

The permissions required to perform tasks on Client Access servers that are running the Microsoft Exchange Unified Messaging Call Router service and Mailbox servers that are running the Microsoft Exchange Unified Messaging service vary depending on the procedure being performed or the cmdlet you want to run.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or the cmdlet you want to run.

2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role groups and management roles, see Understanding Role Based Access Control.
3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:
You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

UM component permissions

You can configure settings for the UM components and features in the following table.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
UM auto attendants	Organization Management UM Management
UM call answering rules	Organization Management UM Management
UM call data and summary reports	Organization Management UM Management
Client Access Server (UM call router service)	Organization Management UM Management
UM dial plans	Organization Management UM Management
UM hunt groups	Organization Management

	UM Management
UM IP gateways	Organization Management UM Management
UM mailbox policies	Organization Management UM Management
UM mailboxes	Organization Management UM Management
UM prompts	Organization Management UM Management
Mailbox server (UM service)	Organization Management Server Management

High availability and site resilience permissions

Exchange Server 2013 > Permissions > Feature permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-02*

The permissions required to configure high availability vary depending on the procedure being performed or the cmdlet you want to run. For more information about high availability, see High availability and site resilience.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or the cmdlet you want to run.
2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role

groups and management roles, see Understanding Role Based Access Control.

3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:
You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Database availability group permissions

You can use the features in the following table to add, remove, and configure settings for database availability groups (DAGs).

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Database availability group membership	Organization Management Database Availability Groups role
Database availability group properties	Organization Management Database Availability Groups role
Database availability groups	Organization Management Database Availability Groups role
Database availability networks	Organization Management Database Availability Groups role

Mailbox database copy permissions

You can use the features in the following table to add, remove, update, and activate mailbox database copies.

Feature	Permissions required
Database switchover	Organization Management

	Database Copies role
Mailbox database copies	Organization Management Database Copies role
Server switchover	Organization Management Database Copies role
Update a mailbox database copy	Organization Management Database Copies role

Exchange and Shell infrastructure permissions

Exchange Server 2013 > Permissions > Feature permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-24*

The permissions required to perform tasks to configure various components of Microsoft Exchange Server 2013 depend on the procedure being performed or the cmdlet you want to run. See each of the sections in this topic for more information about their respective features.

To find out what permissions you need to perform the procedure or run the cmdlet, do the following:

1. In the table below, find the feature that is most related to the procedure you want to perform or the cmdlet you want to run.
2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role groups and management roles, see Understanding Role Based Access Control.
3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:
You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-**

ManagementRoleAssignment cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Note:

Some features may require that you have local administrator permissions on the server you want to manage. To manage these features, you must be a member of the Local Administrators group on that server.

Exchange infrastructure permissions

The following table lists the permissions required to perform tasks that configure general Exchange 2013 settings.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Administrator audit logging	Organization Management Records Management
Exchange Administration Center configuration settings	View-only Organization Management
Exchange Administration Center connectivity	Organization Management Server Management
Exchange server configuration settings	Organization Management Server Management
Exchange Help settings	Organization Management
Message categories	Organization Management Hygiene Management Recipient Management Help Desk
Product key	Organization Management

Test system health	Organization Management Server Management	
View-only administrator audit logging	Organization Management Records Management **Note:** You can also manually assign the View-Only Audit Logs management role to a management role group. For more information, see View-Only Audit Logs role.	
Write to audit log	Users that are members of any role group or assigned any management role can write to the administrator audit log.	

Shell infrastructure permissions

The following table lists the permissions required to perform tasks that configure features that control how the Exchange Management Shell runs.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Active Directory Domain Services server settings	Organization Management Server Management Recipient Management UM Management
Cmdlet extension agents	Organization Management
PowerShell virtual directories	Organization Management Server Management
PowerShell and WinRM installation	Local Server Administrator
Remote Shell	Organization Management

Federation and certificates permissions

The following table lists permissions required for performing tasks related to federation trusts, OAuth configuration, certificate management, and hybrid deployment configuration.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Certificate management	Organization Management Server Management
Federation trusts, OAuth	Organization Management
Test federation trusts, OAuth	Organization Management View-only Organization Management Server Management
Hybrid deployment configuration	Organization Management
Intra-Organization connectors	Organization Management Recipient Management Records Management

Server health and performance permissions

Exchange Server 2013 > Permissions > Feature permissions >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-07-19*

The permissions required to perform tasks to configure various components of Microsoft Exchange Server 2013 depend on the procedure being performed or the cmdlet you want to run. See each of the sections in this topic for more information about their respective features.

To find out what permissions you need to perform the procedure or run the cmdlet, do the

following:

1. In the table below, find the feature that is most related to the procedure you want to perform or the cmdlet you want to run.
2. Next, look at the permissions required for the feature. You must be assigned one of those role groups, an equivalent custom role group, or an equivalent management role. You can also click on a role group to see its management roles. If a feature lists more than one role group, you only need to be assigned one of the role groups to use the feature. For more information about role groups and management roles, see Understanding Role Based Access Control.
3. Now, run the **Get-ManagementRoleAssignment** cmdlet to look at the role groups or management roles assigned to you to see if you have the permissions that are necessary to manage the feature.

Note:
You must be assigned the Role Management management role to run the **Get-ManagementRoleAssignment** cmdlet. If you don't have permissions to run the **Get-ManagementRoleAssignment** cmdlet, ask your Exchange administrator to retrieve the role groups or management roles assigned to you.

If you want to delegate the ability to manage a feature to another user, see Delegate role assignments.

Note:
Some features may require that you have local administrator permissions on the server you want to manage. To manage these features, you must be a member of the Local Administrators group on that server.

Exchange workload management permissions

The following table lists the permissions required to perform tasks that manage the health and performance of your Exchange 2013 organization. For more information, see the following topics:

- Exchange workload management
- Workload management reference

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required	
User throttling	Organization Management Recipient Management View-only Organization Management	
Exchange workload throttling	Organization Management View-only Organization	

	Management	

Exchange event log permissions

The following table lists the permissions required to perform tasks that manage Exchange event log settings.

Users who are assigned the View-Only Management role group can view the configuration of the features in the following table. For more information, see View-only Organization Management.

Feature	Permissions required
Exchange event log management	Organization Management Server Management View-only Organization Management UM Management

Advanced permissions

Exchange Server 2013 > Permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-13*

The procedures in the following sections enable you to configure advanced permissions models for your organization. You should have an in-depth knowledge of Role Based Access Control (RBAC) before performing advanced customization of your permissions model.

Management roles and role entries

Management role scopes

Management role assignments

Managing split permissions

We recommend you manage your permissions using management role groups and management role assignment policies. For more information, see the following topics:

Manage role groups

Manage linked role groups

Manage role assignment policies

For more information about RBAC, see Understanding Role Based Access Control.

Management roles and role entries

Exchange Server 2013 > Permissions > Advanced permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-13*

The following procedures enable you to perform advanced permissions management. You should only use these procedures if management role groups and management role assignment policies don't meet the needs of your organization.

Create a role

View a role

Remove a role

Add a role entry to a role

Change a role entry

View role entries

Remove a role entry from a role

Create an unscoped role

Change a role entry on an unscoped top-level role

Add a role entry to an unscoped top-level role

For more information about managing role groups and role assignment policies, see the following topics:

Manage role groups

Manage linked role groups

Manage role assignment policies

Create a role

Permissions > Advanced permissions > Management roles and role entries >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

You can create a management role, change the management role entries, add a scope if needed, and then assign the role to a role assignee. You should rarely need to perform this procedure. We recommend that you check whether a built-in management role can be used instead of creating a management role. For a list of built-in management roles, see Built-in management roles.

For more information about management roles in Microsoft Exchange Server 2013, see Understanding management roles.

Note:

This topic doesn't discuss how to create an unscoped management role. For information about how to create an unscoped management role, see Create an unscoped role.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete this procedure: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Create the management role

New management roles are based on existing roles. When you create a role, an existing role and its management role entries are copied to the new role. The existing role becomes the parent to the new child role. You must always choose a role that contains all the cmdlets and parameters you need to use, and then remove the ones you don't want. Child roles can't have management role entries that don’t exist in the parent role.

Use the following syntax to create the new role.

```
New-ManagementRole -Parent <existing role to copy> -Name
<name of new role>
```

This example copies the Mail Recipients role and its management role entries to the Seattle Mail Recipients role.

```
New-ManagementRole -Parent "Mail Recipients" -Name "Seattle
Mail Recipients"
```

For detailed syntax and parameter information, see New-ManagementRole.

Step 2: Change the new role's management role entries

After you create your role, you need to change the role's entries. You can remove an entire role entry, which removes access to the associated cmdlet completely. Or, you can remove parameters from a role entry to remove access to those specific parameters on the associated cmdlet.

You can't add new role entries or parameters on role entries unless they exist in the parent role. Because you just created a role from a parent role in Step 1, you can't add any additional role entries or parameters on role entries because they don't exist in the parent role.

When you change a role entry on a role, you can do one of the following:

- Remove a single, entire role entry.
- Remove multiple, entire role entries.
- Remove parameters from a role entry.

To remove role entries from your new role, see Remove a role entry from a role.

Step 3: Create a custom management role scope, if required

Management role scopes determine the objects made available to a user to view or change using the role entries configured in Step 2. New management roles inherit the read and write management role scopes of their parent role. These are called *implicit scopes*. However, there may be cases where you want to change the write scope of the new role to match your business needs. When you create a custom scope, you override the implicit write scope of the role. The implicit read scope of the role doesn't change. For more information about management role scopes, see Understanding management role scopes.

You can create a custom scope, create an exclusive scope, use a predefined scope, or scope an assignment to an organizational unit (OU). The new scope must be within the implicit read scope of the role. To use a predefined scope or to specify an organizational unit, skip to Step 4.

To add a custom scope to your new role, see Create a regular or exclusive scope.

Step 4: Assign the new management role

The final step when you create and configure a role is to assign it to a role assignee.

When you create a role assignment, you can choose to do one of the following:

- Create the role assignment with no scope.
- Create the role assignment with a predefined scope.
- Create the role assignment with an OU without a domain restriction filter.
- Create the role assignment with the custom or exclusive scope you created in Step 3.

Note:
You can't specify a scope when you create an assignment between a role and a management role assignment policy.

You can assign the new role to a role group, a role assignment policy, a user, or a universal security group (USG). For more information, see the following topics:

- Manage role groups
- Manage role assignment policies
- Add a role to a user or USG

View a role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Management roles can be listed in a variety of ways, depending on the information you want. For example, you can choose to return only roles of a specific role type, roles that contain only specific cmdlets and parameters, or view the details of a specific management role. For more information about management roles in Microsoft Exchange Server 2013, see Understanding management roles.

If you want to view a list of all management role entries on a role, see View role entries.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- This topic makes use of pipelining and the **Format-List** and **Format-Table** cmdlets. For more information about these concepts, see the following topics:
  - Pipelining
  - Working with command output
- For information about keyboard shortcuts that may apply to the procedures in this topic, see

Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

View a specific management role

You can view the details of a specific role by retrieving a specific role using the **Get-ManagementRole** cmdlet and piping the output to the **Format-List** cmdlet.

To view the details of a specific role, use the following syntax.

```
Get-ManagementRole <role name> | Format-List
```

This example retrieves the details about the Mail Recipients management role.

```
Get-ManagementRole "Mail Recipients" | Format-List
```

For detailed syntax and parameter information, see Get-ManagementRole.

List all management roles

You can view a list of all the management roles in your organization by not specifying any roles when you run the **Get-ManagementRole** cmdlet. By default, the role name and role type of each role are included in the results.

This example returns a list of all roles in your organization.

```
Get-ManagementRole
```

To return a list of specific properties for all the roles in your organization, you can pipe the results of the **Format-Table** cmdlet and specify the properties you want in the list of results. Use the following syntax.

```
Get-ManagementRole | Format-Table <property 1>, <property 2...>
```

This example returns a list of all the roles in your organization and includes the **Name** property and any property with the word **Implicit** at the beginning of the property name.

```
Get-ManagementRole | Format-Table Name, Implicit*
```

For detailed syntax and parameter information, see Get-ManagementRole.

List management roles that contain a specific cmdlet

You can return a list of roles that contain a cmdlet that you specify by using the *Cmdlet* parameter on the **Get-ManagementRole** cmdlet.

To return a list of roles that contain the cmdlet you specify, use the following syntax.

```
Get-ManagementRole -Cmdlet <cmdlet>
```

This example returns a list of roles that contain the **New-Mailbox** cmdlet.

```
Get-ManagementRole -Cmdlet New-Mailbox
```

For detailed syntax and parameter information, see Get-ManagementRole.

List management roles that contain a specific parameter

You can return a list of roles that contain one or more specified parameters by using the *CmdletParameters* parameter on the **Get-ManagementRole** cmdlet. Only roles that contain all the parameters you specify are returned.

When you use the *CmdletParameters* parameter, you can choose to include the *Cmdlet* parameter. If you include the *Cmdlet* parameter, only roles that contain the parameters you specify on the cmdlet you specify are returned. If you don't include the *Cmdlet* parameter, roles that contain the parameters you specify, regardless of the cmdlet they're on, are returned.

To return a list of roles that contain the parameters you specify, use the following syntax.

```
Get-ManagementRole [-Cmdlet <cmdlet>] -CmdletParameters
<parameter 1>, <parameter 2...>
```

This example returns a list of roles that contain the *Database* and *Server* parameters, regardless of the cmdlets they exist on.

```
Get-ManagementRole -CmdletParameters Database, Server
```

This example returns a list of roles where the *EmailAddresses* parameter exists only on the **Set-Mailbox** cmdlet.

```
Get-ManagementRole -Cmdlet Set-Mailbox -CmdletParameters
EmailAddresses
```

You can also use the wildcard character (*) with either the *Cmdlet* or *CmdletParameters* parameters to match partial cmdlet or parameter names.

For detailed syntax and parameter information, see Get-ManagementRole.

List management roles of a specific role type

You can return a list of roles based on a specified role type by using the *RoleType* parameter on the **Get-ManagementRole** cmdlet.

To return a list of roles that match the role type you specify, use the following syntax.

```
Get-ManagementRole -RoleType <roletype>
```

This example returns a list of roles based on the `UmMailboxes` role type.

```
Get-ManagementRole -RoleType UmMailboxes
```

For detailed syntax and parameter information, see Get-ManagementRole.

List the immediate child roles of a parent role

You can return a list of roles that are the immediate children of the specified parent role by using the *GetChildren* parameter on the **Get-ManagementRole** cmdlet. Only roles that contain the role you specify as the parent role are returned.

To return a list of the immediate children roles of a parent role, use the following syntax.

```
Get-ManagementRole <parent role name> -GetChildren
```

This example returns a list of immediate children of the Disaster Recovery role.

```
Get-ManagementRole "Disaster Recovery" -GetChildren
```

For detailed syntax and parameter information, see Get-ManagementRole.

List all child roles below a parent role

You can return a list of the entire chain of roles from a specified parent role to the last child role by using the *Recurse* parameter on the **Get-ManagementRole** cmdlet. The *Recurse* parameter tells the **Get-ManagementRole** cmdlet to recurse down through every parent and child relationship it finds until it reaches the last child role. The parent role is included in the list that's returned.

This example returns a list of all the child roles of a parent role.

```
Get-ManagementRole <parent role name> -Recurse
```

This example returns all the child roles of the Mail Recipients role.

```
Get-ManagementRole "Mail Recipients" -Recurse
```

For detailed syntax and parameter information, see Get-ManagementRole.

Remove a role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Management roles that are no longer required can be removed from your organization. You can only remove management roles that you created. Built-in management roles can't be removed. For more information about management roles in Microsoft Exchange Server 2013, see Understanding management roles.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- Before you can remove a management role, you must remove all its management role assignments. For more information about how to remove a role assignment, see Remove a role from a user or USG.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Remove a management role with no child roles

To remove a role with no child roles, use the following syntax.

```
Remove-ManagementRole <role name>
```

This example removes the Seattle Server Administrators role.

```
Remove-ManagementRole "Seattle Server Administrators"
```

For detailed syntax and parameter information, see Remove-ManagementRole.

Remove a management role with child roles

If a role that you want to remove has child roles, you must remove all the child roles also. You receive an error message if you try to remove a role that has child roles unless you use the *Recurse* switch. If you use the *Recurse* switch when you remove a role, the role you specify and all its child roles are removed.

Caution:

If you use the *Recurse* switch, all child roles of the specified role you want to remove are also removed. Make sure that you're aware of what roles will be removed before you run this command.

To make sure that you remove only the roles that you want to remove, use the *WhatIf* switch with your command to verify that it's correct. Use the following syntax.

```
Remove-ManagementRole <role name> -Recurse -WhatIf
```

The *WhatIf* switch performs the command without committing any changes and reports which roles it would have removed. For more information about the *WhatIf* switch, see WhatIf, Confirm, and ValidateOnly switches.

After you confirm that only the roles you want to remove will be removed, run the same command without the *WhatIf* switch. This example removes the London Administrators role and all its child roles.

```
Remove-ManagementRole "London Administrators" -Recurse
```

For detailed syntax and parameter information, see Remove-ManagementRole.

Remove an unscoped management role

To remove an unscoped role, the same procedures provided in Remove a management role with no child roles and Remove a management role with child roles earlier in this topic can be used. The only difference is that when you remove an unscoped role, you must specify the *UnScopedTopLevel* switch when you run the command. This example removes an unscoped role and all its child roles.

```
Remove-ManagementRole "Custom IT Scripts" -Recurse -
UnScopedTopLevel
```

As with removing other roles, you should use the *WhatIf* switch to verify that you're removing the correct roles.

For detailed syntax and parameter information, see Remove-ManagementRole.

Add a role entry to a role

Permissions > Advanced permissions > Management roles and role entries >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-04*

If you want to grant access to a cmdlet, you need to add the associated management role entry to a management role. After you add the role entry to a role, the users assigned the role will be able to access that cmdlet. For more information about management role entries in Microsoft Exchange Server 2013, see Understanding management roles.

You can't add role entries to built-in roles. If you want to customize roles, you must create a new role. For more information about how to create a new role, see Create a role.

Note:

This topic doesn't discuss how to add unscoped management role entries to an unscoped management role. For more information about how to add unscoped role entries, see Add a role entry to an unscoped top-level role.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- A role entry that you want to add to a management role must exist in that role's immediate parent management role.
- This topic makes use of pipelining. For more information about pipelining, see Pipelining.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Add a single role entry from a parent role

You can add a role entry to a role exactly as it appears on the parent role by using the following syntax.

```
Add-ManagementRoleEntry <child role name>\<cmdlet>
```

This example adds the **Set-Mailbox** cmdlet to the Recipient Administrators role.

```
Add-ManagementRoleEntry "Recipient Administrators\Set-Mailbox"
```

This command checks the parent role, and if the role entry exists, adds it to the child role. If the role entry already exists on the child role, you can include the *Overwrite* parameter to overwrite the existing role entry.

For detailed syntax and parameter information, see Add-ManagementRoleEntry.

Add a single role entry from a parent role and include only specific parameters

If you want to add a role entry from a parent role, but you want to include only specific parameters in the role entry on the child role, use the following syntax.

```
Add-ManagementRoleEntry <child role name>\<cmdlet> -Parameters <parameter 1>, <parameter 2>, <parameter...>
```

This example adds the **Set-Mailbox** cmdlet to the Help Desk role, but includes only the *DisplayName* and *EmailAddresses* parameters in the entry on the child role.

```
Add-ManagementRoleEntry "Help Desk\Set-Mailbox" -Parameters DisplayName, EmailAddresses
```

This command checks the parent role, and if the role entry exists, adds it to the child role. If the role entry already exists on the child role, you can include the *Overwrite* parameter to overwrite the existing role entry.

For detailed syntax and parameter information, see Add-ManagementRoleEntry.

Add multiple role entries from a parent role

If you want to add more than one role entry to a role, you need to retrieve a list of role entries that exist on the parent role that you want to add to the child role, and then add them to the child role.

To do this, you retrieve the list of role entries on a parent role by using the **Get-ManagementRoleEntry** cmdlet. Then you pipe the output of the **Get-ManagementRoleEntry** cmdlet to the **Add-ManagementRoleEntry** cmdlet. To retrieve multiple role entries, you need to use the wildcard character (*).

To add multiple entries from a parent role to a child role, use the following syntax.

```
Get-ManagementRoleEntry <parent role name>\*<partial cmdlet name>* | Add-ManagementRoleEntry -Role <child role name>
```

This example adds all the role entries that contain the string `Mailbox` in the cmdlet name on the Mail Recipients parent role to the Seattle Mail Recipients child role.

```
Get-ManagementRoleEntry "Mail Recipients\*Mailbox*" | Add-ManagementRoleEntry -Role "Seattle Mail Recipients"
```

If the role entries already exist on the child role, you can include the *Overwrite* parameter to overwrite the existing role entries.

For more information about retrieving a list of management role entries, see View role entries.

For detailed syntax and parameter information, see Get-ManagementRoleEntry and Add-ManagementRoleEntry.

Change a role entry

Permissions > Advanced permissions > Management roles and role entries >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Each management role entry on a management role represents a single cmdlet. By adding parameters to or removing parameters from a role entry, which is then added to a management role, you control whether those parameters are available on that cmdlet. For more information about management role entries in Microsoft Exchange Server 2013, see Understanding management roles.

You can't modify the role entries on built-in management roles.

Note:

This topic doesn't discuss how to modify unscoped management role entries on an unscoped management role. For more information about how to modify unscoped role entries, see Create a role.

Caution:

To add or remove parameters from a role entry, you must use the *AddParameter* or *RemoveParameter* parameters. If you omit the *AddParameter* or *RemoveParameter* parameter when you run the **Set-ManagementRoleEntry** cmdlet, only the parameters you specify using the *Parameters* parameter will be included in the role entry. All other parameters on the role entry will be removed.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- If you want to add parameters to a role entry, the parameters you add must exist in the role entry in the parent role. The parameters must also exist on the cmdlet you specify.
- If you want to remove parameters from a role entry, the parameters you remove can't exist in the role entries of any child roles. You must remove the parameters from the role entries of the child roles. Use the "Use the Shell to remove one or more parameters from a role entry" procedure later in this topic to remove the parameters from the role entries of all child roles.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to add one or more parameters to a role entry

To add parameters to a role entry, you need to specify the parameters you want to add using the *Parameters* parameter. You then need to specify the *AddParameter* parameter to indicate that you want to perform an add operation.

To add parameters to a role entry, use the following syntax.

```
Set-ManagementRoleEntry <role name>\<cmdlet> -Parameters
<parameter 1>, <parameter 2>, <parameter...> -AddParameter
```

This example adds the *EmailAddresses* and *Type* parameters to the **Set-Mailbox** cmdlet on the

Recipient Administrators role.

```
Set-ManagementRoleEntry "Recipient Administrators\Set-
Mailbox" -Parameters EmailAddresses, Type -AddParameter
```

For detailed syntax and parameter information, see Set-ManagementRoleEntry.

Use the Shell to remove one or more parameters from a role entry

To remove parameters from a role entry, you need to specify the parameters you want to remove using the *Parameters* parameter. You then need to specify the *RemoveParameter* parameter to indicate that you want to perform a remove operation.

To remove parameters from a role entry, use the following syntax.

```
Set-ManagementRoleEntry <role name>\<cmdlet> -Parameters
<parameter 1>, <parameter 2>, <parameter...> -
RemoveParameter
```

This example removes the *Port*, *ProtocolLoggingLevel*, and *SmartHostAuthMechanism* parameters from the **Set-SendConnector** cmdlet on the Tier 1 Server Administrators role.

```
Set-ManagementRoleEntry "Tier 1 Server Administrators\Set-
SendConnector" -Parameters Port, ProtocolLoggingLevel,
SmartHostAuthMechanism -RemoveParameter
```

For detailed syntax and parameter information, see Set-ManagementRoleEntry.

Use the Shell to remove all parameters from a role entry

To remove all the parameters from a role entry, you need to specify the value `$Null` on the *Parameters* parameter. You don't need to include the *RemoveParameters* parameter.

Removing all the parameters from a role entry is most useful when you want to make only a few parameters available on a cmdlet and exclude all of the other parameters. If you don't want the role to have access to a cmdlet, remove the associated role entry from the role completely instead of just removing the parameters. For more information about how to remove a role entry from a role, see Remove a role entry from a role.

Caution:
You can't undo remove operations. If you mistakenly remove all the parameters from a role entry, you must add them again manually.

To remove all the parameters from a role entry, use the following syntax.

```
Set-ManagementRoleEntry <role name>\<cmdlet> -Parameters
$Null
```

This example removes all the parameters from the **Set-CASMailbox** cmdlet on the Recipient Administrators role.

```
Set-ManagementRoleEntry "Recipient Administrators\Set-
CASMailbox" -Parameters $Null
```

For detailed syntax and parameter information, see Set-ManagementRoleEntry.

Use the Shell to apply a specific set of parameters

If you want only a specific set of parameters to be included on a role entry, specify the *Parameters* parameter only. Don't include the *AddParameter* or *RemoveParameter* parameters. When you specify only the *Parameters* parameter, only the parameters you specify in the command are included on the role entry. All other parameters are removed.

To specify a specific set of parameters, use the following syntax.

```
Set-ManagementRoleEntry <role name>\<cmdlet> -Parameters
<parameter 1>, <parameter 2>, <parameter...>
```

This example includes only the *Identity*, *DisplayName*, *MissedCallNotificationEnabled*, and *PersonalAuthAttendantEnabled* parameters on the **Set-UMMailbox** cmdlet on the Seattle Mail Recipients role.

```
Set-ManagementRoleEntry "Seattle Mail Recipients\Set-
UMMailbox" -Parameters Identity, DisplayName,
MissedCallNotificationEnabled, PersonalAutoAttendantEnabled
```

For detailed syntax and parameter information, see Set-ManagementRoleEntry.

View role entries

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Each management role entry represents a single cmdlet or script. The parameters included on a

role entry determine what parameters on the cmdlet or script a user can access.

The identity of role entries consists of the management role name that the role entry is associated with, and the cmdlet or script that the role entry refers to. For more information about role entries in Microsoft Exchange Server 2013, see Understanding management roles.

Looking for other management tasks related to role entries? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- This topic makes use of pipelining, the **Format-List** cmdlet, objects, and properties. For more information about these concepts, see the following topics:
  - Pipelining
  - Working with command output
  - Structured data
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

View a list of role entries

You can use the **Get-ManagementRoleEntry** cmdlet to retrieve a list of role entries. When you use the **Get-ManagementRoleEntry** cmdlet, you must specify a value that contains both the role name that contains the role entries you want to list and also the cmdlet name of the role entry you want to list. By combining the role name and cmdlet name with the wildcard character (*), you can return specific or broad lists of role entries.

For detailed syntax and parameter information, see Get-ManagementRoleEntry.

View a list of all role entries on a role

To view a list of role entries on a specific role, use the following syntax.

```
Get-ManagementRoleEntry <role name>\*
```

This examples retrieves all the role entries on the `Recipient Administrators` role.

```
Get-ManagementRole "Recipient Administrators\*"
```

For detailed syntax and parameter information, see Get-ManagementRoleEntry.

View a list of roles that contain a specific role entry

To view a list of all the roles that contain a specific role entry, use the following syntax.

```
Get-ManagementRoleEntry *\<cmdlet name>
```

This example retrieves all the roles that contain the **Set-Mailbox** role entry.

```
Get-ManagementRoleEntry *\Set-Mailbox
```

For detailed syntax and parameter information, see Get-ManagementRoleEntry.

View a targeted list of roles that contain similar role entries

To view a list of targeted roles that contain cmdlets with similar names, use the following syntax.

```
Get-ManagementRoleEntry *<partial role name>*\*<partial cmdlet name>*
```

This example returns a list of role entries that contain the string `Mailbox` that are on roles that contain the string `Tier 1` in their names.

```
Get-ManagementRoleEntry "*Tier 1*\*Mailbox*"
```

For detailed syntax and parameter information, see Get-ManagementRoleEntry.

View a single role entry

To view the details of a single role entry, use the following syntax.

```
Get-ManagementRoleEntry <role name>\<cmdlet name> | Format-List
```

This example retrieves the details of the **Set-Mailbox** role entry on the `Recipient Administrators` role.

```
Get-ManagementRoleEntry "Recipient Administrators\Set-Mailbox" | Format-List
```

If the role entry you view has too many parameters to list using the **Format-List** cmdlet, see "View the parameters on a single role entry" later in this topic.

For detailed syntax and parameter information, see Get-ManagementRoleEntry.

View the parameters on a single role entry

Some role entries have more parameters than can be viewed by piping the results of the **Get-ManagementRoleEntry** cmdlet to the **Format-List** cmdlet. If you need to view all the parameters on a role entry, you need to directly access the **Parameters** property of the role entry object.

To view parameters stored in the **Parameters** property of a role entry object, use the following syntax.

```
(Get-ManagementRoleEntry <role name>\<cmdlet name>).Parameters
```

This example retrieves the parameters on the **Set-Mailbox** role entry on the Mail Recipients role.

```
(Get-ManagementRoleEntry "Mail Recipients\Set-Mailbox").Parameters
```

For detailed syntax and parameter information, see Get-ManagementRoleEntry.

Remove a role entry from a role

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Management role entries on a management role determine what cmdlets and parameters are available on a management role. By removing role entries or parameters on a role entry, you can restrict what users assigned the management role can perform. For more information about management role entries in Microsoft Exchange Server 2013, see Understanding management roles.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see

Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Remove a single entire role entry from a role

When you remove a role entry from a role, you remove the ability for users assigned that role to access the associated cmdlet or script.

Use the following syntax to remove an entire management role entry from a role.

```
Remove-ManagementRoleEntry <management role>\<management role entry>
```

This example removes the **Enable-MailUser** cmdlet from the Seattle Server Administrators role.

```
Remove-ManagementRoleEntry "Seattle Server Administrators\Enable-MailUser"
```

For detailed syntax and parameter information, see Remove-ManagementRoleEntry.

Remove multiple entire role entries from a role

When you remove multiple role entries from a role, you remove the ability for users assigned that role to access the associated cmdlets or scripts.

To remove multiple role entries from a role, you need to retrieve the list of role entries to remove using the **Get-ManagementRoleEntry** cmdlet. Then you need to pipe the output to the **Remove-ManagementRoleEntry** cmdlet. You can use wildcard characters with the **Get-ManagementRoleEntry** cmdlet to match multiple role entries. It's a good idea to use the *WhatIf* switch to verify that you're removing the correct role entries. Use the following syntax.

```
Get-ManagementRoleEntry <management role>\<role entry with wildcard character> | Remove-ManagementRoleEntry -WhatIf
```

This example removes all the role entries that contain the word journal from the Seattle Server Administrators role.

```
Get-ManagementRoleEntry "Seattle Server Administrators\*Journal*" | Remove-ManagementRoleEntry -WhatIf
```

When you run the command with the *WhatIf* switch, the cmdlet returns a list of all the role entries that would be removed. If the list looks correct, run the command again without the *WhatIf* switch to remove the role entries.

```
Get-ManagementRoleEntry "Seattle Server Administrators
\*Journal*" | Remove-ManagementRoleEntry
```

For detailed syntax and parameter information, see Get-ManagementRoleEntry and Remove-ManagementRoleEntry.

Remove parameters from a role entry on a role

When you remove parameters from a role entry on a role, those parameters are no longer available to users assigned the role.

Use the following syntax to remove parameters from a role entry.

```
Set-ManagementRoleEntry <management role>\<role entry> -
Parameters <parameter 1>,<parameter 2...> -RemoveParameter
```

This example removes the *MaxSafeSenders*, *MaxSendSize*, *SecondaryAddress*, and *UseDatabaseQuotaDefaults* parameters from the **Set-Mailbox** role entry on the Seattle Server Administrators role.

```
Set-ManagementRoleEntry "Seattle Server Administrators\Set-
Mailbox" -Parameters
MaxSafeSenders,MaxSendSize,SecondaryAddress,UseDatabaseQuot
aDefaults -RemoveParameter
```

For detailed syntax and parameter information, see Set-ManagementRoleEntry.

Create an unscoped role

Permissions > Advanced permissions > Management roles and role entries >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-09*

An unscoped management role can be used to provide administrators and specialist users access to Windows PowerShell scripts and non-Exchange cmdlets. You can either create an unscoped top-level role and add scripts or non-Exchange cmdlets to that role, or create a role that's based on an existing, unscoped top-level role. After an unscoped role has been created and customized, the role

can be assigned to management role groups, users, and universal security groups (USGs). Unscoped roles can't be assigned to management role assignment policies. For more information about unscoped roles, see Understanding management roles.

Caution:
Unscoped roles can be powerful because, as their name implies, no management scopes are applied to them. This means that the scripts and non-Exchange cmdlets that they contain can be run against any object in your Exchange organization. Consider this when adding scripts or non-Exchange cmdlets to an unscoped role and when assigning the unscoped role.

Note:
If you want to create a role that contains Exchange cmdlets, you must create a role that's based on an existing management role. For more information about creating roles with Exchange cmdlets, see Create a role.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- The ability to create unscoped roles isn't included in any management role group by default. You must first assign the Unscoped Role Management role to a user, or to a USG or role group of which the user is a member, before the user is able to create a role group. For more information about adding a role to a user, USG, or role group, see the following topics:
  - Manage role groups
  - Add a role to a user or USG
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create an unscoped top-level management role

If you want to make scripts or non-Exchange cmdlets available to administrators or specialists in your organization, you need to create an unscoped top-level role. Scripts and non-Exchange cmdlets can only be added to an unscoped role that's created as a top-level role, because the initial unscoped role doesn't inherit from other roles. The new, unscoped top-level role can then be

a parent to other unscoped roles that can also use the added scripts and non-Exchange cmdlets.

Here are the steps to create an unscoped top-level role:

Step 1: Create the unscoped top-level role

Unscoped top-level roles don't have a parent role. You need to specify the *UnscopedTopLevel* switch to create a role without a parent. Use the following syntax to create the new role.

```
New-ManagementRole <name of new role> -UnscopedTopLevel
```

This example creates the IT Scripts unscoped top-level role.

```
New-ManagementRole "IT Scripts" -UnscopedTopLevel
```

After it's created, the role is empty until you add scripts or non-Exchange cmdlets to it.

For detailed syntax and parameter information, see New-ManagementRole.

Step 2a: Add script management role entries

If you want to add a script to the new unscoped role, use this step. If you want to add a non-Exchange cmdlet to the new unscoped role, use Step 2b.

To add a Windows PowerShell script to an unscoped top-level role, you must add a management role entry to the role. The role entry contains the script's name and the parameters on the script that you want to make available to the role.

The script must reside in the `RemoteScripts` directory in the Microsoft Exchange Server 2013 installation path on every server running Exchange 2013 where users might connect to run the script. If a user has access to run a script, but the script isn't located on the Exchange 2013 server the user is connected to, an error occurs. By default, the path to the `RemoteScripts` directory is C:\Program Files\Microsoft\Exchange Server\V15\RemoteScripts.

After you copy the script to the appropriate Exchange 2013 servers and you decide what script parameters should be used, create the role entry using the following syntax.

```
Add-ManagementRoleEntry <unscoped top-level role name>
\<script filename> -Parameters <parameter 1, parameter 2,
parameter...> -Type Script -UnscopedTopLevel
```

This example adds the BulkProvisionUsers.ps1 script to the IT Scripts role with the *Name* and *Location* parameters.

```
Add-ManagementRoleEntry "IT Scripts\BulkProvisionUsers.ps1"
-Parameters Name, Location -Type Script -UnscopedTopLevel
```

Note:

The **Add-ManagementRoleEntry** cmdlet performs basic validation to make sure that you add only the parameters that exist in the script. However, no further validation is done after the role entry is added. If parameters are later added or removed, you must manually update the role entries that contain the script.

Step 2b: Add non-Exchange cmdlet role entries

If you want to add a non-Exchange cmdlet to the new unscoped role, use this step. If you want to add a script to the new unscoped role, use Step 2a.

To add a non-Exchange cmdlet to an unscoped top-level role, you must add a management role entry to the role. The role entry contains the cmdlet snap-in, cmdlet name, and the parameters on the cmdlet that you want to make available to the role.

If you add non-Exchange cmdlets to the new role, the cmdlets must be installed on every Exchange 2013 server where users might connect to run the cmdlets. To learn how to properly install and register the Windows PowerShell snap-ins that contain the cmdlets you want to use, refer to the documentation for your product.

After you install the Windows PowerShell snap-in that contains the cmdlets on the appropriate Exchange 2013 servers and you decide what cmdlet parameters should be used, create the role entry using the following syntax.

```
Add-ManagementRoleEntry <unscoped top-level role name>
\<cmdlet name> -PSSnapinName <snap-in name> -Parameters
<parameter 1, parameter 2, parameter...> -Type Cmdlet -
UnscopedTopLevel
```

This example adds the **Set-WidgetConfiguration** cmdlet in the Contoso.Admin.Cmdlets snap-in to the Widget Cmdlets role with the *Database* and *Size* parameters.

```
Add-ManagementRoleEntry "Widget Cmdlets\Set-
WidgetConfiguration" -PSSnapinName Contoso.Admin.Cmdlets -
Parameters Database, Size -Type Cmdlet -UnscopedTopLevel
```

Note:

The **Add-ManagementRoleEntry** cmdlet performs basic validation to make sure that you add only the parameters that exist in the cmdlet. However, no further validation is done after the role entry is added. If the cmdlet is later changed, and parameters are added or removed, you must manually update the role entries that contain the cmdlet.

Step 3: Assign the management role

The final step when you create and configure a role is to assign it to a role assignee.

Important:

Management scopes can't be configured on role assignments that assign an unscoped role.

Whether you choose to create a role assignment for a role group, user, or USG, you must choose the option to create a role assignment without a management scope.

You can assign the new role to a role group, user, or USG. For more information, see the following topics:

- Manage role groups
- Add a role to a user or USG

Create an unscoped role based on another unscoped role

If you have an existing, unscoped top-level role or other unscoped roles that you want to base new unscoped roles on, you can create child unscoped roles. The child unscoped roles can contain a subset of the scripts and cmdlets that exist on the parent unscoped roles. This is useful, for example, if you want to give only a subset of the scripts or cmdlets available on a parent unscoped role to a less experienced administrator.

Here are the steps to create an unscoped child role:

Step 1: Create the unscoped child role

New, unscoped child roles can be based on existing unscoped roles. When you create a role, an existing role and its management role entries are copied to the new role. The existing role becomes the parent to the new child role. If you create an unscoped role that's based on another unscoped role, you must choose a role that contains all the cmdlets and parameters you need to use, and then remove the ones you don't want. Child unscoped roles can't have management role entries that don’t exist in the parent role.

Note:

If you need to create an unscoped role that contains scripts or non-Exchange cmdlets that don't exist in any other unscoped role, create an unscoped top-level role. For more information, see Create an unscoped top-level management role earlier in this topic.

Use the following syntax to create the new role.

```
New-ManagementRole -Parent <existing unscoped role to copy>
-Name <name of new unscoped role>
```

This example copies the IT Global Scripts role and its management role entries to the Diagnostic IT Scripts role.

```
New-ManagementRole -Parent "IT Global Scripts" -Name
"Diagnostic IT Scripts"
```

For detailed syntax and parameter information, see New-ManagementRole.

Step 2: Change the role's management role entries

After you create your role, you need to change the role's entries. You can remove an entire role entry, which removes access to the associated script or non-Exchange cmdlet completely. Or, you can remove parameters from a role entry to remove access to those specific parameters on the associated script or non-Exchange cmdlet.

You can't add role entries or parameters on role entries unless they exist in the parent role. Because you just created a role from a parent role in Step 1, you can't add any additional role entries or parameters on role entries because they don't exist in the parent role.

When you change a role entry on a role, you can do one of the following:

- Remove a single, entire role entry.
- Remove multiple, entire role entries.
- Remove parameters from a role entry.

To remove role entries from your new role, see Remove a role entry from a role.

Step 3: Assign the management role

The final step when you create and configure a role is to assign it to a role assignee.

Important:
Management scopes can't be configured on role assignments that assign an unscoped role. Whether you choose to create a role assignment for a role group, user, or USG, you must choose the option to create a role assignment without a management scope.

You can assign the new role to a role group, user, or USG. For more information, see the following topics:

- Manage role groups
- Add a role to a user or USG

Change a role entry on an unscoped top-level role

Permissions > Advanced permissions > Management roles and role entries >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Management role entries on unscoped top-level management roles refer to the scripts and non-Exchange cmdlets, and their parameters, that you want to make available to those assigned the role. By changing the parameters available on a role entry, you control what those assigned the role can do with the script or non-Exchange cmdlet. For more information about unscoped role entries, see Understanding management roles.

Note:

If you want to change a role entry on a management role that contains Exchange cmdlets, see Change a role entry.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- The ability to change a role entry on an unscoped top-level role isn't included in any management role group by default. You must first assign the Unscoped Role Management role to a user, or to a universal security group (USG) or role group of which the user is a member, before the user is able to add or change an unscoped top-level role entry. For more information about adding a role to a user, USG, or role group, see the following topics:
  - Manage role groups
  - Add a role to a user or USG
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to add one or more parameters to a role entry

To add parameters to an unscoped top-level role entry, you need to do the following:

- Specify the parameters you want to add using the *Parameters* parameter.
- Specify the *AddParameter* parameter to indicate that you want to perform an add operation.
- Specify the *UnscopedTopLevel* parameter to indicate that you're changing a role entry on an unscoped top-level role. If you don't specify this parameter when you change a role entry on an unscoped role, an error occurs.

To add parameters to a role entry, use the following syntax.

```
Set-ManagementRoleEntry <role name>\<script or non-
```

```
Exchange cmdlet> -Parameters <parameter 1>, <parameter 2>,
<parameter...> -AddParameter -UnscopedTopLevel
```

This example adds the *EmailAddress* and *City* parameters to the **CreateUsers.ps1** script on the Recipient Administrators unscoped role.

```
Set-ManagementRoleEntry "Recipient Administrators
\CreateUsers.ps1" -Parameters EmailAddress, City -
AddParameter -UnscopedTopLevel
```

For detailed syntax and parameter information, see Set-ManagementRoleEntry.

Use the Shell to remove one or more parameters from a role entry

To remove parameters from a role entry, you need to do the following:

- Specify the parameters you want to remove using the *Parameters* parameter.
- Specify the *RemoveParameter* parameter to indicate that you want to perform a remove operation.
- Specify the *UnscopedTopLevel* parameter to indicate that you're changing a role entry on an unscoped top-level role. If you don't specify this parameter when you change a role entry on an unscoped role, an error occurs.

Caution:
You can't undo remove operations. If you mistakenly remove a parameter from a role entry, you must add it again manually.

To remove parameters from a role entry, use the following syntax.

```
Set-ManagementRoleEntry <role name>\<script or non-Exchange
cmdlet> -Parameters <parameter 1>, <parameter 2>,
<parameter...> -RemoveParameter -UnscopedTopLevel
```

This example removes the *Delay*, *Force*, and *Credential* parameters from the **Start-Widget** non-Exchange cmdlet on the Tier 1 Server Administrators role.

```
Set-ManagementRoleEntry "Tier 1 Server Administrators
\Start-Widget" -Parameters Delay, Force, Credential -
RemoveParameter -UnscopedTopLevel
```

For detailed syntax and parameter information, see Set-ManagementRoleEntry.

Use the Shell to remove all parameters from a role entry

To remove all of the parameters from a role entry, you need to do the following:

- Specify the value `$Null` on the *Parameters* parameter. You don't need to include the *RemoveParameter* parameter.
- Specify the *UnscopedTopLevel* parameter to indicate that you're changing a role entry on an unscoped top-level role. If you don't specify this parameter when you change a role entry on an unscoped role, an error occurs.

Removing all the parameters from a role entry is most useful when you want to make only a few parameters available on a script or non-Exchange cmdlet and exclude all of the other parameters.

If you don't want the role to have access to a script or non-Exchange cmdlet, remove the associated role entry from the role completely instead of just removing the parameters. For more information about how to remove a role entry from a role, see Remove a role entry from a role.

Caution:

You can't undo remove operations. If you mistakenly remove all the parameters from a role entry, you must add them again manually.

To remove all the parameters from a role entry, use the following syntax.

```
Set-ManagementRoleEntry <role name>\<script or non-Exchange cmdlet> -Parameters $Null -UnscopedTopLevel
```

This example removes all the parameters from the FindMailboxesOverQuota.ps1 script on the Recipient Administrators role.

```
Set-ManagementRoleEntry "Recipient Administrators\FindMailboxesOverQuota.ps1" -Parameters $Null -UnscopedTopLevel
```

For detailed syntax and parameter information, see Set-ManagementRoleEntry.

Use the Shell to apply a specific set of parameters

If you want only a specific set of parameters to be included on a role entry, you need to do the following:

- Specify the *Parameters* parameter only. Don't include the *AddParameter* or *RemoveParameter* parameters.
- Specify the *UnscopedTopLevel* parameter to indicate that you're changing a role entry on an unscoped role. If you don't specify this parameter when you change a role entry on an unscoped top-level role, an error occurs.

Caution:

When you specify only the *Parameters* parameter, only the parameters you specify in the command are included on the role entry. All other parameters are removed.

To specify a specific set of parameters, use the following syntax.

```
Set-ManagementRoleEntry <role name>\<script or non-Exchange cmdlet> -Parameters <parameter 1>, <parameter 2>, <parameter...> -UnscopedTopLevel
```

This example includes only the *Alias*, *DisplayName*, *WidgetConfig*, and *Enabled* parameters on the **Set-Widget** cmdlet on the Seattle Mail Recipient Admins role.

```
Set-ManagementRoleEntry "Seattle Mail Recipient Admins\Set-UMMailbox" -Parameters Alias, DisplayName, WidgetConfig, Enabled -UnscopedTopLevel
```

For detailed syntax and parameter information, see Set-ManagementRoleEntry.

Other tasks

After you change a role entry on an unscoped top-level role, you may also want to:

Add a role entry to a role

Manage role groups

Manage role group members

Add a role to a user or USG

Remove a role from a user or USG

Add a role entry to an unscoped top-level role

Permissions > Advanced permissions > Management roles and role entries >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

You can add scripts and non-Exchange cmdlets to unscoped top-level management roles if you want to make new scripts or non-Exchange cmdlets available to existing unscoped roles. These scripts and non-Exchange cmdlets are added as management role entries to unscoped top-level management roles. They can then be used by those unscoped top-level role entries or any unscoped roles derived from the top-level roles. For more information about unscoped role entries, see Understanding management roles.

Note:

If you want to change a role entry on a management role that contains Exchange cmdlets, see Change a role entry.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- The ability to add a role entry on an unscoped top-level role isn't included in any management role group by default. You must first assign the Unscoped Role Management role to a user, or to a universal security group (USG) or role group of which the user is a member, before the user is able to add an unscoped top-level role entry. For more information about adding a role to a role group, user, or USG, see the following topics:
  - Manage role groups
  - Add a role to a user or USG
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Add a script role entry to an unscoped top-level role

If you want to add a script to an existing unscoped role, use this procedure. If you want to add a non-Exchange cmdlet to an existing unscoped role, see "Add a non-Exchange cmdlet role entry to an unscoped top-level role" later in this topic.

To add a Windows PowerShell script to an unscoped top-level role, you must add a management role entry to the role. The role entry contains the script's name and the parameters on the script that you want to make available to the role.

The script must reside in the Scripts directory in the Microsoft Exchange Server 2013 installation path on every server running Exchange 2013 where users might connect to run the script. If a user has access to run a script, but the script isn't located on the Exchange 2013 server the user is connected to, an error occurs. By default, the path to the Scripts directory is C:\Program Files \Microsoft\Exchange Server\V15\Scripts.

After you copy the script to the appropriate Exchange 2013 servers and you decide what script parameters should be used, create the role entry using the following syntax.

```
Add-ManagementRoleEntry <unscoped top-level role name>
\<script filename> -Parameters <parameter 1, parameter 2,
parameter...> -Type Script -UnscopedTopLevel
```

This example adds the BulkProvisionUsers.ps1 script to the IT Scripts role with the *Name* and *Location* parameters.

```
Add-ManagementRoleEntry "IT Scripts\BulkProvisionUsers.ps1"
-Parameters Name, Location -Type Script -UnscopedTopLevel
```

Note:

The **Add-ManagementRoleEntry** cmdlet performs basic validation to make sure that you add only the parameters that exist in the script. However, no further validation is done after the role entry is added. If parameters are later added or removed, you must manually update the role entries that contain the script.

Add a non-Exchange cmdlet role entry to an unscoped top-level role

If you want to add a non-Exchange cmdlet to an existing unscoped role, use this procedure. If you want to add a script to an existing unscoped role, see "Add a script role entry to an unscoped top-level role" earlier in this topic.

To add a non-Exchange cmdlet to an unscoped top-level role, you must add a management role entry to the role. The role entry contains the cmdlet snap-in, cmdlet name, and the parameters on the cmdlet that you want to make available to the role.

If you add non-Exchange cmdlets to the new role, the cmdlets must be installed on every Exchange 2013 server where users might connect to run the cmdlets. To learn how to properly install and register the Windows PowerShell snap-ins that contain the cmdlets you want to use, refer to the documentation for your product.

After you install the Windows PowerShell snap-in that contains the cmdlets on the appropriate the Exchange 2013 servers and you decide what cmdlet parameters should be used, create the role entry using the following syntax.

```
Add-ManagementRoleEntry <unscoped top-level role name>
\<cmdlet name> -PSSnapinName <snap-in name> -Parameters
<parameter 1, parameter 2, parameter...> -Type Cmdlet -
UnscopedTopLevel
```

This example adds the **Set-WidgetConfiguration** cmdlet in the Contoso.Admin.Cmdlets snap-in to the Widget Cmdlets role with the *Database* and *Size* parameters.

```
Add-ManagementRoleEntry "Widget Cmdlets\Set-
WidgetConfiguration" -PSSnapinName Contoso.Admin.Cmdlets -
Parameters Database, Size -Type Cmdlet -UnscopedTopLevel
```

Note:

The **Add-ManagementRoleEntry** cmdlet performs basic validation to make sure that you add only the parameters that exist in the cmdlet. However, no further validation is done after the role entry is added. If the cmdlet is later changed, and parameters are added or removed, you must manually update the role entries that contain the cmdlet.

Other tasks

After you add a role entry or an unscoped top-level role, you may also want to:

Add a role entry to a role

Manage role groups

Manage role group members

Add a role to a user or USG

Remove a role from a user or USG

Management role scopes

Exchange Server 2013 > Permissions > Advanced permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-13*

The following procedures enable you to perform advanced permissions management. You should only use these procedures if management role groups and management role assignment policies don't meet the needs of your organization.

Create a regular or exclusive scope

Change a role scope

View role scopes

Remove a role scope

Control automatic mailbox distribution using database scopes

For more information about managing role groups and role assignment policies, see the following topics:

Manage role groups

Manage linked role groups

Manage role assignment policies

Create a regular or exclusive scope

Permissions > Advanced permissions > Management role scopes >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

Management role scopes determine what objects are made available to a user so that the objects can be changed using the cmdlets and parameters assigned to them. By adding a management scope, you can configure management role assignments so users can administer specific servers, databases, recipients, and other objects in your organization while being restricted from changing other objects.

Important:
When you create a regular or exclusive scope, you override the write scope that's defined on the management role you're assigning. You can't override the read scope that's configured on the management role.

You can create a custom management scope and add or change a management role assignment. If you want to create a management role assignment with a prebuilt or organizational unit (OU) management scope, see Add a role to a user or USG.

For more information about management role scopes and assignments in Microsoft Exchange Server 2013, see the following topics:

- Understanding management role scopes
- Understanding management role assignments

Looking for other management tasks related to scopes? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management scopes" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Create a custom scope

To create a custom scope, choose one of the following types of scopes.

Recipient filter scope

Recipient filter-based scopes are created by using the *RecipientRestrictionFilter* parameter on the **New-ManagementScope** cmdlet. When you create a recipient filter, in addition to the recipient properties to filter, you can specify the OU in which the filter query runs. When you specify a base OU, you further restrict the write scope of the role.

For more information about management scope filters, see Understanding management role scope filters.

Use the following syntax to create a domain restriction filter scope with a base OU.

```
New-ManagementScope -Name <scope name> -
RecipientRestrictionFilter <filter query> [-RecipientRoot
<OU>]
```

This example creates a scope that includes all mailboxes within the contoso.com/Sales OU.

```
New-ManagementScope -Name "Mailboxes in Sales OU" -
RecipientRestrictionFilter { RecipientType -eq
'UserMailbox' } -RecipientRoot "contoso.com/Sales OU"
```

Note:
You can omit the *RecipientRoot* parameter if you want the filter to apply to the entire implicit read scope of the management role and not just within a specific OU.

For detailed syntax and parameter information, see New-ManagementScope.

Server filter configuration scope

Server filter-based configuration scopes are created by using the *ServerRestrictionFilter* parameter on the **New-ManagementScope** cmdlet. A server filter enables you to create a scope that applies only to the servers that match the filter you specify.

For more information about management scope filters and for a list of filterable server properties, see Understanding management role scope filters.

Use the following syntax to create a server filter scope.

```
New-ManagementScope -Name <scope name> -
ServerRestrictionFilter <filter query>
```

This example creates a scope that includes all the servers within the 'CN=Redmond,CN=Sites,CN=Configuration,DC=contoso,DC=com' AD (Active Directory) site.

```
New-ManagementScope -Name "Servers in Seattle AD site" -
ServerRestrictionFilter { ServerSite -eq
'CN=Redmond,CN=Sites,CN=Configuration,DC=contoso,DC=com' }
```

For detailed syntax and parameter information, see New-ManagementScope.

Server list configuration scope

Server list-based configuration scopes are created by using the *ServerList* parameter on the **New-ManagementScope** cmdlet. A server list scope enables you to create a scope that applies only to the servers you specify in a list.

Use the following syntax to create a server list scope.

```
New-ManagementScope -Name <scope name> -ServerList <server
1>, <server 2...>
```

This example creates a scope that applies only to MBX1, MBX3, and MBX5.

```
New-ManagementScope -Name "Mailbox servers" -ServerList
MBX1,MBX3,MBX5
```

For detailed syntax and parameter information, see New-ManagementScope.

Database filter configuration scope

Database filter-based configuration scopes are created by using the *DatabaseRestrictionFilter* parameter on the **New-ManagementScope** cmdlet. A database filter enables you to create a scope that applies only to the databases that match the filter you specify.

Important:
Role assignments associated with database scopes are applied only to users who connect to servers running Microsoft Exchange Server 2010 Service Pack 1 (SP1) or later or Exchange 2013. If a user assigned a role assignment associated with a database scope connects to a pre-Exchange 2010 SP1 server, the role assignment isn't applied to the user, and the user won't be granted any permissions provided by the role assignment.

For more information about management scope filters and for a list of filterable database properties, see Understanding management role scope filters.

Use the following syntax to create a database restriction filter.

```
New-ManagementScope -Name <scope name> -
DatabaseRestrictionFilter <filter query>
```

This example creates a scope that includes all the databases that contain the string "Executive" in the **Name** property of the database.

```
New-ManagementScope -Name "Executive Databases" -
DatabaseRestrictionFilter { Name -Like '*Executive*' }
```

For detailed syntax and parameter information, see New-ManagementScope.

Database list configuration scope

Database list-based configuration scopes are created by using the *DatabaseList* parameter on the **New-ManagementScope** cmdlet. A database list scope enables you to create a scope that applies only to the databases you specify in a list.

Important:
Role assignments associated with database scopes are applied only to users who connect to servers running Microsoft Exchange Server 2010 Service Pack 1 (SP1) or later or Exchange 2013. If a user assigned a role assignment associated with a database scope connects to a pre-Exchange 2010 SP1 server, the role assignment isn't applied to the user, and the user won't be granted any permissions provided by the role assignment.

Use the following syntax to create a database list scope.

```
New-ManagementScope -Name <scope name> -DatabaseList
<database 1>, <database 2...>
```

This example creates a scope that applies only to the databases Database 1, Database 2, and Database 3.

```
New-ManagementScope -Name "Primary databases" -DatabaseList
"Database 1", "Database 2", "Database 3"
```

For detailed syntax and parameter information, see New-ManagementScope.

Exclusive scope

Any scope that you create with the **New-ManagementScope** cmdlet can be designated as an exclusive scope. To create an exclusive scope, you use the same commands in one of the preceding sections to create a recipient filter-based scope, server filter-based scope, server list-based scope,

database filter-based scope, or database list-based scope, and then add the *Exclusive* switch to the command.

Caution:
When you create exclusive management scopes, only the role assignees assigned exclusive scopes that contain objects to be modified can access those objects. Only those administrators assigned a role with the exclusive scope can access these exclusive, or protected, objects.

This example creates an exclusive recipient filter-based scope that matches any user in the Executives department.

```
New-ManagementScope "Executive Users Exclusive Scope" -
RecipientRestrictionFilter { Department -Eq "Executives" }
-Exclusive
```

By default, when an exclusive scope is created, you're required to acknowledge that you created an exclusive scope and that you're aware of the impact that an exclusive scope has on existing role assignments that aren't exclusive. If you want to suppress the warning, you can use the *Force* switch. This example creates the same scope as the previous example, but without a warning.

```
New-ManagementScope "Executive Users Exclusive Scope" -
RecipientRestrictionFilter { Department -Eq "Executives" }
-Exclusive -Force
```

For detailed syntax and parameter information, see New-ManagementScope.

Step 2: Add or change a management role assignment

After you create the scope, you must add it to a new or existing management role assignment.

If you create a management scope and want to add it to a new management role assignment that you're going to create, see the following topics:

- Manage role groups
- Add a role to a user or USG

If you create a management role scope and want to add it to an existing management role assignment, see the following topics:

- Manage role assignment policies
- Change a role assignment

Change a role scope

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Management role scopes determine what objects are made available to a user so that the objects can be changed using the cmdlets and parameters assigned to them. By changing a scope, you can change what objects are made available to users to create, change, or remove.

You can change a custom management scope. You can change either exclusive or regular scopes. If you change an exclusive scope, the new scope takes effect immediately. If you want to change a management role assignment with a predefined or organizational unit (OU) management scope, see Change a role assignment.

For more information about management role scopes and assignments in Microsoft Exchange Server 2013, see the following topics:

- Understanding management role scopes
- Understanding management role assignments

Looking for other management tasks related to role scopes? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management scopes" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Change the name of a scope

To change the name of a scope, use the following syntax.

```
Set-ManagementScope <current scope name> -Name <new scope name>
```

This example changes the Seattle Servers scope to Seattle Exchange Servers.

```
Set-ManagementScope "Seattle Servers" -Name "Seattle
```

```
Exchange Servers"
```

For detailed syntax and parameter information, see Set-ManagementScope.

Change a recipient filter on a scope

To change the recipient filter on a scope, use the following syntax.

```
Set-ManagementScope <scope name> -
RecipientRestrictionFilter { <new recipient filter> }
```

This example changes the recipient filter to match all the recipient objects where the **Company** property is set to contoso.

```
Set-ManagementScope "Company Scope" -
RecipientRestrictionFilter { Company -eq 'contoso' }
```

For detailed syntax and parameter information, see Set-ManagementScope.

For more information about recipient filters and to see a list of filterable recipient properties, see Understanding management role scope filters.

Change the organizational unit root on a scope

To change the OU root on a scope, use the following syntax.

```
Set-ManagementScope <scope name> -RecipientRoot <OU>
```

This example changes the OU root to the North America/Sales Sales Users OU under the contoso.com domain.

```
Set-ManagementScope "Sales Users" -RecipientRoot
"contoso.com/North America/Sales"
```

For detailed syntax and parameter information, see Set-ManagementScope.

Change a server filter on a scope

To change the server filter on a scope, use the following syntax.

```
Set-ManagementScope <scope name> -ServerRestrictionFilter
{ <new server filter> }
```

This example changes the server filter to match all the server objects where the **ServerSite** property is set to 'CN=Redmond,CN=Sites,CN=Configuration,DC=contoso,DC=com'.

```
Set-ManagementScope "Company Scope" -
ServerRestrictionFilter { ServerSite -eq
'CN=Redmond,CN=Sites,CN=Configuration,DC=contoso,DC=com' }
```

For detailed syntax and parameter information, see Set-ManagementScope.

For more information about server filters and to see a list of filterable server properties, see Understanding management role scope filters.

Change the server list on a scope

You can't change the list of servers on a scope. If you need to change the server list, you need to do the following:

1. If needed, retrieve the current server list in the scope to be replaced by using the "View a specific scope" procedure in the View role scopes topic.
2. Create a scope with the new server list by using the "Step 1: Create a custom scope" procedure in the Create a regular or exclusive scope topic.
3. Change all the management role assignments that use the old scope to use the new scope by using the "Use the Shell to change the server filter or list-based scope on a role assignment" procedure in the Change a role assignment topic.
4. Remove the old scope by using the procedure in the Remove a role scope topic.

Change a database filter on a scope

To change the database filter on a scope, use the following syntax.

```
Set-ManagementScope <scope name> -DatabaseRestrictionFilter
{ <new database filter> }
```

This example changes the database filter to match all the database objects where the **Name** property contains the string "Executive".

```
Set-ManagementScope "Database Executive Scope" -
DatabaseRestrictionFilter { Name -Like "*Executive*" }
```

For detailed syntax and parameter information, see Set-ManagementScope.

For more information about database filters and to see a list of filterable database properties, see Understanding management role scope filters.

Change the database list on a scope

You can't change the list of databases on a scope. If you need to change the database list, you need to do the following:

1. If needed, retrieve the current database list in the scope to be replaced by using the "View a specific scope" procedure in the View role scopes topic.
2. Create a scope with the new database list by using the "Step 1: Create a custom scope" procedure in the Create a regular or exclusive scope topic.
3. Change all the management role assignments that use the old scope to use the new scope by using the "Use the Shell to change the database filter or list-based scope on a role assignment" procedure in the Change a role assignment topic.
4. Remove the old scope by using the procedure in the Remove a role scope topic.

View role scopes

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Management role scopes determine what objects are made available to a user so that the objects can be changed using the cmdlets and parameters assigned to them. You can view scopes to determine what scopes have been added to your organization, the configuration of a specific scope, or what scopes are orphans.

For more information about management role scopes in Microsoft Exchange Server 2013, see Understanding management role scopes.

Looking for other management tasks related to role scopes? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management scopes" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- This topic makes use of pipelining and the **Format-List** cmdlet. For more information about these concepts, see the following topics:
  - Pipelining
  - Working with command output
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

View a specific scope

You can view the details of a scope by piping the output of the **Get-ManagementScope** cmdlet to the **Format-List** cmdlet.

To view the details of a specific scope, use the following syntax.

```
Get-ManagementScope <scope name> | Format-List
```

This example retrieves the details of the Seattle Servers scope.

```
Get-ManagementScope "Seattle Servers" | Format-List
```

For detailed syntax and parameter information, see Get-ManagementScope.

List all scopes

This example retrieves a list of scopes in your organization.

```
Get-ManagementScope
```

This cmdlet retrieves both exclusive and regular scopes. If you only want to return exclusive scopes or regular scopes, see "List all exclusive or regular scopes only" later in this topic.

For detailed syntax and parameter information, see Get-ManagementScope.

List all orphaned scopes

Orphaned scopes are scopes that haven't been associated with any management role assignments.

This examples retrieves a list of orphaned scopes.

```
Get-ManagementScope -Orphan
```

For detailed syntax and parameter information, see Get-ManagementScope.

List all exclusive or regular scopes only

By default, the **Get-ManagementScope** cmdlet returns a list of scopes that contains both exclusive and regular scopes. If you want to return only exclusive scopes or only regular scopes use the following syntax.

```
Get-ManagementScope -Exclusive < $true | $false >
```

This example returns only exclusive scopes.

```
Get-ManagementScope -Exclusive $true
```

This example returns a list of regular scopes only.

```
Get-ManagementScope -Exclusive $false
```

For detailed syntax and parameter information, see Get-ManagementScope.

Remove a role scope

Permissions > Advanced permissions > Management role scopes >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-02*

Management role scopes determine what objects are made available to a user who can then change the objects using the cmdlets and parameters assigned to the user. If you're no longer using a scope, it can be removed. For more information about management role scopes in Microsoft Exchange Server 2013, see Understanding management role scopes.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management scopes" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- Before you can remove a scope, you must remove the scope from any management role assignments that might be using it. For more information about how to remove a scope from a role assignment, see Change a role assignment.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to remove a scope

To remove a scope, use the following syntax.

```
Remove-ManagementScope <scope name>
```

For example, to remove the "Dublin Servers" scope, use the following command.

```
Remove-ManagementScope "Dublin Servers"
```

Control automatic mailbox distribution using database scopes

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-08-19*

Automatic mailbox distribution is a feature in Microsoft Exchange Server 2013 that randomly selects a mailbox database to store a new or moved mailbox when you don't specify a database explicitly. This feature can be helpful when you want to allow junior administrators or help desk staff to create mailboxes without needing to know which mailbox databases mailboxes should be created on.

You can use database management scopes to control which mailbox databases can be selected by automatic mailbox distribution. When you apply database scopes to an administrator, only the databases that match the defined database scope are available to the administrator. Because automatic mailbox distribution uses the context of the current user, it's also constrained by the database scopes applied to the administrator.

For more information about automatic mailbox distribution, database scopes, and role assignments, see the following topics:

- Automatic mailbox distribution
- Understanding management role scopes
- Understanding management role assignments

Looking for other management tasks related to scopes? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete this procedure: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Management scopes" entry in the Role management permissions topic.

- You must use the Shell to perform these procedures.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Create a database scope

In this step, decide which databases you want to include in the database scope. Also, decide whether you want to specify a static list of databases, or whether you want to create a database filter that includes only the databases that match the criteria you specify.

Important:
Role assignments associated with database scopes are applied only to users who connect to servers running Microsoft Exchange Server 2010 Service Pack 1 (SP1) or later or Exchange 2013. If a user assigned a role assignment associated with a database scope connects to a pre-Exchange 2010 SP1 server, the role assignment isn't applied to the user, and the user won't be granted any permissions provided by the role assignment.

Use a database list scope

Use a database list if you want to define a static list of mailbox databases that should be included in this scope. Use the following syntax to create a database list scope.

```
New-ManagementScope -Name <scope name> -DatabaseList
<database 1>, <database 2...>
```

This example creates a scope that applies only to the databases Database 1, Database 2, and Database 3.

```
New-ManagementScope -Name "Accounting databases" -
DatabaseList "Database 1", "Database 2", "Database 3"
```

For detailed syntax and parameter information, see New-ManagementScope.

Use a database filter scope

Use a database filter if you want to create a dynamic database scope that includes only the databases that match the criteria you specify. This can be useful if you don't want to manage the database scope after it's created and you've defined standard values for your organization that can identify specific sets of mailbox databases.

For a list of filterable database properties, see Understanding management role scope filters.

Use the following syntax to create a database filter scope.

```
New-ManagementScope -Name <scope name> -
DatabaseRestrictionFilter <filter query>
```

This example creates a scope that includes all the databases that contain the string "ACCT" in the **Name** property of the database.

```
New-ManagementScope -Name "Accounting Databases" -
DatabaseRestrictionFilter { Name -Like '*ACCT*' }
```

For detailed syntax and parameter information, see New-ManagementScope.

Step 2: Add the database scope to a management role assignment

After you create the scope, you must add it to a new or existing management role assignment. We recommend that you use management role groups to control administrative permissions, so the examples in this step use an example role group called Accounting Administrators. For more information about how to create a role group, see Manage role groups.

After you assign the role to a role group with the database scope, the members of the role group will only be able to create mailboxes on, and move mailboxes to, the databases included in the scope.

For a list of built-in roles that you can assign to role groups, see Built-in management roles.

Add a new role assignment

Use this procedure if you've just created a role group and you need to add roles to it.

Use the following syntax to create a role assignment between the management role you want to assign and the new role group, with the new database scope.

```
New-ManagementRoleAssignment -SecurityGroup <role group
name> -Role <role name> -CustomConfigWriteScope <database
scope name>
```

This example creates a role assignment between the Mail Recipients and Mail Recipient Creation roles and the Accounting Administrators role group, using the Accounting Databases database scope.

```
New-ManagementRoleAssignment -SecurityGroup "Accounting
```

```
Administrators" -Role "Mail Recipients" -
CustomConfigWriteScope "Accounting Databases"
New-ManagementRoleAssignment -SecurityGroup "Accounting
Administrators" -Role "Mail Recipient Creation" -
CustomConfigWriteScope "Accounting Databases"
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Modify an existing role assignment

Use this procedure if you have an existing role group that already has role assignments between it and the roles you want to apply the new database scope to.

This procedure uses pipelining. For more information, see Pipelining.

Use the following syntax to modify a role assignment between the management role that you want to apply the database scope to, and an existing role group.

```
Get-ManagementRoleAssignment -RoleAssignee <role group
name> -Role <role name> | Set-ManagementRoleAssignment -
CustomConfigWriteScope <database scope name>
```

This example adds the Accounting Databases database scope to the Mail Recipients and Mail Recipient Creation roles assigned to the Accounting Administrators role group.

```
Get-ManagementRoleAssignment -RoleAssignee "Accounting
Administrators" -Role "Mail Recipients" | Set-
ManagementRoleAssignment -CustomConfigWriteScope
"Accounting Databases"
Get-ManagementRoleAssignment -RoleAssignee "Accounting
Administrators" -Role "Mail Recipient Creation" | Set-
ManagementRoleAssignment -CustomConfigWriteScope
"Accounting Databases"
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment or Set-ManagementRoleAssignment.

Step 3: Add members to a role group (if applicable)

If you want to add members to a role group, see Manage role group members.

Important:
If you add members to this role group to restrict what databases they can create users on, or move mailboxes to, make sure they aren't members of other role groups that could grant extra permissions.

Step 4: Remove members from a role group (if applicable)

If you've added members to a new role group that restricts what databases they can create mailbox on, or move mailboxes to, and they're members of another role group that has additional permissions, remove them from the old role group. For more information, see Manage role group members.

Management role assignments

Exchange Server 2013 > Permissions > Advanced permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-13*

The following procedures enable you to perform advanced permissions management. You should only use these procedures if management role groups and management role assignment policies don't meet the needs of your organization.

Add a role to a user or USG

Change a role assignment

View role assignments

Remove a role from a user or USG

Delegate role assignments

For more information about managing role groups and role assignment policies, see the following topics:

Manage role groups

Manage linked role groups

Manage role assignment policies

Add a role to a user or USG

Permissions > Advanced permissions > Management role assignments >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Management role assignments can assign a management role to a user or universal security group (USG). By assigning a role to a user or USG, you enable those users to perform tasks dependent on cmdlets or scripts and their parameters defined on the management role.

If you want to assign roles to a management role group or a management role assignment policy, see the following topics:

- Manage role groups
- Manage role assignment policies

If you want to add members to a role group or assign a role assignment policy to an end user, see the following topics:

- Manage role group members
- Change the assignment policy on a mailbox

For more information, see Understanding Role Based Access Control.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role assignments" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- Although you can assign roles directly to users and USGs, the recommended method of granting permissions to administrators and end users is to use management role groups and management role assignment policies. When you use role groups and assignment policies, you simplify your permissions model.
- Role assignments are additive. This means that all the roles are added together when they're evaluated. If two roles are assigned to a user and one role contains a cmdlet but the other doesn't, the cmdlet will still be available to the user.

By default, role assignments don't grant the ability to assign roles to other users. To enable a user to assign roles to other users or USGs, see Delegate role assignments.

- If you create an assignment with a scope, the scope overrides the role's implicit write scope. However, the role's implicit read scope still applies. The new scope can't return objects outside of the role's implicit read scope. For more information, see Understanding management role scopes.
- All the procedures in this topic use the *SecurityGroup* parameter to assign roles to a USG. If you want to assign the role to a specific user, use the *User* parameter instead of the *SecurityGroup* parameter. All other syntax for each command is the same.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server,

Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a role assignment with no scope

You can create a role assignment with no scope. When you do this, the implicit read and implicit write scopes of the role apply.

Use the following syntax to assign a role to a USG without any scope.

```
New-ManagementRoleAssignment -Name <assignment name> -
SecurityGroup <USG> -Role <role name>
```

This example assigns the Exchange Servers role to the SeattleAdmins USG.

```
New-ManagementRoleAssignment -Name "Exchange
Servers_SeattleAdmins" -SecurityGroup SeattleAdmins -Role
"Exchange Servers"
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Create a role assignment with a predefined relative scope

If a predefined relative scope meets your business requirements, you can apply that scope to the role assignment rather than create a custom scope. For a list of predefined scopes and their descriptions, see Understanding management role scopes.

Use the following syntax to assign a role to a USG with a predefined scope.

```
New-ManagementRoleAssignment -Name <assignment name> -
SecurityGroup < USG> -Role <role name> -
RecipientRelativeWriteScope < MyDistributionGroups |
Organization | Self >
```

This example assigns the Exchange Servers role to the SeattleAdmins USG and applies the Organization predefined scope.

```
New-ManagementRoleAssignment -Name "Exchange
Servers_SeattleAdmins" -SecurityGroup SeattleAdmins -Role
"Exchange Servers" -RecipientRelativeWriteScope
Organization
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Create a role assignment with a recipient filter-based scope

If you created a recipient filter-based scope and want to use it with a role assignment, you need to include the scope in the command used to assign the role to a USG by using the *CustomRecipientWriteScope* parameter. If you use the *CustomRecipientWriteScope* parameter, you can't use the *RecipientOrganizationalUnitScope* parameter.

Before you can add a scope to a role assignment, you need to create one. For more information, see Create a regular or exclusive scope.

Use the following syntax to assign a role to a USG with a recipient filter-based scope.

```
New-ManagementRoleAssignment -Name <assignment name> -
SecurityGroup < USG> -Role <role name> -
CustomRecipientWriteScope <role scope name>
```

This example assigns the Mail Recipients role to the Seattle Recipient Admins USG and applies the Seattle Recipients scope.

```
New-ManagementRoleAssignment -Name "Mail Recipients_Seattle
Recipient Admins" -SecurityGroup "Seattle Recipient Admins"
-Role "Mail Recipients" -CustomRecipientWriteScope "Seattle
Recipients"
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Create a role assignment with a server or database filter or list-based configuration scope

If you created a server or database filter or list-based configuration scope and want to use it with a role assignment, you need to include the scope in the command used to assign the role to a USG by using the *CustomConfigWriteScope* parameter.

Before you can add a scope to a role assignment, you need to create one. For more information, see Create a regular or exclusive scope.

Use the following syntax to assign a role to a USG with a configuration scope.

```
New-ManagementRoleAssignment -Name <assignment name> -
```

```
SecurityGroup <USG> -Role <role name> -
CustomConfigWriteScope <role scope name>
```

This example assigns the Exchange Servers role to the MailboxAdmins USG and applies the Mailbox Servers scope.

```
New-ManagementRoleAssignment -Name "Exchange
Servers_MailboxAdmins" -SecurityGroup MailboxAdmins -Role
"Exchange Servers" -CustomConfigWriteScope "Mailbox
Servers"
```

The preceding example shows how to add a role assignment with a server configuration scope. The syntax to add a database configuration scope is the same. You specify the name of a database scope instead of a server scope.

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Create a role assignment with an OU scope

If you want to scope a role's write scope to an organizational unit (OU), you can specify the OU in the *RecipientOrganizationalUnitScope* parameter directly. If you use the *RecipientOrganizationalUnitScope* parameter, you can't use the *CustomRecipientWriteScope* parameter.

Use the following syntax to assign a role to a USG and restrict the write scope of a role to a specific OU.

```
New-ManagementRoleAssignment -Name <assignment name> -
SecurityGroup <USG> -Role <role name> -
RecipientOrganizationalUnitScope <OU>
```

This example assigns the Mail Recipients role to the SalesRecipientAdmins USG and scopes the assignment to the sales/users OU in the contoso.com domain.

```
New-ManagementRoleAssignment -Name "Mail
Recipients_SalesRecipientAdmins" -SecurityGroup
SalesRecipientAdmins -Role "Mail Recipients" -
RecipientOrganizationalUnitScope contoso.com/sales/users
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Create a role assignment with an exclusive recipient or

configuration scope

To create an exclusive role assignment with an exclusive recipient or configuration scope, the same procedures provided in the Create a role assignment with a recipient filter-based scope and Create a role assignment with a server or database filter or list-based configuration scope sections can be used. The only difference is that when you create a role assignment with an exclusive scope, you must specify the following exclusive parameters depending on whether you're using an exclusive recipient scope or an exclusive configuration scope:

- **Exclusive recipient scopes** Use the *ExclusiveRecipientWriteScope* parameter instead of the *CustomRecipientWriteScope* parameter.
- **Exclusive configuration scopes** Use the *ExclusiveConfigWriteScope* parameter instead of the *CustomConfigWriteScope* parameter.

When you perform this procedure, the role assignees assigned the role can perform actions against the objects included in the exclusive scope. For more information about exclusive scopes, see Understanding exclusive scopes.

You can't create a role assignment with both exclusive and regular scopes.

This example assigns the Mail Recipients role to the Protected User Admins USG and applies the Protected Users exclusive scope.

```
New-ManagementRoleAssignment -Name "Mail
Recipients_Protected User Admins" -SecurityGroup "Protected
User Admins" -Role "Mail Recipients" -
ExclusiveRecipientWriteScope "Protected Users"
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Change a role assignment

Permissions > Advanced permissions > Management role assignments >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Management role assignments assign a management role to a role assignee. By changing the role assignment, you can control what objects role assignees assigned a role can change. Management role scopes applied to role assignments override the role's implicit write scope. However, the role's implicit read scope still applies. Scopes that you apply can't return objects outside of the role's implicit read scope.

For more information about management role scopes and assignments in Microsoft Exchange

Server 2013, see the following topics:

- Understanding management role assignments
- Understanding management role scopes

Looking for other management tasks related to role assignments? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role assignments" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable a role assignment

Role assignments are enabled by default, meaning that the associated role is applied to the role assignee to which the role is assigned. If a role assignment is disabled, the associated role isn't applied to the role assignee.

To enable a role assignment, use the following syntax.

```
Set-ManagementRoleAssignment <role assignment> -Enabled
$true
```

To disable a role assignment, use the following syntax.

```
Set-ManagementRoleAssignment <role assignment> -Enabled
$false
```

This example disables the Help Desk Assignment role assignment.

```
Set-ManagementRoleAssignment "Help Desk Assignment" -
Enabled $false
```

For detailed syntax and parameter information, see Set-ManagementRoleAssignment.

Use the Shell to change a management role or role assignee on a role assignment

You can't change the management role or role assignee specified on a role assignment. If you want a role assignment to be associated with another role or role assignee, you must create a new role assignment, and then delete the old role assignment. For more information about how to add and remove role assignments, see the following topics:

- Add a role to a user or USG
- Remove a role from a user or USG

If you've created assignments directly to a user or universal security group (USG), we recommend that you consider using management role groups and management role assignment policies. Role groups and assignment policies enable you to simplify your permissions model and reduce the number of role assignments you need to manage. For more information, see Understanding Role Based Access Control.

Use the Shell to change a predefined relative scope on a role assignment

You can change or add a predefined relative scope on a role assignment. If you add or change a predefined scope, any previously specified recipient scopes are removed from the role assignment. For a list of predefined scopes and their descriptions, see Understanding management role scopes.

To change or add a predefined scope on a role assignment, use the following syntax.

```
Set-ManagementRoleAssignment <assignment name> -
RecipientRelativeWriteScope < MyDistributionGroups |
Organization | Self >
```

This example changes the predefined scope on the John's Assignment role assignment to MyDistributionGroups.

```
Set-ManagementRoleAssignment "John's Assignment" -
RecipientRelativeWriteScope MyDistributionGroups
```

For detailed syntax and parameter information, see Set-ManagementRoleAssignment.

Use the Shell to change a recipient filter scope on a role assignment

You can either specify a new recipient filter-based scope or change the recipient filter-based scope that's already applied to the role assignment. If you add a recipient filter scope, any previously defined recipient scopes are removed from the role assignment.

To specify a new recipient filter-based scope or replace an existing one, use the following syntax.

```
Set-ManagementRoleAssignment <assignment name> -
CustomRecipientWriteScope <role scope name>
```

This example adds or changes the recipient filter-based scope to Redmond Recipients.

```
Set-ManagementRoleAssignment "Redmond Recipient
Administrators Assignment" -CustomRecipientWriteScope
"Redmond Recipients"
```

If you want to keep the same recipient filter-based scope that's applied to the role assignment but change the recipient filter used to match recipient objects, you need to change the recipient filter on the scope itself. For more information about how to change scopes, see Change a role scope.

For detailed syntax and parameter information, see Set-ManagementRoleAssignment.

Use the Shell to change the server filter or list-based configuration scope on a role assignment

You can either specify a new server filter or list-based configuration scope, or change the scope that's already applied to the role assignment. If you add or change the configuration scope, any previously specified configuration scopes are removed from the role assignment.

To specify a new configuration scope or replace an existing one, use the following syntax.

```
Set-ManagementRoleAssignment <assignment name> -
CustomConfigWriteScope <role scope name>
```

This example adds or changes the configuration scope to Redmond Servers.

```
Set-ManagementRoleAssignment "Redmond Administrators
Assignment" -CustomConfigWriteScope "Redmond Servers"
```

If you want to keep the same configuration scope that's applied to the role assignment but change the server filter or server list on the scope, you need to change the configuration scope itself. For more information about how to change scopes, see Change a role scope.

For detailed syntax and parameter information, see Set-ManagementRoleAssignment.

Use the Shell to change the database filter or list-based configuration scope on a role assignment

You can either specify a new database filter or list-based configuration scope, or change the scope that's already applied to the role assignment. If you add or change the configuration scope, any previously specified configuration scopes are removed from the role assignment.

To specify a new configuration scope or replace an existing one, use the following syntax.

```
Set-ManagementRoleAssignment <assignment name> -
CustomConfigWriteScope <role scope name>
```

This example adds or changes the configuration scope to Redmond Databases.

```
Set-ManagementRoleAssignment "Redmond Database Admins" -
CustomConfigWriteScope "Redmond Databases"
```

If you want to keep the same configuration scope that's applied to the role assignment but change the database filter or database list on the scope, you need to change the configuration scope itself. For more information about how to change scopes, see Change a role scope.

For detailed syntax and parameter information, see Set-ManagementRoleAssignment.

Use the Shell to change the organizational unit on a role assignment

You can either add a new OU or change an OU that's already applied to the role assignment. If you specify a new OU, any previously specified recipient scopes are removed from the role assignment.

To change or add a new OU on a role assignment, use the following syntax.

```
Set-ManagementRoleAssignment <assignment name> -
RecipientOrganizationalUnitScope <OU>
```

This example adds the Engineering\Users OU in the contoso.com domain to the Engineering Help Desk role assignment.

```
Set-ManagementRoleAssignment "Engineering Help Desk" -
RecipientOrganizationalUnitScope contoso.com/Engineering/
Users
```

For detailed syntax and parameter information, see Set-ManagementRoleAssignment.

Use the Shell to change an exclusive recipient or configuration scope

To change exclusive recipient or exclusive configuration scopes, you can use the procedures provided in the "Use the Shell to change a recipient filter scope on a role assignment," "Use the Shell to change the server filter or list-based configuration scope on a role assignment," and "Use the Shell to change the database filter or list-based configuration scope on a role assignment" sections earlier in this topic. The only difference is that when you change an exclusive scope, you must specify the following exclusive parameters depending on whether you're changing an exclusive recipient scope or an exclusive configuration scope:

- **Exclusive recipient scopes** Use the *ExclusiveRecipientWriteScope* parameter instead of the *CustomRecipientWriteScope* parameter.
- **Exclusive server and database configuration scopes** Use the *ExclusiveConfigWriteScope* parameter instead of the *CustomConfigWriteScope* parameter.

As with regular recipient and configuration scopes, if you add or change an exclusive scope, any previously defined recipient or configuration scopes are replaced.

This example changes an exclusive recipient write scope.

```
Set-ManagementRoleAssignment "Exclusive Executive Users" -
ExclusiveRecipientWriteScope "Exclusive Executives"
```

For detailed syntax and parameter information, see Set-ManagementRoleAssignment.

View role assignments

Permissions > Advanced permissions > Management role assignments >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Management role assignments assign a management role to a role assignee. For more information about management role assignments in Microsoft Exchange Server 2013, see Understanding management role assignments.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "Role assignments" entry in the Role management permissions topic.

- You must use the Shell to perform these procedures.
- This topic makes use of pipelining and the **Format-List** cmdlet. For more information about these concepts, see the following topics:
  - Pipelining
  - Working with command output
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

View a list of all role assignments

You can view a list of all role assignments configured in your organization by running the **Get-ManagementRoleAssignment** cmdlet. If you want to retrieve a list of role assignments that match a partial string that you specify, use wildcard characters (*). This example retrieves a list of all the role assignments that start with the string "Tier 1".

```
Get-ManagementRoleAssignment "Tier 1*"
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

View the details of a specific role assignment

You can view the details of a role assignment by piping the results of the **Get-ManagementRoleAssignment** cmdlet to the **Format-List** cmdlet. Use the following syntax.

```
Get-ManagementRoleAssignment <assignment name> | Format-List
```

This example retrieves the details of the Help Desk Assignment role assignment.

```
Get-ManagementRoleAssignment "Help Desk Assignment" | Format-List
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

View the list of role assignments assigned to a specific role

assignee

To view a list of role assignments associated with a management role group, role, or role assignment policy, or associated with a user or universal security group (USG), use the following syntax.

```
Get-ManagementRoleAssignment -RoleAssignee <role assignee name>
```

This example retrieves all of the role assignments associated with the Server Management role group.

```
Get-ManagementRoleAssignment -RoleAssignee "Server Management"
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

View the role assignments associated with a specific role

Each role can have multiple role assignments. You can use the **Get-ManagementRoleAssigment** cmdlet to view a list of role assignments associated with a specified role.

To view a list of role assignments associated with a specified role, use the following syntax.

```
Get-ManagementRoleAssignment -Role <role name>
```

This example retrieves all of the role assignments associated with the Mail Recipients role.

```
Get-ManagementRoleAssignment -Role "Mail Recipients"
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

View a list of role assignments that use a specific predefined scope

To view a list of role assignments that use a specific predefined scope, use the following syntax.

```
Get-ManagementRoleAssignment -RecipientWriteScope < MyGAL | MyDistributionGroups | Organization | Self | CustomRecipientScope | ExecutiveRecipientScope >
```

This example retrieves all of the role assignments that use the Organization predefined scope.

```
Get-ManagementRoleAssignment -RecipientWriteScope
Organization
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

View a list of role assignments that have been scoped to a specific OU

To view a list of role assignments that have been scoped to a specific organizational unit (OU), use the following syntax.

```
Get-ManagementRoleAssignment -
RecipientOrganizationalUnitScope <OU>
```

This example retrieves all of the role assignments that have been scoped to the North America \Engineering\Users OU in the contoso.com domain.

```
Get-ManagementRoleAssignment -
RecipientOrganizationalUnitScope "contoso.com/North
America/Engineering/Users"
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

View a list of assignments that use a specific custom scope

To view a list of role assignments that use a specific custom scope, you need to first determine whether the scope is a recipient scope, configuration scope, exclusive recipient scope, or exclusive configuration scope. Each type of scope uses a different parameter on the **Get-ManagementRoleAssignment** cmdlet. The following lists each scope and its associated parameter:

- **Recipient scopes** *CustomRecipientWriteScope*
- **Configuration scopes** *CustomConfigWriteScope*
- **Exclusive recipient scopes** *ExclusiveRecipientWriteScope*
- **Exclusive configuration scopes** *ExclusiveConfigWriteScope*

The syntax for each parameter is the same. Specify the name of the scope with the parameter that matches the type of scope it is.

This example retrieves all of the role assignments that use the Vancouver Recipients recipient scope.

```
Get-ManagementRoleAssignment -CustomRecipientWriteScope
"Vancouver Recipients"
```

This example retrieves all of the role assignments that use the Seattle AD Site exclusive configuration scope.

```
Get-ManagementRoleAssignment -ExclusiveConfigWriteScope
"Seattle AD Site"
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

View a list of exclusive or regular scopes

To view a list of exclusive or regular role assignments, use the following syntax.

```
Get-ManagementRoleAssignment -Exclusive < $True | $False >
```

For example, to view a list of exclusive scopes, run the following command:

```
Get-ManagementRoleAssignment -Exclusive $True
```

This example retrieves a list of regular scopes without any exclusive scopes.

```
Get-ManagementRoleAssignment -Exclusive $False
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

View who can modify a specific recipient or server

To view a list of role assignments that can modify a specific recipient or server, use the *WritableRecipient* and *WritableServer* parameters. Specify the name of the recipient with the *WritableRecipient* parameter, and the name of the server with the *WritableServer* parameter.

This example retrieves a list of role assignments that can modify the recipient Brian.

```
Get-ManagementRoleAssignment -WritableRecipient "Brian"
```

You can combine the *WritableRecipient* and *WritableServer* parameters with other parameters, such as the *RoleAssignee* parameter and the *GetEffectiveUsers* switch to refine your query and expand any role groups or USGs. This example retrieves all of the users who can modify the server EX02 and who are assigned the Server Management role group.

```
Get-ManagementRoleAssignment -WritableServer EX02 -
RoleAssignee "Server Management" -GetEffectiveUsers
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

View the users who receive permissions from an assignment via a role group or USG

To view a list of users that receive permissions from a role assignment, use the following syntax.

```
Get-ManagementRoleAssignment <assignment name> -
GetEffectiveUsers
```

This example retrieves a list of users in the Help Desk Assignment role assignment.

```
Get-ManagementRoleAssignment "Help Desk Assignment" -
GetEffectiveUsers
```

You can also combine the *GetEffectiveUsers* switch with several other parameters on the **Get-ManagementRoleAssignment** cmdlet to expand the role groups and USGs that the role assignments are assigned to. For an example of how the *GetEffectiveUsers* switch is used with other parameters, see "View who can modify a specific recipient or server" earlier in this topic.

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

View a list of role assignments that are enabled or disabled

To view a list of role assignments that are enabled or disabled, use the following syntax.

```
Get-ManagementRoleAssignment -Enabled < $True | $False >
```

This example retrieves a list of role assignments that are disabled.

```
Get-ManagementRoleAssignment -Enabled $False
```

For detailed syntax and parameter information, see Get-ManagementRoleAssignment.

Remove a role from a user or USG

Permissions > Advanced permissions > Management role assignments >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-02*

Management role assignments assign a management role to a user or universal security group (USG). If you remove a role assignment, the users assigned the role will no longer have access to the

cmdlets available on that role. For more information about management role assignments in Microsoft Exchange Server 2013, see Understanding management role assignments.

Looking for other management tasks related to roles? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete this procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role assignments" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Remove a management role assignment

If you know the name of the role assignment you want to remove, use the following syntax.

```
Remove-ManagementRoleAssignment <assignment name>
```

For example, to remove the "Tier 2 Help Desk Assignment" role assignment, use the following command.

```
Remove-ManagementRoleAssignment "Tier 2 Help Desk Assignment"
```

If you don't know the name of the role assignment, you can use the following syntax.

```
Get-ManagementRoleAssignment -RoleAssignee <user or USG> -Role <role name> -Delegating <$true | $false> | Remove-ManagementRoleAssignment
```

For example, if you want to remove the Mail Recipients regular role assignment from the user davids, use the following command.

```
Get-ManagementRoleAssignment -RoleAssignee davids -Role "Mail Recipients" -Delegating $false | Remove-ManagementRoleAssignment
```

Delegate role assignments

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-02*

Management role delegation enables role assignees to assign a specified management role to other management role groups, management role assignment policies, users, or universal security groups (USG). By default, only members of the Organization Management management role group can delegate role assignments. When a new installation of Microsoft Exchange Server 2013 is deployed, only the user account that installed Exchange 2013 is a member of the Organization Management role group.

If you assign a delegating role assignment to a role group, any member of the role group can delegate the associated management role to other role assignees.

Important:

Delegating role assignments doesn't give the role assignee the permissions granted by the role, only the ability to assign the role to others. If you want to also give the permissions granted by the role to the role assignee, you must also create a regular role assignment. To create a regular role assignment, see the following topics:
Manage role groups
Manage role assignment policies
Add a role to a user or USG

Note:

This topic discusses management role assignment delegation. If you want to delegate who can add members to or remove members from role groups, which is the recommended method of delegation, see Manage role groups.

For more information about regular role assignments and delegating management role assignments, see Understanding management role assignments.

Looking for other management tasks related to managing permissions? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete this procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role assignments" entry in the Role management permissions topic.
- You must use the Shell to perform these procedures.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see

Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to delegate a management role

You can create delegating role assignments using the same predefined scopes, recipient filter or server-filter-based scopes, server list-based scopes, and organizational unit (OU) scopes that can be used to create regular or exclusive scopes. The only difference between creating a regular role assignment and a delegating role assignment is the addition of the *Delegating* switch to the command. For more information about how to create role assignments, see the following topics:

- Manage role groups
- Add a role to a user or USG

Note:
You can't create a delegating role assignment to a management role assignment policy.

This example creates a delegating role assignment to enable members of the Senior Admins role group to assign the Mail Recipients role to any role assignee in the Exchange organization.

```
New-ManagementRoleAssignment -Role "Mail Recipients" -
SecurityGroup "Senior Admins" -Name "Mail Recipients_Senior
Admin - Delegate" -Delegating
```

This example creates a delegating role assignment to enable members of the Senior Admins role group to assign the Mail Recipients role only to users in the Sales/Users OU in the contoso.com domain.

```
New-ManagementRoleAssignment -Role "Mail Recipients" -
SecurityGroup "Senior Admins" -Name "Mail Recipients_Senior
Admins - Delegate" -RecipientOrganizationalUnitScope
contoso.com/sales/users -Delegating
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Managing split permissions

Exchange Server 2013 > Permissions > Advanced permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-13*

The following procedures enable you to perform advanced permissions management. You should only use these procedures if management role groups and management role assignment policies don't meet the needs of your organization.

Configure Exchange 2013 for split permissions

Configure Exchange 2013 for shared permissions

For more information about managing role groups and role assignment policies, see the following topics:

Manage role groups

Manage linked role groups

Manage role assignment policies

Configure Exchange 2013 for split permissions

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

Split permissions enable two separate groups, such as Active Directory administrators and Microsoft Exchange Server 2013 administrators, to manage their respective services, objects, and attributes. Active Directory administrators manage security principals, such as users, that provide permissions to access an Active Directory forest. Exchange administrators manage the Exchange-related attributes on Active Directory objects and Exchange-specific object creation and management.

Microsoft Exchange Server 2013 offers the following types of split permissions models:

- **RBAC split permissions** Permissions to create security principals in the Active Directory domain partition are controlled by Role Based Access Control (RBAC). Only those who are members of the appropriate role groups can create security principals.
- **Active Directory split permissions** Permissions to create security principals in the Active Directory domain partition are completely removed from any Exchange user, service, or server. No option is provided in RBAC to create security principals. Creation of security principals in Active Directory must be performed using Active Directory management tools.

Note:
Active Directory split permissions are available in organizations running Microsoft Exchange

Server 2010 Service Pack 1 (SP1) or later, Exchange 2013, or both versions of Exchange.

The model that you choose depends on the structure and needs of your organization. Choose the procedure that follows that's applicable to the model you want to configure. We recommend that you use the RBAC split permissions model. The RBAC split permissions model provides significantly more flexibility while providing the same administration separation as Active Directory split permissions.

For more information about shared and split permissions, see Understanding split permissions.

For more information about management role groups, management roles, and regular and delegating management role assignments, see the following topics:

- Understanding Role Based Access Control
- Understanding management role groups
- Understanding management roles
- Understanding management role assignments

Looking for other management tasks related to permissions? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Active Directory split permissions" entry in the Role management permissions topic.
- You must use Windows PowerShell, Windows Command Shell, or both, to perform these procedures. For more information, see each procedure.
- If you have Exchange 2010 servers in your organization, the permissions model you select will also be applied to those servers.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Switch to RBAC split permissions

You can configure your Exchange 2013 organization for RBAC split permissions. When you are done, only Active Directory administrators will be able to create Active Directory security principals. This means that Exchange administrators won't be able to use the following cmdlets:

- **New-Mailbox**
- **New-MailContact**

- **New-MailUser**
- **New-RemoteMailbox**
- **Remove-Mailbox**
- **Remove-MailContact**
- **Remove-MailUser**
- **Remove-RemoteMailbox**

Exchange administrators will only be able to manage the Exchange attributes on existing Active Directory security principals. However, They will be able to create and manage Exchange-specific objects, such as transport rules and distribution groups. For more information, see the "RBAC Split Permissions" section in Understanding split permissions.

To configure Exchange 2013 for split permissions, you must assign the Mail Recipient Creation role and the Security Group Creation and Membership role to a role group that contains members that are Active Directory administrators. You must then remove the assignments between those roles and any role group or universal security group (USG) that contains Exchange administrators.

To configure RBAC split permissions, do the following:

1. If your organization is currently configured for Active Directory split permissions, do the following from a Windows command shell prompt.
   a. Disable Active Directory split permissions by running the following command from the Exchange 2013 installation media.

```
setup.exe /PrepareAD /ActiveDirectorySplitPermissions:false
```

   b. Restart the Exchange 2013 servers in your organization or wait for the Active Directory access token to replicate to all of your Exchange 2013 servers.

Note:
If you have Exchange 2010 servers in your organization, you also need to restart those servers.

2. Do the following from the Exchange Management Shell:
   a. Create a role group for the Active Directory administrators. In addition to creating the role group, the command creates regular role assignments between the new role group and the Mail Recipient Creation role and the Security Group Creation and Membership role.

```
New-RoleGroup "Active Directory Administrators" -Roles
"Mail Recipient Creation", "Security Group Creation and
Membership"
```

Note:
If you want members of this role group to be able to create role assignments, include the Role Management role. You don't have to add this role now. However, if you ever want to assign either the Mail Recipient Creation role or Security Group Creation and Membership role to other role assignees, the Role Management role must be assigned to this new role group. The steps that follow configure the Active Directory Administrators role group as the only role group that can delegate these roles.

b. Create delegating role assignments between the new role group and the Mail Recipient Creation role and Security Group Creation and Membership role using the following commands.

```
New-ManagementRoleAssignment -Role "Mail Recipient
Creation" -SecurityGroup "Active Directory Administrators"
-Delegating
New-ManagementRoleAssignment -Role "Security Group Creation
and Membership" -SecurityGroup "Active Directory
Administrators" -Delegating
```

c. Add members to the new role group using the following command.

```
Add-RoleGroupMember "Active Directory Administrators" -
Member <user to add>
```

d. Replace the delegate list on the new role group so that only members of the role group can add or remove members.

```
Set-RoleGroup "Active Directory Administrators" -ManagedBy
"Active Directory Administrators"
```

Important:

Members of the Organization Management role group, or those who are assigned the Role Management role, either directly or through another role group or USG, can bypass this delegate security check. If you want to prevent any Exchange administrator from adding himself or herself to the new role group, you must remove the role assignment between the Role Management role and any Exchange administrator and assign it to another role group.

e. Find all of the regular and delegating role assignments to the Mail Recipient Creation role using the following command. The command displays only the **Name**, **Role**, and **RoleAssigneeName** properties.

```
Get-ManagementRoleAssignment -Role "Mail Recipient
Creation" | Format-Table Name, Role, RoleAssigneeName -Auto
```

f. Remove all of the regular and delegating role assignments to the Mail Recipient Creation role that aren't associated with the new role group or any other role groups, USGs, or direct assignments you want to keep using the following command.

```
Remove-ManagementRoleAssignment <Mail Recipient Creation
role assignment to remove>
```

Note:

If you want to remove all of the regular and delegating role assignments to the Mail Recipient Creation role on any role assignee other than the Active Directory Administrators role group,

use the following command. The *WhatIf* switch lets you see what role assignments will be removed. Remove the *WhatIf* switch and run the command again to remove the role assignments.

```
Get-ManagementRoleAssignment -Role "Mail Recipient
Creation" | Where { $_.RoleAssigneeName -NE "Active
Directory Administrators" } | Remove-
ManagementRoleAssignment -WhatIf
```

g. Find all of the regular and delegating role assignments to the Security Group Creation and Membership role using the following command. The command displays only the **Name**, **Role**, and **RoleAssigneeName** properties.

```
Get-ManagementRoleAssignment -Role "Security Group Creation
and Membership" | Format-Table Name, Role, RoleAssigneeName
-Auto
```

h. Remove all of the regular and delegating role assignments to the Security Group Creation and Membership role that aren't associated with the new role group or any other role groups, USGs, or direct assignments you want to keep using the following command.

```
Remove-ManagementRoleAssignment <Security Group Creation
and Membership role assignment to remove>
```

Note:

You can use the same command in the preceding Note to remove all of the regular and delegating role assignments to the Security Group Creation and Membership role on any role assignee other than the Active Directory Administrators role group, as shown in this example.

```
Get-ManagementRoleAssignment -Role "Security Group Creation
and Membership" | Where { $_.RoleAssigneeName -NE "Active
Directory Administrators" } | Remove-
ManagementRoleAssignment -WhatIf
```

For detailed syntax and parameter information, see the following topics:

- New-RoleGroup
- New-ManagementRoleAssignment
- Add-RoleGroupMember
- Set-RoleGroup
- Get-ManagementRoleAssignment
- Remove-ManagementRoleAssignment

Switch to Active Directory split permissions

You can configure your Exchange 2013 organization for Active Directory split permissions. Active Directory split permissions completely remove the permissions that allow Exchange administrators and servers from creating security principals in Active Directory or modifying non-Exchange attributes on those objects. When you are done, only Active Directory administrators will be able to create Active Directory security principals. This means that Exchange administrators won't be able to use the following cmdlets:

- **Add-DistributionGroupMember**
- **New-DistributionGroup**
- **New-Mailbox**
- **New-MailContact**
- **New-MailUser**
- **New-RemoteMailbox**
- **Remove-DistributionGroup**
- **Remove-DistributionGroupMember**
- **Remove-Mailbox**
- **Remove-MailContact**
- **Remove-MailUser**
- **Remove-RemoteMailbox**
- **Update-DistributionGroupMember**

Exchange administrators and servers will only be able to manage the Exchange attributes on existing Active Directory security principals. However, they will be able to create and manage Exchange-specific objects, such as transport rules and Unified Messaging dial plans.

Caution:

After you enable Active Directory split permissions, Exchange administrators and servers will no longer be able to create security principals in Active Directory, and they won't be able to manage distribution group membership. These tasks must be performed using Active Directory management tools with the required Active Directory permissions. Before you make this change, you should understand the impact it will have on your administration processes and third-party applications that integrate with Exchange 2013 and the RBAC permissions model.
For more information, see the "Active Directory split permissions" section in Understanding split permissions.

To switch from shared or RBAC split permissions to Active Directory split permissions, do the following:

1. From a Windows command shell, run the following command from the Exchange 2013 installation media to enable Active Directory split permissions.

```
setup.exe /PrepareAD /ActiveDirectorySplitPermissions:true
```

2. If you have multiple Active Directory domains in your organization, you must either run `setup.exe /PrepareDomain` in each child domain that contains Exchange servers or objects or run `setup.exe /PrepareAllDomains` from a site that has an Active Directory server from every domain.
3. Restart the Exchange 2013 servers in your organization or wait for the Active Directory access

token to replicate to all of you Exchange 2013 servers.

Note:
If you have Exchange 2010 servers in your organization, you also need to restart those servers.

Configure Exchange 2013 for shared permissions

Permissions > Advanced permissions > Managing split permissions >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Shared permissions enable you, as an administrator of Microsoft Exchange Server 2013, to create Active Directory security principals, such as users, and then configure them as Exchange recipients. Unlike split permissions, which separate management tasks between groups of Exchange administrators and Active Directory administrators, there's no separation of tasks with shared permissions.

For more information about shared and split permissions, see Understanding split permissions.

You can configure your Exchange 2013 organization for shared permissions if you've previously set your organization for split permissions. The procedure to switch to shared permissions is different depending on whether you're currently using Role Based Access Control (RBAC) split permissions or Active Directory split permissions. Choose the procedure that follows that's applicable to your current configuration. If the following are true, your organization is using Active Directory split permissions:

- The Microsoft Exchange Protected Groups organizational unit (OU) exists.
- The Exchange Windows Permissions security group is located in the Microsoft Exchange Protected Groups OU.
- The Exchange Trusted Subsystem security group is a member of the Exchange Windows Permissions security group.
- There are no regular management role assignments to the Mail Recipient Creation role or Security Group Creation and Membership role.

If you've never configured your organization for split permissions, you don't need to perform this procedure. Exchange 2013 is configured for shared permissions by default.

For more information about management role groups, management roles, and regular and delegating management role assignments, see the following topics:

- Understanding Role Based Access Control
- Understanding management role groups
- Understanding management roles

- Understanding management role assignments

Looking for other management tasks related to permissions? Check out Advanced permissions.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- You must use Windows PowerShell, the Windows Command Shell, or both, to perform these procedures. For more information, see each procedure.
- The Exchange 2013 organization must currently be configured for RBAC or Active Directory split permissions.
- If you have Microsoft Exchange Server 2010 servers in your organization, the permissions model you select will also be applied to those servers.
- You must have permissions to delegate the Mail Recipient Creation management role and the Security Group Creation and Membership management role to the Organization Management management role group or another role group that's assigned the Mail Recipients role.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Switch from RBAC split permissions to shared permissions

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.

To switch from RBAC split permissions to Exchange 2013 shared permissions, you must assign the Mail Recipient Creation role and the Security Group Creation and Membership role to a role group that's also assigned the Mail Recipients role and has Exchange 2013 administrators as members. In the default shared permissions configuration, the Organization Management role group contains each of these roles. Because of this, the Organization Management role group is in this procedure.

Configure shared permissions

To configure shared permissions on the Organization Management role group, do the following using an account that has permissions to delegate role assignments for the Mail Recipient Creation role and the Security Group Creation and Membership role:

1. Add delegating role assignments for the Mail Recipient Creation role and Security Group Creation and Membership role to the Organization Management role group using the following

commands.

```
New-ManagementRoleAssignment -Role "Mail Recipient
Creation" -SecurityGroup "Organization Management" -
Delegating
New-ManagementRoleAssignment -Role "Security Group Creation
and Membership" -SecurityGroup "Organization Management" -
Delegating
```

Note:

The role group (in this procedure, the Active Directory Administrators role group) that has delegating role assignments for the Mail Recipient Creation role and Security Group Creation and Membership role must be assigned the Role Management role to run the **New-ManagementRoleAssignment** cmdlet. The role assignee that can delegate the Role Management role must assign that role to the Active Directory Administrators role group.

2. Add regular role assignments for the Mail Recipient Creation role to the Organization Management and Recipient Management role groups using the following commands.

```
New-ManagementRoleAssignment -Role "Mail Recipient
Creation" -SecurityGroup "Organization Management"
New-ManagementRoleAssignment -Role "Security Group Creation
and Membership" -SecurityGroup "Recipient Management"
```

3. Add a regular role assignment for the Security Group Creation and Membership role to the Organization Management role group using the following command.

```
New-ManagementRoleAssignment -Role "Security Group Creation
and Membership" -SecurityGroup "Organization Management"
```

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Remove permissions from Active Directory administrators (Optional)

You can optionally remove the permissions granted to Active Directory administrators if you no longer want them to be able to create or manage Active Directory objects using the Exchange management tools. If you want to remove permissions from Active Directory administrators, perform this procedure.

Note:

Although you can remove permissions for Active Directory administrators to manage Active Directory objects using the Exchange management tools, Active Directory administrators can continue to manage Active Directory objects using Active Directory management tools, if their Active Directory permissions allow it. They won't, however, be able to manage Exchange-specific attributes on Active Directory objects. For more information, see Understanding split permissions.

To remove Exchange-related split permissions from Active Directory administrators, do the following:

1. Remove the regular and delegating role assignments that assign the Mail Recipient Creation role to the role group or universal security group (USG) that contains the Active Directory administrators as members using the following command. This command uses the Active Directory Administrators role group as an example. The *WhatIf* switch lets you see what role assignments will be removed. Remove the *WhatIf* switch, and run the command again to remove the role assignments.

```
Get-ManagementRoleAssignment -Role "Mail Recipient
Creation" | Where { $_.RoleAssigneeName -EQ "Active
Directory Administrators" } | Remove-
ManagementRoleAssignment -WhatIf
```

2. Remove the regular and delegating role assignments that assign the Security Group Creation and Membership role to the role group or USG that contains the Active Directory administrators as members using the following command. This command uses the Active Directory Administrators role group as an example. The *WhatIf* switch lets you see what role assignments will be removed. Remove the *WhatIf* switch, and run the command again to remove the role assignments.

```
Get-ManagementRoleAssignment -Role "Security Group Creation
and Membership" | Where { $_.RoleAssigneeName -EQ "Active
Directory Administrators" } | Remove-
ManagementRoleAssignment -WhatIf
```

3. Optional. If you want to remove all Exchange permissions from the Active Directory administrators, you can remove the role group or USG in which they're members. For more information about how to remove a role group, see Manage role groups.

For detailed syntax and parameter information, see Get-ManagementRoleAssignment or Remove-ManagementRoleAssignment.

Switch from Active Directory split permissions to shared permissions

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Active Directory split permissions" entry in the Role management permissions topic.

To switch from Active Directory split permissions to Exchange 2013 shared permissions, you must rerun Exchange Setup to disable Active Directory split permissions in the Exchange organization, and then create role assignments between a role group and the Mail Recipient Creation role and Security Group Creation and Membership role. In the default shared permissions configuration, the

Organization Management role group contains each of these roles. Because of this, the Organization Management role group is in this procedure.

Important:

The setup.com command in this procedure makes changes to Active Directory. You must use an account that has the permissions required to make these changes. This account might not be the same account that has permissions to create role assignments using the **New-ManagementRoleAssignment** cmdlet. Use the account, or accounts, with the permissions necessary to successfully complete each step in this procedure.

To switch from Active Directory split permissions to shared permissions, do the following:

1. From a Windows command shell, run the following command from the Exchange 2013 installation media to disable Active Directory split permissions.

```
setup.exe /PrepareAD /ActiveDirectorySplitPermissions:false
```

2. From the Exchange Management Shell, run the following commands to add regular role assignments between the Mail Recipient Creation role and Security Group Creation and Management role and the Organization Management and Recipient Management role groups.

```
New-ManagementRoleAssignment "Mail Recipient
Creation_Organization Management" -Role "Mail Recipient
Creation" -SecurityGroup "Organization Management"
New-ManagementRoleAssignment "Security Group Creation and
Membership_Org Management" -Role "Security Group Creation
and Membership" -SecurityGroup "Organization Management"
New-ManagementRoleAssignment "Mail Recipient
Creation_Recipient Management" -Role "Mail Recipient
Creation" -SecurityGroup "Recipient Management"
```

3. Restart the Exchange 2013 servers in your organization.

Note:

If you have Exchange 2010 servers in your organization, you also need to restart those servers.

For detailed syntax and parameter information, see New-ManagementRoleAssignment.

Messaging policy and compliance

Exchange Server 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-22*

Email has become a reliable and ubiquitous communication medium for information workers in organizations of all sizes. Messaging stores and mailboxes have become repositories of valuable data. It's important for organizations to formulate messaging policies that dictate the fair use of their messaging systems, provide user guidelines for how to act on the policies, and where required, provide details about the types of communication that may not be allowed.

Organizations must also create policies to manage email lifecycle, retain messages for the length of time based on business, legal, and regulatory requirements, preserve email records for litigation and investigation purposes, and be prepared to search and provide the required email records to fulfill eDiscovery requests.

Leakage of sensitive information such as intellectual property, trade secrets, business plans, and personally identifiable information (PII) collected or handled by your organization must also be protected.

Messaging policy and compliance in Exchange 2013

The following table provides an overview of the messaging policy and compliance features in Microsoft Exchange Server 2013 and includes links to topics that will help you learn about and manage these features.

Feature	Description	Resources
Messaging records management (MRM)	To comply with applicable regulations or meet legal or business requirements, organizations include email lifecycle policies as part of their messaging policy. Common questions that should be addressed by these policies include: • How long should messages be retained? • Where should the messages be retained? • Should all messages be retained for the same period? Exchange 2013 includes MRM features that allow you to	Messaging records management

	implement your organization's email lifecycle policies. You can use MRM to apply uniform retention settings to all messages, use custom retention policies to apply a baseline retention setting for the mailbox, and optionally allow users to classify messages so that they can be retained for a specified duration.	
In-Place Archiving	*In-Place Archiving* helps you regain control of your organization's messaging data by eliminating the need for personal store (.pst) files and allowing users to store messages in an *archive mailbox* accessible in Outlook 2010 and later and Outlook Web App.	In-Place Archiving
In-Place Hold	When a reasonable expectation of litigation exists, organizations are required to preserve electronically stored information (ESI), including email that's relevant to the case. In-Place Hold allows you to search and preserve messages matching query parameters. Messages are protected from deletion, modification, and tampering and can be preserved indefinitely or for a specified period.	In-Place Hold

In-Place eDiscovery	In-Place eDiscovery allows you to search mailbox data across your Exchange organization, preview search results, and copy them to a Discovery mailbox.	In-Place eDiscovery
Journaling	Journaling can help your organization respond to legal, regulatory, and organizational compliance requirements by recording inbound and outbound email communications. When planning for messaging retention and compliance, it's important to understand journaling, how it fits in your organization's compliance policies, and how Exchange 2013 can help you secure journaled messages.	Journaling
Transport Rules	Using Transport rules, you can look for specific conditions for messages that pass through your organization and then take action on them. Transport rules let you apply messaging policies to email messages, secure messages, protect messaging systems, and prevent information leakage.	Transport rules
Data Loss Prevention (DLP)	DLP capabilities help you protect your sensitive data and inform users of your policies and regulations. DLP can also help	Data loss prevention

	you prevent users from mistakenly sending sensitive information to unauthorized people. When you configure DLP polices, you can identify and protect sensitive data by analyzing the content of your messaging system, which includes numerous associated file types. The DLP policy templates supplied in Exchange 2013 are based on regulatory standards such as PII and payment card industry data security standards (PCI-DSS). DLP is extensible, which allows you to include other policies that important to your organization. Additionally, the new Policy Tips capability allows you to inform users about policy violations before sensitive data is sent.	
Information Rights Management (IRM)	Information Rights Management (IRM) provides persistent online and offline protection for email messages and attachments using Active Directory Rights Management Services (AD RMS).	Information Rights Management
S/MIME	Secure/Multipurpose Internet Mail Extensions (S/MIME) allows people who have Office 365 mailboxes and Exchange 2013	S/MIME for message signing and encryption

	and Exchange Online to help protect sensitive information by sending signed and encrypted email within their organization. Administrators can enable S/MIME for Office 365 mailboxes by synchronizing user certificates between Office 365 and their on-premises server and then configuring Outlook Online to support S/MIME.	
Mailbox audit logging	Because mailboxes can potentially contain sensitive, high business impact (HBI) information and PII, it's important that you track who logs on to the mailboxes in your organization and what actions are taken. It's especially important to track access to mailboxes by users other than the mailbox owner (known as delegate users). Using mailbox audit logging, you can log mailbox access by mailbox owners, delegates (including administrators with full mailbox access permissions), and administrators.	Mailbox audit logging Exchange auditing reports
Administrator audit logging	Administrator audit logs enable you to keep a log of changes made by administrators to Exchange server and organization configuration and to Exchange	Administrator audit logging

	recipients. You might use administrator audit logging as part of your change control process or to track changes and access to configuration and recipients for compliance purposes.	

In-Place Archiving

Exchange Server 2013 > Messaging policy and compliance >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-25*

In-Place Archiving helps you regain control of your organization's messaging data by eliminating the need for personal store (.pst) files and allowing users to store messages in an *archive mailbox* accessible in Microsoft Outlook 2010 and later and Microsoft Office Outlook Web App.

In Microsoft Exchange Server 2013, In-Place Archiving provides users with an alternate storage location in which to store historical messaging data. An In-Place Archive is an additional mailbox (called an archive mailbox) enabled for a mailbox user. Outlook 2007 and later and Outlook Web App users have seamless access to their archive mailbox. Using either of these client applications, users can view an archive mailbox and move or copy messages between their primary mailbox and the archive. In-Place Archiving presents a consistent view of messaging data to users and eliminates the user overhead required to manage .pst files.

You can provision a user's archive on the same mailbox database as the user's primary mailbox, another mailbox database on the same Mailbox server, or a mailbox database on another Mailbox server in the same Active Directory site. This provides flexibility to use tiered storage architecture and to store archive mailboxes on a different storage subsystem, such as near-line storage. In cross-premises Exchange 2010 and later deployments, you can also provision a cloud-based archive for mailboxes located on your on-premises Mailbox servers.

Contents

Archive and retention policies

Default MRM policy

Archive quotas

In-Place Archives and other Exchange features

Managing archive mailboxes

Messaging data and .pst files

Outlook uses .pst files to store data locally on users' computers or network shares. Unlike offline store (.ost) files (which are used by Outlook in Cached Exchange Mode to store a copy of the mailbox for offline access), .pst files aren't synchronized with the user's Exchange mailbox. If a user moves messages to a .pst file, those messages are removed from the mailbox.

Using .pst files to manage messaging data can result in the following issues:

- **Unmanaged files** Generally, .pst files are created by users and reside on their computers or network shares. They aren't managed by your organization. As a result, users can create several .pst files containing the same or different messages and store them in different locations, with no organizational control.
- **Increased discovery costs** Lawsuits and some business or regulatory requirements sometimes result in discovery requests. Locating messaging data that resides in .pst files on users' computers can be a costly manual effort. Because tracking unmanaged .pst files can be difficult, .pst data may be undiscoverable in many cases. This could possibly expose your organization to legal and financial risks.
- **Inability to apply messaging retention policies** Messaging retention policies can't be applied to messages located in .pst files. As a result, depending on business or applicable regulations, your organization may not be in compliance.
- **Risk of data theft** Messaging data stored in .pst files is vulnerable to data theft. For example, .pst files are often stored in portable devices such as laptops, removable hard drives, and portable media such as USB drives, CDs, and DVDs.
- **Fragmented view of messaging data** Users who store information in .pst files don't get a uniform view of their data. Messages stored in .pst files are generally available only on the computer where the .pst file resides. As a result, if users access their mailboxes using Outlook Web App or Outlook on another computer, the messages stored in their .pst files are inaccessible.

Client access to archive mailboxes

The following table lists the client applications that can be used to access archive mailboxes.

Client	Access to archive mailbox
Outlook 2013, Outlook 2010, Outlook 2007, and Outlook Web App	Yes. Outlook 2013, Outlook 2010, Outlook 2007 and Outlook Web App users can copy or

	move items from their primary mailbox to their archive mailbox, and can also use retention policies to move items to the archive. **Note:** Outlook 2010 and later and Outlook 2007 users can also copy or move items from .pst files to their archive mailbox. Outlook 2007 users require the Office 2007 Cumulative Update for February 2011. Some differences in archive support exist between Outlook 2010 and later and Outlook 2007. For more information, see Exchange Team Blog article, see Yes Virginia, there is Exchange 2010 archive support in Outlook 2007.
Exchange ActiveSync	No

Note:
In-Place Archiving is a premium feature and requires an Exchange Enterprise client access license (CAL). For details about how to license Exchange, see Exchange Server Licensing. For details about the versions of Outlook required to access an archive mailbox, see License requirements for Personal Archive and retention policies.

Outlook doesn't create a local copy of the archive mailbox on a user's computer, even if it's configured to use Cached Exchange Mode. Users can access an archive mailbox in online mode only.

Delegate access

Delegate access is when a user or set of users is provided access to another user's mailbox. There are several scenarios for providing delegate access, including:

- Providing one or more users with access to the mailbox of a user who is no longer employed by the organization. In this case, users who may be given delegate access include the departed user's manager or supervisor or another user who will assume the departed user's responsibilities.
- Providing one or more users with access to a shared mailbox.
- Providing executive assistants with access to the mailboxes of the executives they're assisting.

In Exchange 2013, when you assign Full Access permissions to a mailbox, the delegate to which you assign the permissions can also access the user's archive. Delegates must use Outlook to access the mailbox, and they must connect to an Exchange 2010 SP1 or later Client Access server for Autodiscover purposes. Autodiscover is an Exchange service that provides configuration settings to automatically configure Outlook clients. When delegates use Outlook to access an Exchange 2013 mailbox, both the primary mailbox and the archive to which they have access are visible from

Outlook.

Moving messages to the archive mailbox

There are several ways to move messages to archive mailboxes:

- **Move or copy messages manually** Mailbox users can manually move or copy messages from their primary mailbox or a .pst file to their archive mailbox. The archive mailbox appears as another mailbox or .pst file in Outlook and Outlook Web App.
- **Move or copy messages using Inbox rules** Mailbox users can create Inbox rules in Outlook or Outlook Web App to automatically move messages to a folder in their archive mailbox. To learn more, see Learn About Inbox Rules.
- **Move messages using retention policies** You can use retention policies to automatically move messages to the archive. Users can also apply a personal tag to move messages to the archive. For details about archive and retention policies, see Archive and retention policies later in this topic.

Note:
Personal tags are available only in Outlook Web App and Outlook 2010 and later.

- **Import messages from .pst files** In Exchange 2013, you can use a mailbox import request to import messages from a .pst file to a user's archive or primary mailbox. For details, see Mailbox import and export requests. You can also use the PST Capture tool to search for .pst files on computers in your organization and import .pst file data to users' archives. Additionally, tools used to locate .pst files within an organization are also available from Microsoft partners. For a list of Microsoft partners for archiving, see "Archive and Compliance Partners" in Independent Software Vendors.

Archive and retention policies

In Exchange 2013, you can apply archive policies to a mailbox to automatically move messages from a user's primary mailbox to the archive mailbox after a specified period. Archive policies are implemented by creating retention tags that use the **Move to Archive** retention action.

Messages are moved to a folder in the archive mailbox that has the same name as the source folder in the primary mailbox. If a folder with the same name doesn't exist in the archive mailbox, it's created when the Managed Folder Assistant moves a message. Re-creating the same folder hierarchy in the archive mailbox allows users to find messages easily.

To learn more about retention policies, retention tags, and the **Move to Archive** retention action, see Retention tags and retention policies.

Default MRM policy

Exchange 2013 Setup creates a default archive and retention policy named **Default MRM Policy**.

This policy contains retention tags that have the **Move to Archive** action, as shown in the following table.

Note:
In Exchange 2010, the default archive and retention policy created by Exchange Setup is named **Default Archive and Retention Policy**.

Retention tag name	Tag type	Description
Default 2 year move to archive	Default (DPT)	Messages are automatically moved to the archive mailbox after two years. Applies to items in the entire mailbox that don't have a retention tag applied explicitly or inherited from the folder.
Personal 1 year move to archive	Personal	Messages are automatically moved to the archive mailbox after one year.
Personal 5 year move to archive	Personal	Messages are automatically moved to the archive mailbox after five years.
Personal never move to archive	Personal	Messages are never moved to the archive mailbox.
Recoverable Items 14 days move to archive	Recoverable Items Folder	Messages are moved from the Recoverable Items folder of the user's primary mailbox to the Recoverable Items folder of the archive mailbox. Users attempting to recover deleted items in the archive must use the Recover Deleted Items feature on the archive mailbox.

If you enable an In-Place Archive for a mailbox user and the mailbox doesn't already have a retention policy assigned, the default archive and retention policy is automatically assigned. After

the Managed Folder Assistant processes the mailbox, these tags become available to the user, who can then tag folders or messages to be moved to the archive mailbox. By default, email messages from the entire mailbox are moved after two years.

Before provisioning archive mailboxes for your users, we recommend that you inform them about the archive policies that will be applied to their mailbox and provide subsequent training or documentation to meet their needs. This should include details about the following:

- Functionality available within the archive, the default archive, and retention policies.
- Information about when messages may be moved automatically to the archive.
- Information about the folder hierarchy created in the archive mailbox.
- How to apply personal tags (displayed in the Archive policy menu in Outlook and Outlook Web App).

Note:

If you apply a retention policy to users who have an archive mailbox, the retention policy replaces the default MRM policy. You can create one or more retention tags with the **Move to Archive** action, and then link the tags to the retention policy. You can also add the default **Move to Archive** tags (which are created by Setup and linked to the Default MRM Policy) to any retention policies you create.

For information about compliance and archiving in Outlook 2010 and later, see Plan for compliance and archiving in Outlook 2010.

Archive quotas

Archive mailboxes are designed so that users can store historical messaging data outside their primary mailbox. Often, users use .pst files due to low mailbox storage quotas and the restrictions imposed when these quotas are exceeded. For example, users can be prevented from sending messages when their mailbox size exceeds the *Prohibit send quota*. Similarly, users can be prevented from sending and receiving messages when their mailbox size exceeds the *Prohibit send and receive quota*.

To eliminate the need for .pst files, you can provide an archive mailbox with storage limits that meet the user's requirements. However, you may still want to retain some control of the storage quotas and growth of archive mailboxes to help monitor costs and expansion.

To help with this control, you can configure archive mailboxes with an *archive warning quota* and an *archive quota*. When an archive mailbox exceeds the specified archive warning quota, a warning event is logged in the Application event log. When an archive mailbox exceeds the specified archive quota, messages are no longer moved to the archive, a warning event is logged in the Application event log, and a quota message is sent to the mailbox user. By default, in Exchange 2013, the archive warning quota is set to 45 gigabytes (GB) and the archive quota is set to 50 GB.

The following table lists the events logged and warning messages sent when the archive warning quota and archive quota are met.

Quota	Event ID	Type	Source	Category	Message

Archive warning quota	10022	Warning	MSExchangeMailboxAssistants	Managed Folder Assistant	The archive mailbox '<Display Name>:<GUID>:<Mailbox Database>:<Server FQDN>' exceeded the archive warning quota '<Archive warning quota>'. Archive mailbox size is '<Size>' bytes.
Archive quota	8537	Warning	MSExchangeIS	General	The archive mailbox for <Legacy DN> has exceeded the maximum archive mailbox size. You can't copy or move items into the archive mailbox. All message retention actions that move items to the archive

					mailbox will fail, and the primary mailbox may contain items with expired retention tags until the archive mailbox is within the maximum size limit. The mailbox owner should be notified about the condition of the archive mailbox.

In-Place Archives and other Exchange features

This section explains the functionality between In-Place Archives and various Exchange features:

- **Exchange Search** The ability to quickly search messages becomes even more critical with archive mailboxes. For Exchange Search, there's no difference between the primary and archive mailbox. Content in both mailboxes is indexed. Because the archive mailbox isn't cached on a user's computer (even when using Outlook in Cached Exchange Mode), search results for the archive are always provided by Exchange Search. When searching the entire mailbox in Outlook 2010 and later and Outlook Web App, search results include the users' primary and archive mailbox.
- **In-Place eDiscovery** When a discovery manager performs an In-Place eDiscovery search, users' archive mailboxes are also searched. There's no option to exclude archive mailboxes when creating a discovery search from the Exchange Administration Center (EAC). When using the Exchange Management Shell to create a discovery search, you can exclude the archive by using the *DoNotIncludeArchive* switch. For details, see New-MailboxSearch. To learn more, see In-Place eDiscovery.

Important:

You can't use In-Place eDiscovery to search a disconnected mailbox.

- **In-Place Hold and litigation hold** When you put a mailbox on In-Place Hold or litigation hold, the hold is placed on both the primary and the archive mailbox. To learn more about In-Place Hold and litigation hold, see In-Place Hold.
- **Recoverable Items folder** The archive mailbox contains its own Recoverable Items folder and is subject to the same Recoverable Items folder quotas as the primary mailbox. To learn more about recoverable items, see Recoverable Items folder.
- **Archiving Lync content in Exchange** You can archive instant messaging conversations and shared online meeting documents in the user's primary mailbox. The mailbox must reside on an Exchange 2013 Mailbox server and you must have Microsoft Lync 2013 deployed in your organization. For details, see Integration with SharePoint and Lync.

Managing archive mailboxes

In Exchange 2013, creating and managing archive mailboxes is integrated with common mailbox management tasks, including:

- **Creating an archive mailbox** You can create an archive mailbox when creating a mailbox, or you can enable an archive mailbox for an existing mailbox. For details, see Manage In-Place Archives.
- **Moving an archive mailbox** You can move a user's archive mailbox to another mailbox database on the same Mailbox server or to another server, independent of the primary mailbox. To move a user's archive mailbox, you must create a mailbox move request. For details, see Manage on-premises moves.

Important:

Locating a user's mailbox and archive on different versions of Exchange Server is not supported.

- **Disabling an archive mailbox** You may want to disable a user's archive mailbox for troubleshooting purposes or if you're moving the primary mailbox to a version of Exchange that doesn't support In-Place Archiving. Disabling an archive is similar to disabling a primary mailbox. For details, see Manage In-Place Archives. In on-premises deployments, a disabled archive mailbox is retained in the mailbox database until the deleted mailbox retention period for that database is reached. During this period, you can reconnect the archive to a mailbox user. When the deleted mailbox retention period is reached, the disconnected archive mailbox is purged from the mailbox database.
- **Retrieving mailbox statistics and folder statistics** You can retrieve mailbox statistics and mailbox folder statistics for a user's archive mailbox by using the *Archive* switch with the Get-MailboxStatistics and Get-MailboxFolderStatistics cmdlets.
- **Test archive connectivity** In Exchange 2013, you can use the Test-ArchiveConnectivity cmdlet to test connectivity to a specified user's on-premises or cloud-based archive.

Manage In-Place Archives

Exchange Server 2013 > Messaging policy and compliance > In-Place Archiving >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-25*

In-Place Archiving helps you regain control of your organization's messaging data by eliminating the need for personal store (.pst) files and allowing you to meet your organization's message retention and eDiscovery requirements. With archiving enabled, users can store messages in an archive mailbox, which is accessbile by using Microsoft Outlook and Outlook Web App.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place Archive" entry in the Messaging policy and compliance permissions topic.
- It's not supported to have a user's primary mailbox reside on an older Exchange version than the user's archive. If the user's primary mailbox is still on Exchange 2010, you must move it to Exchange 2013 before or at the same time when you move the archive to Exchange 2013.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a mailbox and enable an on-premises archive

Use the EAC

1. Navigate to **Recipients** > **Mailboxes**.
2. Click **New** > **User mailbox**.
3. On the **New user mailbox** page, in the **Alias** box, type an alias for the user.

Note:

If you leave this box blank, the value you type in the **User logon name** box is used for the alias.

4. Select one of the following options:
   - **Existing user** Click this button and then click **Browse** to open the **Select User – Entire Forest**

dialog box. This dialog box displays a list of Active Directory user accounts in the forest that aren't mail-enabled or don't have Exchange mailboxes. Select the user account you want to mail-enable, and then click **OK**. If you select this option, you don't have to provide user account information because this information already exists in Active Directory.

- **New user** Click this button to create a new user account in Active Directory and create a mailbox for the user. If you select this option, you'll have to provide the required user account information.

Note:

The Active Directory account that is associated with user mailboxes must reside in the same forest as the Exchange server. To create a mailbox for a user account that resides in a trusted forest, you have to create a linked mailbox. For details, see Manage linked mailboxes.

5. Click **More options** to configure the following settings.
   - **Mailbox database** Click **Browse** to select a mailbox database in which to store the mailbox. If you don't select a database, Exchange will automatically assign one.
   - **Archive** Select this check box to create an archive mailbox for the mailbox. If you create an archive mailbox, mailbox items will be moved automatically from the primary mailbox to the archive, based on the default retention policy settings or those you define.

Click **Browse** to select a database that resides in the local forest to store the archive mailbox.

To learn more, see In-Place Archiving.

   - **Address book policy** Use this list to select an address book policy (ABP) for the mailbox. ABPs contain a global address list (GAL), an offline address book (OAB), a room list, and a set of address lists. When assigned to mailbox users, an ABP provides them with access to a customized GAL in Outlook and Outlook Web App. To learn more, see Address book policies.

6. When you're finished, click **Save** to create the mailbox.

Use the Shell

This example creates the user Chris Ashton in Active Directory, creates the mailbox on mailbox database DB01, and enables an archive. The password must be reset at the next logon. To set the initial value of the password, this example creates a variable ($password), prompts you to enter a password, and assigns that password to the variable as a SecureString object.

```
$password = Read-Host "Enter password" -AsSecureString
New-Mailbox -UserPrincipalName chris@contoso.com -Alias
chris -Archive -Database "DB01" -Name ChrisAshton -
OrganizationalUnit Users -Password $password -FirstName
Chris -LastName Ashton -DisplayName "Chris Ashton"
```

For detailed syntax and parameter information, see New-Mailbox.

How do you know this worked?

To verify that you've successfully created a user mailbox with an on-premises archive, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**, and then select the new user mailbox from the list. In the details pane, under **In-Place Archive**, confirm that it is set to **Enabled**. Click **View details** to view archive properties, including archive status and the mailbox database in which it is created.
- In the Shell, run the following command to display information about the new user mailbox and archive.

```
Get-Mailbox <Name> | FL
Name,RecipientTypeDetails,PrimarySmtpAddress,*Archive*
```

- In the Shell, use the **Test-ArchiveConnectivity** cmdlet to test connectivity to the archive. For an example of how to test archive connectivity, see the Examples section in Test-ArchiveConnectivity.

Enable an on-premises archive for existing mailbox

You can also create archives for existing users that have a mailbox but aren't archive-enabled. This is known as *enabling an archive* for an existing mailbox.

Use the EAC

1. Navigate to **Recipients** > **Mailboxes**.
2. Select a mailbox.
3. In the details pane, under **In-Place Archive**, click **Enable**

Tip:
You can also bulk-enable archives by selecting multiple mailboxes (use the Shift or Ctrl keys). After selecting multiple mailboxes, in the details pane, click **More options**. Then, under **Archive** click **Enable**.

4. On the **Create in-place archive** page, click **OK** to have Exchange automatically select a mailbox database for the archive or click **Browse** to specify one.

Use the Shell

This example enables the archive for Tony Smith's mailbox.

```
Enable-Mailbox "Tony Smith" -Archive
```

This example retrieves mailboxes in database DB01 that don't have an on-premises or cloud-based archive enabled and don't have a name starting with DiscoverySearchMailbox. It pipes the result to the **Enable-Mailbox** cmdlet to enable the archive for all mailboxes on mailbox database DB01.

```
Get-Mailbox -Database DB01 -Filter {ArchiveGuid -Eq $null -
AND ArchiveDomain -eq $null -AND Name -NotLike
"DiscoverySearchMailbox*"} | Enable-Mailbox -Archive
```

For detailed syntax and parameter information, see Enable-Mailbox and Get-Mailbox.

How do you know this worked?

To verify that you've successfully enabled an on-premises archive for an existing mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**, and then select the mailbox from the list. In the details pane, under **In-Place Archive**, confirm that it is set to **Enabled**. Click **View details** to view archive properties, including archive status and the mailbox database in which it is created.
- In the Shell, run the following command to display information about the new archive.

```
Get-Mailbox <Name> | FL Name,*Archive*
```

- In the Shell, use the **Test-ArchiveConnectivity** cmdlet to test connectivity to the archive. For an example of how to test archive connectivity, see Examples in Test-ArchiveConnectivity.

Disable an on-premises archive

You may want to disable a user's archive for troubleshooting purposes or if you're moving the mailbox to a version of Exchange that doesn't support In-Place Archiving. If you disable an on-premises archive, all information in the archive will be kept in the mailbox database until the mailbox retention time passes and the archive is permanently deleted. (By default, Exchange keeps disconnected mailboxes, including archive mailboxes, for thirty days.)

Important:
Disabling the archive will remove the archive from the mailbox and mark it in the mailbox database for deletion.

If you want to reconnect the on-premises archive to that mailbox, you can use the Connect-Mailbox cmdlet with the *Archive* parameter.

Use the EAC

1. Navigate to **Recipients** > **Mailboxes**.
2. Select a mailbox.
3. In the details pane, under **In-Place Archive**, click **Disable**.

Tip:
You can also bulk-disable archives by selecting multiple mailboxes (use the Shift or Ctrl keys). After selecting multiple mailboxes, in the details pane, click **More options**. Then, under **Archive** click **Disable**.

Use the Shell

This example disables the archive for Chris Ashton's mailbox. It doesn't disable the mailbox.

```
Disable-Mailbox -Identity "Chris Ashton" -Archive
```

For detailed syntax and parameter information, see Disable-Mailbox.

How do you know this worked?

To verify that you have successfully disabled an archive, do the following:

- In the EAC, select the mailbox. Then, in the details pane check its archive status under **In-Place Archive**.
- In the Shell, run the following command to check the archive properties for the mailbox user.

```
Get-Mailbox -Identity "Chris Ashton" | Format-List
*Archive*
```

If the archive is disabled, the following values are returned for archive-related properties.

Property	Value
ArchiveDatabase (for on-premises archives)	<blank>
ArchiveState	None
DisabledArchiveDatabase (for on-premises archives)	<name of mailbox database>
DisabledArchiveGuid	<guid of disabled archive>

Connect an on-premises archive

When you disable an archive mailbox, it becomes disconnected. A disconnected archive mailbox is retained in the mailbox database for a specified amount of time. By default, Exchange retains disconnected archives for 30 days. During this time, you can recover the archive by associating it with an existing mailbox. You can modify the deleted mailbox retention period to retain a deleted mailbox or archive for a longer or shorter period.

Caution:

If you disable an archive for a user and then enable an archive for that same user, the user will get a new archive. The new archive won't contain the data that was in the user's disconnected archive. If you want to reconnect a user to his or her disconnected archive, you must perform this procedure.

Note:

You can't use the EAC to connect a disconnected archive to a mailbox user.

Use the Shell

1. If you don't know the name of the archive, you can view it in the Shell by running the following command. This example retrieves the mailbox database DB01, pipes it to the **Get-MailboxStatistics** cmdlet to retrieve mailbox statistics for all mailboxes on the database, and then uses the **Where-Object** cmdlet to filter the results and retrieve a list of disconnected archives. The command displays additional information about each archive such as the GUID and item count.

```
Get-MailboxDatabase "DB01" | Get-MailboxStatistics | Where {($_.DisconnectDate -ne $null) -and ($_.IsArchiveMailbox -eq $true)} | Format-List
```

2. Connect the archive to the primary mailbox. This example connects Chris Ashton's archive to Chris Ashton's primary mailbox and uses the GUID as the archive's identity.

```
Enable-Mailbox -ArchiveGuid "8734c04e-981e-4ccf-a547-1c1ac7ebf3e2" -ArchiveDatabase "DB01" -Identity "Chris Ashton"
```

For detailed syntax and parameter information, see the following topics:

- Get-MailboxDatabase
- Get-MailboxStatistics
- Enable-Mailbox

How do you know this worked?

To verify that you have successfully connected a disconnected archive to a mailbox user, run the following Shell command to retrieve the mailbox user's archive properties and verify the values returned for the *ArchiveGuid* and *ArchiveDatabase* properties.:

```
Get-Mailbox -Identity "Chris Ashton" | Format-List *Archive*
```

Modify archive policies

Exchange Server 2013 > Messaging policy and compliance > In-Place Archiving >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-14*

In Microsoft Exchange Server 2013, you can use archive policies to automatically move mailbox items to personal (on-premises) or cloud-based archives. Archive policies are retention tags that use the **Move to Archive** retention action.

Exchange Setup creates a retention policy called **Default MRM Policy**. This policy has a default policy tag (DPT) assigned that moves items to the archive mailbox after two years. The policy also includes a number of personal tags that users can apply to folders or mailbox items to automatically move or delete messages. If a mailbox doesn't have a retention policy assigned when it's archive-enabled, the **Default MRM Policy** is automatically applied to it by Exchange. You can also create your own archive and retention policies and apply them to mailbox users. To learn more, see Retention tags and retention policies.

You can modify retention tags included in the default policy to meet your business requirements. For example, you can modify the archive DPT to move items to the archive after three years instead of two. You can also create additional personal tags and either add them to a retention policy, including the **Default MRM Policy**, or allow users to add personal tags to their mailboxes from Outlook Web App Options.

For additional management tasks related to archives, see Manage In-Place Archives.

What do you need to know before you begin?

- Estimated time to completion: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to modify the default archive policy

1. Navigate to **Compliance management** > **Retention tags** and then.
2. In the list view, select the tag **Default 2 year move to archive** and then click **Edit** ✎.

Tip:

You can click the **TYPE** column to sort retention tags by type. The default archive policy is displayed as type **Default** and has the **Archive** retention action. Alternatively, click **NAME** to sort retention tags by name.

3. In **Retention Tag**, view or modify the following settings, and then click **Save**:
   - **Name** Use this box at the top of the page to view or change the tag name.
   - **Retention tag type** This read-only field displays the tag type.
   - **Retention action** Don't modify this field for archive policies.
   - **Retention period** Select one of the following options:
     - **Never** Click this button to disable the tag. If the DPT is disabled, the tag is no longer applied to the mailbox.

Important:

Items that have a disabled retention tag applied aren't processed by the Mailbox Assistant. If you want to prevent a tag from being applied to items, we recommend disabling the tag rather than deleting it. When you delete a tag, the tag configuration is deleted from Active Directory, and the Mailbox Assistant processes all messages to remove the deleted tag.

Note:
If a user applies a tag to an item believing the item will never be moved, enabling the tag later may move items the user wanted to retain in the primary mailbox.

- **When the item reaches the following age (in days)** Click this button to specify that items be moved to archive after a certain period. By default, this setting is configured to move items to the archive after two years (730 days). To modify this setting, in the corresponding text box, type the number of days in the retention period. The range of values is from 1 through 24,855 days.

- **Comment** Use this box to type a comment that will be displayed to Outlook and Outlook Web App users.

Use the Shell to modify archive policies

This example modifies the `Default 2 year move to archive` tag to move items after 1,095 days (3 years).

```
Set-RetentionPolicyTag "Default 2 year move to archive" -
Name "Default 3 year move to archive" -AgeLimitForRetention
1095
```

This example disables the `Default 2 year move to archive` tag.

```
Set-RetentionPolicyTag "Default 2 year move to archive" -
RetentionEnabled $false
```

This example retrieves all archive DPTs and personal tags and disables them.

```
Get-RetentionPolicyTag | ? {$_.RetentionAction -eq
"MoveToArchive"} | Set-RetentionPolicyTag -RetentionEnabled
$false
```

For detailed syntax and parameter information, see Set-RetentionPolicyTag and Get-RetentionPolicyTag.

How do you know this worked?

Use the Get-RetentionPolicyTag cmdlet to retrieve settings of the retention tag.

This command retrieves properties of the `Default 2 year move to archive` retention tag and pipes the output to the **Format-List** cmdlet to display all properties in a list format.

```
Get-RetentionPolicyTag "Default 2 year move to archive" |
Format-List
```

Configure archive quotas for an In-Place Archive (on-premises)

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-04*

In on-premises deployments, In-Place Archives are created with unlimited storage quotas by default. As a result, you'll need to edit a mailbox's properties to set storage quotas for the archive. You can set the following quotas for an archive:

- **Archive warning quota** When an In-Place Archive exceeds the specified archive warning quota, an event is logged for the Exchange administrator and a warning message is sent to the mailbox user.
- **Archive quota** When an In-Place Archive exceeds the specified archive quota, messages are no longer moved to the archive and a warning message is sent to the mailbox user.

To learn more about In-Place Archives, see In-Place Archiving.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- In the Exchange Administration Center (EAC), you can use a drop-down list with fixed values to configure the archive quota and archive warning quota. If you want to set either quota to a value that's not listed in the EAC, use the Shell.
- Configure the archive warning quota to a lower value than the archive quota. Depending on the rate of archive growth for a user, the difference between the archive warning quota and the archive quota should allow for sufficient time for the user to take appropriate actions, such as deleting items from the archive or requesting that an administrator or IT helpdesk to raise the archive quota.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to configure the archive quota and archive warning quota for a mailbox

1. Navigate to **Recipients** > **Mailboxes**
2. In the list view, select a mailbox,
3. In the details pane, under **In-Place Archive**, click **View details**.
4. In **Archive Mailbox**, use the **Quota value (GB)** and **Issue warning at (GB)** lists to select the desired values.
5. Click **OK**.

Use the Shell to configure the archive quota and archive warning quota for a mailbox

This example sets Chris Ashton's mailbox archive quota to 10 gigabyte (GB), at which time the user will receive a warning message that the In-Place Archive is full and he will no longer be able to move items to the archive. This example also sets the archive warning quota to 9.5 GB, at which time the user will receive a warning message that the In-Place Archive is almost full.

```
Set-Mailbox -Identity "Chris Ashton" -ArchiveQuota 10GB -
ArchiveWarningQuota 9.5GB
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To verify that you've successfully enabled an on-premises archive for an existing mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes** and select the mailbox you want. In the details pane, under **In-Place Archive**, click **View Details** and verify the archive's quota settings.
- In the Shell, run the following command to display quota information about the archive.

```
Get-Mailbox <Name> | FL Name,Archive*Quota
```

Archive Lync conversations and meeting content to Exchange

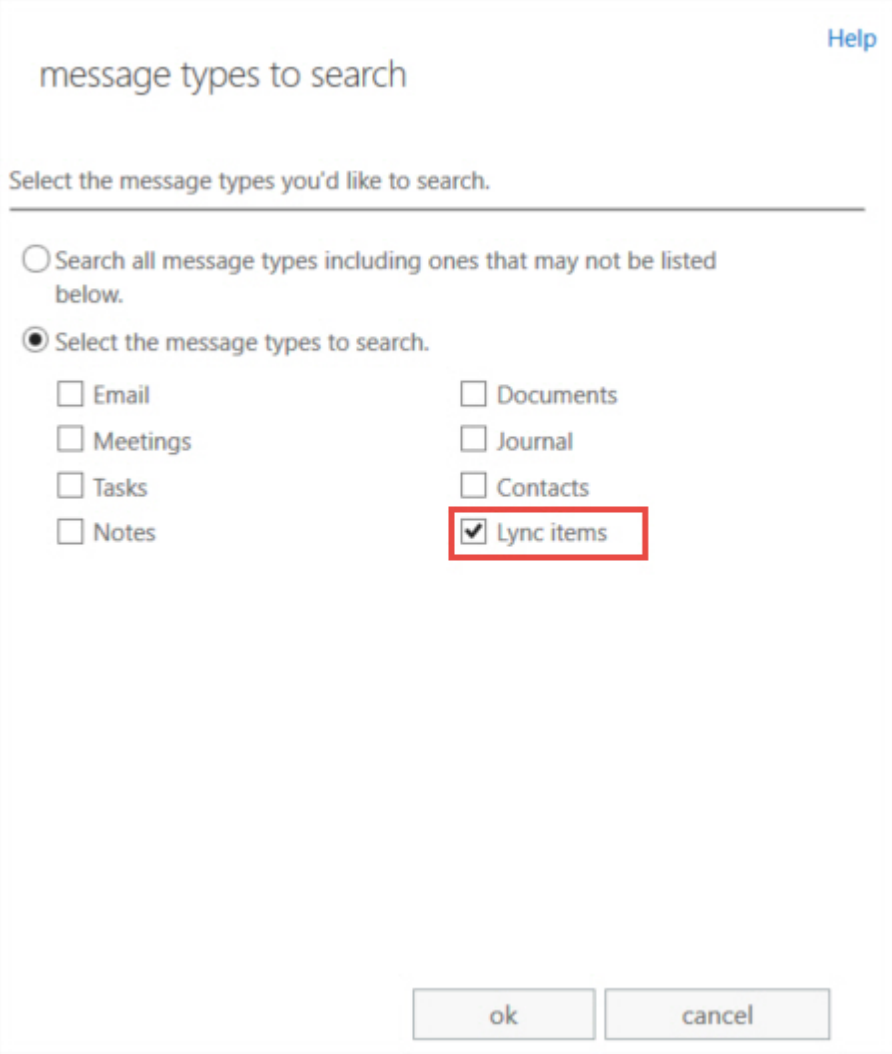
Exchange Server 2013 > Messaging policy and compliance > In-Place Archiving >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-23*

You can archive Lync Online content, such as IM conversations, to a user's mailbox in Exchange Online. In on-premises deployments, you can archive Lync 2013 content to Exchange 2013 mailboxes. This requires placing the user on In-Place Hold. Archived Lync content is preserved in the Recoverable Items folder in the user's mailbox. It's not visible to your users but it is included in eDiscovery searches.

When you create an In-Place Hold, all content types in the mailbox, including Lync items, are preserved by default. If you want to preserve only Lync items, use the **select message types** option from the **In-Place eDiscovery & Hold** wizard and select **Lync items**, as shown in the screenshot below.



You can also specify how long the content should be preserved. See In-Place Hold for details.

For step-by-step instructions for placing a user on In-Place Hold, see Create or remove an In-Place Hold.

For additional management tasks related to archiving, see **Manage In-Place Archives in Exchange Online**.

More information

- Archiving of Lync content occurs on the server, independent of whether the user has Lync client configured to save Lync IM conversations in the Conversation History folder.
- Archiving of Lync content begins after the user is placed on In-Place Hold. To ensure user's Lync communications are archived from the time their account is created, place the account on In-Place Hold immediately after it's created.

Additionally, in on-premises Exchange 2013 and Lync 2013 deployments:

- You must configure OAuth authentication between Lync 2013 and Exchange 2013. For details, see Integration with SharePoint and Lync.
- You can also archive Lync 2013 content to Exchange 2013 regardless of whether a user is placed on In-Place Hold. This is done by configuring the user's Exchange Archiving Policy. Use the `Set-CsUser` cmdlet on Lync 2013 server to set the Lync user's *ExchangeArchivingPolicy* property to `ArchivingToExchange`.
- For more details about archiving Lync content in on-premises deployments, see Planning for Archiving in Lync 2013 documentation.

Using OAuth authentication to support Archiving in an Exchange hybrid deployment

Exchange Server 2013 > Messaging policy and compliance > In-Place Archiving >

Topic Last Modified: *2014-05-22*

If you're in an Exchange 2013 hybrid deployment and use Exchange Online Archiving (EOA) for Exchange Server, you must configure OAuth authentication between your on-premises and Exchange Online organizations after upgrading to Exchange 2013 Cumulative Update 5 (CU5). EOA allows you to have a cloud-based archive for your users with on-premises mailboxes. In this scenario, the Messaging Records Management (MRM) assistant on your on-premises mailbox server applies archiving policies and moves messages automatically from a user's mailbox to their cloud-based archive. In Exchange 2013 CU5, it uses OAuth authentication.

For step-by-step instructions for configuring OAuth authentication, see Configure OAuth authentication between Exchange and Exchange Online organizations.

What is OAuth authentication?

OAuth authentication is a server-to-server authentication protocol that allows applications to

authenticate to each other. With OAuth authentication, user credentials and passwords are not passed from one computer to another. Instead, authentication and authorization is based on the exchange of security tokens, which grant access to a specific set of resources for a specific amount of time.

OAuth authentication typically involves three parties: a single authorization server and the two realms that need to communicate with one another. Security tokens are issued by the authorization server (also known as a security token server) to the two realms that need to communicate; these tokens verify that communications originating from one realm should be trusted by the other realm. When using OAuth authentication between an on-premises Exchange organization and Exchange Online, the function of the authorization server is provided by Microsoft Azure Active Directory Access Control Services (ACS) in your Office 365 organization. For example, during a cross-premises eDiscovery search, Windows Azure ACS issues tokens that verify that an administrator or compliance officer from the Exchange on-premises organization is able to access mailboxes in the Exchange Online organization, and vice-versa.

Configuring OAuth authentication to support Archiving

As previously stated, see Configure OAuth authentication between Exchange and Exchange Online organizations for instructions to configure OAuth authentication to support Archiving in an Exchange hybrid deployment.

If OAuth isn't configured for your Exchange hybrid deployment, you can't use archive policies to automatically move items from a user's primary mailbox in your on-premises organization to the user's cloud-based archive in Exchange Online.

More information

- You must also configure OAuth authentication to perform cross-premises eDiscovery searches of your on-premises and cloud-based mailboxes in a single eDiscovery search. See Using OAuth authentication to support eDiscovery in an Exchange hybrid deployment.
- You can also configure OAuth authentication to allow other applications, such as SharePoint 2013 and Lync Server 2013, to authenticate to Exchange 2013. For more information, see Configure OAuth authentication with SharePoint 2013 and Lync 2013.
- You can configure server-to-server authentication between Exchange 2013 and SharePoint 2013 so administrators and compliance officers can use the eDiscovery Center in SharePoint 2013 to search Exchange 2013 mailboxes. For more information, see Configure Exchange for SharePoint eDiscovery Center.
- You can configure an Exchange hybrid deployment using the Hybrid Configuration Wizard in Exchange 2013. For a customized, step-by-step hybrid deployment configuration checklist, see the Exchange Server Deployment Assistant.

In-Place Hold

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-11*

When a reasonable expectation of litigation exists, organizations are required to preserve electronically stored information (ESI), including email that's relevant to the case. This expectation often exists before the specifics of the case are known, and preservation is often broad. Organizations may need to preserve all email related to a specific topic or all email for certain individuals. Depending on the organization's electronic discovery (eDiscovery) practices, the following measures can be adopted to preserve email:

- End users may be asked to preserve email by not deleting any messages. However, users can still delete email knowingly or inadvertently.
- Automated deletion mechanisms such as messaging records management (MRM) may be suspended. This could result in large volumes of email cluttering the user mailbox, and thus impacting user productivity. Suspending automated deletion also doesn't prevent users from manually deleting email.
- Some organizations copy or move email to an archive to make sure it isn't deleted, altered, or tampered with. This increases costs due to the manual efforts required to copy or move messages to an archive, or third-party products used to collect and store email outside Exchange.

Failure to preserve email can expose an organization to legal and financial risks such as scrutiny of the organization's records retention and discovery processes, adverse legal judgments, sanctions, or fines.

In Exchange 2013, you can use In-Place Hold to accomplish the following goals:

- Place user mailboxes on hold and preserve mailbox items immutably
- Preserve mailbox items deleted by users or automatic deletion processes such as MRM
- Use query-based In-Place Hold to search for and retain items matching specified criteria
- Preserve items indefinitely or for a specific duration
- Place a user on multiple holds for different cases or investigations
- Keep In-Place Hold transparent from the user by not having to suspend MRM
- Enable In-Place eDiscovery searches of items placed on hold

Contents

In-Place Hold and litigation hold

Preserving archived Lync content

In-Place Hold scenarios

In Exchange Server 2010, the notion of legal hold is to hold all mailbox data for a user indefinitely or until when hold is removed. In Exchange 2013, In-Place Hold introduces a new model that allows you to specify the following parameters:

- **What to hold** You can specify which items to hold by using query parameters such as keywords, senders and recipients, start and end dates, and also specify the message types such as email messages, calendar items, etc. that you want to place on hold.
- **How long to hold** You can specify a duration for items on hold.

Using this new model, In-Place Hold allows you to create granular hold policies to preserve mailbox items in the following scenarios:

- **Indefinite hold** The indefinite hold scenario is similar to litigation hold in Exchange 2010. It's intended to preserve mailbox items so you can meet eDiscovery requirements. During the period of litigation or investigation, items are never deleted. The duration isn't known in advance, so no end date is configured. To hold all mail items indefinitely, you don't specify any query parameters or time duration when creating an In-Place Hold.
- **Query-based hold** If your organization preserves items based on specified query parameters, you can use a query-based In-Place Hold. You can specify query parameters such as keywords, start and end dates, sender and recipient addresses, and message types. After you create a query-based In-Place Hold, all existing and future mailbox items (including messages received at a later date) that match the query parameters are preserved.

Important:

Items that are marked as unsearchable, generally because of failure to index an attachment, are also preserved because it can't be determined whether they match query parameters. For more details about unsearchable item, see Unsearchable items in Exchange eDiscovery.

- **Time-based hold** Whereas litigation hold places all mailbox content on hold indefinitely or until you remove the hold, In-Place Hold allows you to specify a duration of time for which to hold items. The duration is calculated from the date a mailbox item is received or created.

If your organization requires that all mailbox items be preserved for a specific period, for example 7 years, you can create a time-based hold. In Exchange 2013, you can specify a retention period for items on hold. Items on hold are aged based on their date received. For example, consider a mailbox that's placed on a time-based In-Place Hold and has a retention period set to 365 days. If an item in that mailbox is deleted after 300 days from the date it was received, it's held for an additional 65 days before being permanently deleted. You can use a time-based In-Place Hold in conjunction with a retention policy to make sure items are preserved for the specified duration and permanently removed after that period.

You can use In-Place Hold to place a user on multiple holds. When a user is placed on multiple

holds, search parameters of all In-Place Holds are applied together (using an OR operator). If a mailbox is placed on more than five holds, all items are held until the holds are removed, replicating the litigation hold behavior until the number of holds on the mailbox is reduced to five or less.

If an administrator deletes a user account that has a mailbox, the Exchange Information store will eventually detect that the mailbox is no longer connected to a user account and mark that mailbox for deletion, even if the mailbox is on hold. If you want to retain the mailbox you must do the following:

1. Instead of deleting the user account, disable the user account.
2. Change the properties of the mailbox to restrict its use and access to the mailbox. For example, set send and receive quotas equal to 1, block who can send messages to the mailbox, and restrict who can access the mailbox.
3. Retain the mailbox until all data has been expunged, or until hold is no longer required.

Placing a mailbox on In-Place Hold

Authorized users that have been added to the Discovery Management role-based access control (RBAC) role group or assigned the Legal Hold and Mailbox Search management roles can place mailbox users on In-Place Hold. You can delegate the task to records managers, compliance officers, or attorneys in your organization's legal department, while assigning the least privileges. To learn more about assigning the Discovery Management role group, see Add a user to the Discovery Management role group.

Important:

In Exchange 2010, the Legal Hold role provided user with sufficient permissions to place mailboxes on litigation hold. In Exchange 2013, you can use the same permission to place mailboxes on an indefinite or time-based In-Place Hold. However, to create a query-based In-Place Hold, the user must be assigned the Mailbox Search role. The Discovery Management role group has both these roles assigned.

In Exchange 2013, In-Place Hold functionality is integrated with In-Place eDiscovery searches. You can use the **In-Place eDiscovery & Hold** wizard in the Exchange Administration Center (EAC) or the **New-MailboxSearch** and related cmdlets in Exchange Management Shell to place a mailbox on In-Place Hold. To learn more about placing a mailbox on In-Place Hold, see Create or remove an In-Place Hold.

Note:

If you use Exchange Online Archiving to provision a cloud-based archive for your on-premises mailboxes, you must manage In-Place Hold from your on-premises Exchange 2013 organization. Hold settings are automatically propagated to the cloud-based archive using DirSync.

Many organizations require that users be informed when they're placed on hold. Additionally, when a mailbox is on hold, any retention policies applicable to the mailbox user don't need to be

suspended. Because messages continue to be deleted as expected, users may not notice they're on hold. If your organization requires that users on hold be informed, you can add a notification message to the mailbox user's **Retention Comment** property and use the **RetentionUrl** property to link to a web page for more information. Outlook 2010 and later displays the notification and URL in the backstage area. You must use the Shell to add and manage these properties for a mailbox.

In-Place Hold and the Recoverable Items Folder

In-Place Hold uses the Recoverable Items folder to preserve items. The Recoverable Items folder replaces the feature informally known as the *dumpster* in previous versions of Exchange. The Recoverable Items folder is hidden from the default view of Outlook, Outlook Web App, and other email clients. To learn more about the Recoverable Items folder, see Recoverable Items folder.

By default, when a user deletes a message from a folder other than the Deleted Items folder, the message is moved to the Deleted Items folder. This is known as a *move*. When a user *soft deletes* an item (accomplished by pressing the SHIFT and DELETE keys) or deletes an item from the Deleted Items folder, the message is moved to the Recoverable Items folder, thereby disappearing from the user's view.

Items in the Recoverable Items folder are retained for the deleted item retention period configured on the user's mailbox database. By default, the deleted item retention period is set to 14 days for mailbox databases. You can also configure a storage quota for the Recoverable Items folder. This protects the organization from a potential denial of service (DoS) attack due to rapid growth of the Recoverable Items folder and therefore the mailbox database. If a mailbox isn't placed on In-Place Hold or litigation hold, items are purged permanently from the Recoverable Items folder on a first in, first out basis when the Recoverable Items warning quota is exceeded, or the item has resided in the folder for a longer duration than the deleted item retention period.

The Recoverable Items folder contains the following subfolders used to store deleted items in various sites and facilitate In-Place Hold and litigation hold:

1. **Deletions** Items removed from the Deleted Items folder or soft-deleted from other folders are moved to the Deletions subfolder and are visible to the user when using the Recover Deleted Items feature in Outlook and Outlook Web App. By default, items reside in this folder until the deleted item retention period configured for the mailbox database or the mailbox expires.
2. **Purges** When a user deletes an item from the Recoverable Items folder (by using the Recover Deleted Items tool in Outlook and Outlook Web App, the item is moved to the Purges folder. Items that exceed the deleted item retention period configured on the mailbox database or the mailbox are also moved to the Purges folder. Items in this folder aren't visible to users if they use the Recover Deleted Items tool. When the mailbox assistant processes the mailbox, items in the Purges folder are purged from the mailbox database. When you place the mailbox user on litigation hold, the mailbox assistant doesn't purge items in this folder.
3. **DiscoveryHold** If a user is placed on an In-Place Hold, deleted items are moved to this folder. When the mailbox assistant processes the mailbox, it evaluates messages in this folder. Items

matching the In-Place Hold query are retained until the hold period specified in the query. If no hold period is specified, items are held indefinitely or until the user is removed from the hold.

4. **Versions** When a user placed on In-Place Hold or litigation hold, mailbox items must be protected from tampering or modification by the user or a process. This is accomplished using a *copy-on-write* process. When a user or a process changes specific properties of a mailbox item, a copy of the original item is saved in the Versions folder before the change is committed. The process is repeated for subsequent changes. Items captured in the Versions folder are also indexed and returned in In-Place eDiscovery searches. After the hold is removed, copies in the Versions folder are removed by the Managed Folder Assistant.

Properties that trigger copy-on-write

Item type	Properties that trigger copy-on-write
Messages (IPM.Note*) Posts (IPM.Post*)	• Subject • Body • Attachments • Senders/Recipients • Sent/Received Dates
Items other than messages and posts	Any change to a visible property, except the following: • Item location (when an item is moved between folders) • Item status change (read or unread) • Changes to retention tag applied to an item
Items in the default folder Drafts	None (items in the Drafts folder are exempt from copy on write)

Important:
Copy-on-write is disabled for calendar items in the organizer's mailbox when meeting responses are received from attendees and the tracking information for the meeting is updated. For calendar items and items that have a reminder set, copy-on-write is disabled for the ReminderTime and ReminderSignalTime properties. Changes to these properties are not captured by copy-on-write. Changes to RSS feeds aren't captured by copy-on-write.

Although the DiscoveryHold, Purges, and Versions folders aren't visible to the user, all items in the Recoverable Items folder are indexed by Exchange Search and are discoverable using In-Place eDiscovery. After a mailbox user is removed from In-Place Hold or litigation hold, items in the DiscoveryHold, Purges, and Versions folders are purged by the Managed Folder Assistant.

In-Place Hold and mailbox quotas

Items in the Recoverable Items folder aren't calculated toward the user's mailbox quota. In Exchange 2013, the Recoverable Items folder has its own quota. When a user's Recoverable Items folder exceeds the warning quota for recoverable items (as specified by the *RecoverableItemsWarningQuota* parameter), an event is logged in the Application event log of the Mailbox server. When the folder exceeds the quota for recoverable items (as specified by the *RecoverableItemsQuota* parameter), users won't be able to empty the Deleted Items folder or permanently delete mailbox items. Also copy-on-write won't be able to create copies of modified items. Therefore, it's critical that you monitor Recoverable Items quotas for mailbox users placed on In-Place Hold.

For mailbox databases, the default *RecoverableItemsWarningQuota* and *RecoverableItemsQuota* values are set to 20 Gb and 30 Gb respectively. These settings are usually sufficient for storing several years of mailbox data when on In-Place Hold. To modify these values for a mailbox database, use the Set-MailboxDatabase cmdlet. To modify them for individual mailboxes, use the Set-Mailbox cmdlet.

Note:

In Exchange Online, it’s possible to reach or exceed the 30 GB quota for the Recoverable Items folder for a mailbox placed on In-Place Hold. If this happens, you can contact Office 365 support to request an increase of the Recoverable Items quota for a mailbox on In-Place Hold.

In-Place Hold and litigation hold

Litigation hold, the hold feature introduced in Exchange 2010 to preserve data for eDiscovery, is still available in Exchange 2013. Litigation hold uses the **LitigationHoldEnabled** property of a mailbox. Whereas In-Place Hold provides granular hold capability based on query parameters, hold period, and the ability to place multiple holds, litigation hold only allows you to place all items on hold indefinitely or until hold is removed.

In Exchange Online, you can also specify the litigation hold duration for a mailbox.

When an item is placed on one or more In-Place Holds and litigation hold at the same time, all items are held indefinitely or until the holds are removed. If you remove litigation hold and the user is still placed on one or more In-Place Holds, items matching the In-Place Hold criteria are held for the period specified in the hold settings. When you move a mailbox that’s on litigation hold in Exchange 2010 to an Exchange 2013 Mailbox server, the litigation hold setting continues to apply, ensuring that compliance requirements are met during and after the move.

Preserving archived Lync content

Exchange 2013, Microsoft Lync 2013 and Microsoft SharePoint 2013 provide an integrated preservation and eDiscovery experience that allows you to preserve and search for items across the

different data stores. Exchange 2013 allows you to archive Lync Server 2013 content in Exchange, removing the requirement of having a separate SQL Server database to store archived Lync content. The integrated hold and eDiscovery capability in SharePoint 2013 allows you to preserve and search data across all stores from a single console.

When you place an Exchange 2013 mailbox on In-Place Hold or litigation hold, Microsoft Lync 2013 content (such as instant messaging conversations and files shared in an online meeting) are archived in the mailbox. If you search the mailbox using the eDiscovery Center in Microsoft SharePoint 2013 or In-Place eDiscovery in Exchange 2013, any archived Lync content matching the search query is also returned in search results. You can also restrict the search to Lync content archived in the mailbox.

To enable archiving of Lync content in Exchange 2013 mailbox, you must configure Lync 2013 integration with Exchange 2013. For details, see the following topics:

- Planning for Archiving
- Deploying Archiving

Create or remove an In-Place Hold

Exchange Server 2013 > Messaging policy and compliance > In-Place Hold >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-26*

An In-Place Hold preserves all mailbox content, including deleted items and original versions of modified items. All such mailbox items are returned in an In-Place eDiscovery search.

Note:

Depending on your Active Directory topology and replication latency, it may take up to an hour for an In-Place Hold to take effect.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place Hold" entry in the Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create an In-Place Hold

Use the EAC to create an In-Place Hold

1. Navigate to **Compliance management** > **In-place eDiscovery & hold**.
2. Click **New +**.
3. In **In-Place eDiscovery & Hold**, on the **Name and description** page, type a name for the search and an optional description, and then click **Next**.
4. On the **Mailboxes** page, click **Specify mailboxes to search**, click **Add +**, select the mailboxes you want to place on hold, and then click **Next**.

Important:

You can use the **Search all mailboxes** option for In-Place eDiscovery searches. You can't select this option to place all mailboxes on hold. To create an In-Place Hold, you must select the specific mailboxes you want to be placed on hold.

5. On the **Search query** page, complete the following fields, and then click **Next**:
   - **Include all user mailbox content** Click this button to place all content in selected mailboxes on hold.

Note:

To get the same results as litigation hold in Exchange 2010, you can use this option and select **Hold indefinitely** on the next page (**In-Place Hold settings**).

   - **Filter based on criteria** Click this button to specify search criteria, including keywords, start and end dates, sender and recipient addresses, and message types.

Important:

If a mailbox is placed on more than five query-based holds, the query parameters are ignored and all mailbox content is placed on hold.

6. On the **In-Place Hold settings** page, select the **Place content matching the search query in selected mailboxes on hold** check box and then select one of the following options:
   - **Hold indefinitely** Click this button to place items returned by the search on an indefinite hold. Items on hold will be preserved until you remove the mailbox from the search or remove the search.
   - **Specify number of days to hold items relative to their received date** Click this button to hold items for a specific period. For example, you can use this option if your organization requires that all messages be retained for at least seven years. You can use a *time-based* In-Place Hold along with a retention policy to make sure items are deleted in seven years. To learn more about retention polices, see Retention tags and retention policies.

Use the Shell to create an In-Place Hold

This example creates an In-Place Hold named Hold-CaseId012 and adds the mailbox joe@contoso.com to the hold.

Important:

If you don't specify additional search parameters for an In-Place Hold, all items in the specified source mailboxes are placed on hold. If you don't specify the *ItemHoldPeriod* parameter, items are placed on hold indefinitely or until the mailbox is either removed from hold or the hold is deleted.

```
New-MailboxSearch "Hold-CaseId012" -SourceMailboxes
"joe@contoso.com" -InPlaceHoldEnabled $true
```

For detailed syntax and parameter information, see New-MailboxSearch.

How do you know this worked?

To verify that you have successfully created the In-Place Hold, do one of the following:

- Use the EAC to verify that the In-Place Hold is listed in the list view of the **In-place eDiscovery & hold** tab.
- Use the **Get-MailboxSearch** cmdlet to retrieve the mailbox search and check the search parameters. For an example of how to retrieve a mailbox search, see the examples in Get-MailboxSearch.

Remove an In-Place Hold

Important:

In Exchange 2013, mailbox searches can be used for an In-Place Hold and In-Place eDiscovery. You can't remove a mailbox search that's used for In-Place Hold. You must first disable the In-Place Hold by clearing the **Place content matching the search query in selected mailboxes on hold** check box on the **In-Place Hold settings** page or by setting the *InPlaceHoldEnabled* parameter to `$false` in the Shell. You can also remove a mailbox by using the *SourceMailboxes* parameter specified in the search.

Use the EAC to remove an In-Place Hold

1. Navigate to **Compliance management** > **In-Place eDiscovery & hold**.
2. In the list view, select the In-Place Hold you want to remove and then click **Edit** .
3. In **In-Place eDiscovery & Hold** properties, on the **In-Place Hold** page, clear the **Place content matching the search query in selected mailboxes on hold**, and then click **Save**.
4. Select the In-Place Hold again from the list view, and then click **Delete** .
5. In warning, click **Yes** to remove the search.

Use the Shell to remove an In-Place Hold

This example first disables In-Place Hold named Hold-CaseId012 and then removes the mailbox search.

```
Set-MailboxSearch "Hold-CaseId012" -InPlaceHoldEnabled
$false
Remove-MailboxSearch "Hold-CaseId012"
```

For detailed syntax and parameter information, see Set-MailboxSearch.

How do you know this worked?

To verify that you have successfully removed an In-Place Hold, do one of the following:

- Use the EAC to verify that the In-Place Hold doesn't appear in the list view of the **In-place eDiscovery & hold** tab.
- Use the **Get-MailboxSearch** cmdlet to retrieve all mailbox searches and check that the search you removed is no longer listed. For an example of how to retrieve a mailbox search, see the examples in Get-MailboxSearch.

Place a mailbox on Litigation Hold

Exchange Server 2013 > Messaging policy and compliance > In-Place Hold >

Topic Last Modified: *2014-06-05*

Place a mailbox on Litigation Hold to preserve all mailbox content, including deleted items and original versions of modified items. Deleted and modified items are preserved for a specified period, or until you remove the mailbox from Litigation Hold. All such mailbox items are returned in an In-Place eDiscovery search.

Caution:
Litigation Hold preserves items in the Recoverable Items folder in the user's mailbox. Depending on number and size of items deleted or modified, the size of the Recoverable Items folder of the mailbox may increase quickly. The Recoverable Items folder is configured with a high quota by default. We recommend that you monitor mailboxes that are placed on Litigation Hold on a weekly basis to ensure they do not reach the Recoverable Items quotas.

Important:
Litigation Hold setting may take up to 60 minutes to take effect.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place Hold" entry in the Messaging policy and compliance permissions topic.
- Litigation Hold preserves deleted items and also preserves original versions of modified items until the hold is removed. You can optionally specify a hold duration, which preserves a mailbox item for the specified duration. To preserve items that meet your specified criteria, use an In-Place Hold to create a *query-based* hold.
- You can't use the Exchange admin center (EAC) to specify the litigation hold duration option.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see

Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to place a mailbox on Litigation Hold

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to configure message delivery restrictions for, and then click **Edit**.
3. On the mailbox properties page, click **Mailbox Features.**
4. Under **Litigation Hold**, click **Enable** to place the mailbox on Litigation Hold.
5. On the Litigation Hold page, enter the following optional information:
   - **Note** Use this field to inform the user their mailbox is on Litigation Hold. The note will appear in the user's mailbox if they're using Outlook 2010 or later.
   - **URL** Use this field to direct the user to a website for more information about Liigation Hold. This URL appears in the user's mailbox if they are using Outlook 2010 or later.

Use the Shell to place a mailbox on Litigation Hold

This example places the mailbox bsuneja@contoso.com on Litigation Hold.

```
Set-Mailbox bsuneja@contoso.com -LitigationHoldEnabled
$true
```

Use the Shell to place a mailbox on Litigation Hold and preserve items for a specified duration

This examples places the mailbox bsuneja@contoso.com on Litigation Hold and preserves items for 2555 days (approximately 7 years).

```
Set-Mailbox bsuneja@contoso.com -LitigationHoldEnabled
$true -LitigationHoldDuration 2555
```

Use the Shell to place all mailboxes on Litigation Hold for a specified duration

Your organization may require that all mailbox data be preserved for a specified duration. Before you place all mailboxes in an organization on Litigation Hold, consider the following:

- When you use the command to place all mailboxes in an organization, or a subset of mailboxes

matching a specified filter on Litigation Hold, only mailboxes that exist at the time you run the command are placed on hold. If you create new mailboxes later, you must run the command again to place the new mailboxs on hold. If you frequently create new mailboxes, you can run the command as a scheduled task as frequently as required..

- Placing all mailboxes in an organization on hold can significantly impact mailbox sizes. In on-premises deployments, plan for adequate storage to meet your organization's preservation requirements.
- Preserving mailbox data for a long duration will result in growth of the Recoverable Items folder in a user's mailbox and archive. The Recoverable Items folder has its own storage limit, so items in the folder don't count towards the mailbox storage limit For most users, the default storage limit is sufficient for storing several years' worth of messages. We recommend monitoring the size of the folder periodically to ensure it doesn't reach the limit. See Recoverable Items folder for more details

This examples places all the mailboxes in an organization on Litigation Hold for 2555 days (approximately 7 years). The hold is not applied to the Discovery mailboxes.

```
Get-Mailbox -ResultSize Unlimited -Filter
{RecipientTypeDetails -ne "DiscoveryMailbox"} | Set-Mailbox
-LitigationHoldEnabled $true -LitigationHoldDuration 2555
```

The example uses the Get-Mailbox cmdlet to retrieve all mailboxes in the organization, specifies a recipient filter to exclude the Discovery mailboxes, and pipes the mailboxes to the Set-Mailbox cmdlet to enable Litigation Hold. You can use other user properties in a filter to exclude or include one or more mailboxes. For details, see Filterable properties for the -Filter parameter.

Use the Shell to remove a mailbox from Litigation Hold

This example removes the mailbox bsuneja@contoso.com from Litigation Hold.

```
Set-Mailbox bsuneja@contoso.com -LitigationHoldEnabled
$false
```

How do you know this worked?

To verify that you have successfully placed a mailbox on Litigation Hold, do the following:

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to verify Litigation Hold settings for, and then click **Edit**.
3. On the mailbox properties page, click **Mailbox Features.**
4. Under **Litigation Hold**, verify that hold is enabled.
5. Click View details to verify when the mailbox was placed on Litigation Hold and by whom. You can also verify or change the values in the optional **Note** and **URL** fields.

Place all mailboxes on hold

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-15*

Your organization may require all mailbox data to be preserved for a specific period. You can use Litigation Hold or In-Place Hold to meet this requirement. After you place a mailbox on Litigation Hold or In-Place Hold, mailbox items attempted to be permanently deleted or modified are preserved in the Recoverable Items folder. For more details, see In-Place Hold.

Before you place all mailboxes in an organization on Litigation Hold or In-Place Hold, consider the following:

- Placing all mailboxes in an organization on hold can significantly impact mailbox sizes. In on-premises deployments, plan for adequate storage to meet your organization's preservation requirements. In Exchange Online, mailbox storage limits for your users depend on the user's subscription license.
- Preserving mailbox data for a long duration will result in growth of the Recoverable Items folder in a user's mailbox and archive. The Recoverable Items folder has its own storage limit, so items in the folder don't count towards the mailbox storage limit. For most users, the default storage limit is sufficient for storing several years' worth of messages. We recommend monitoring the size of the folder periodically to ensure it doesn't reach the limit. See Recoverable Items folder for more details.

Choosing between Litigation Hold and In-Place Hold to place all mailboxes on hold

Here are some factors you should consider in deciding the hold feature you should use to place all mailboxes on hold.

You want to...	Use Litigation Hold	Use In-Place Hold
Use the EAC	Yes. For setting Litigation Hold, EAC is best suited for quick one-off actions on a few mailboxes. We recommend using the Shell for setting Litigation Hold for all users in an organization. You	Yes. However, you can't select more than 500 source mailboxes in the EAC.

	can't specify the litigation hold duration in the EAC.	
Use the Shell	Yes	Yes
Place more than 10,000 mailboxes on hold	Yes. Litigation Hold is a mailbox property. You can place all mailboxes in an organization on hold using a Shell command.	Yes, using multiple In-Place Holds. You can specify a maximum of 10,000 mailboxes in a single In-Place Hold. To place additional mailboxes on hold, you must create additional In-Place Holds. This will result in additional management overhead. Using Litigation Hold is simpler.
Place many different mailboxes on hold for different periods.	Yes Litigation Hold is a mailbox property. You can place each mailbox (or sets of mailboxes) on hold for a different duration.	Yes If you're placing individual holds on thousands of mailboxes, we recommend using Litigation Hold. If you're creating holds for specific events that involve multiple users, use a single in-Place hold for the group of users. It's not recommended to create separate in-place holds for each mailbox as this will create many in-place hold queries that will be more difficult to manage than litigation holds. A large number of In-Place Hold objects may also result in slow

		performance in the EAC when refreshing, creating or modifying hold objects.
Automatically place new mailboxes on hold	No You must place a new mailbox on litigation hold after it's created. You can also schedule the command or script to run as frequently as required to achieve the same effect.	No You must add a new mailbox to an In-Place Hold. You can also schedule the command or script to run as frequently as required to achieve the same effect. We recommend that the script check if an existing In-Place Hold has already reached the 10,000 mailbox limit and creating a new In-Place Hold if required.

Place all mailboxes on Litigation Hold

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place Hold" entry in the Messaging policy and compliance permissions topic.

You can easily and quickly place all mailboxes on hold indefinitely or for a specified duration using the Shelll. This command places all mailboxes on hold for 7 years. The hold is not applied to Discovery mailboxes.

```
Get-Mailbox -ResultSize Unlimited -Filter
{RecipientTypeDetails -ne "DiscoveryMailbox"} | Set-Mailbox
-LitigationHoldEnabled $true -LitigationHoldDuration 2555
```

The example uses the Get-Mailbox cmdlet to retrieve all mailboxes in your organization, specifies a recipient filter to exclude Discovery mailboxes and pipes the mailboxes to the Set-Mailbox cmdlet to enable Litigation Hold. You can use other user properties in a filter to exclude or include one or more mailboxes. For a list of properties you can use in a filter, see Filterable properties for the -Filter parameter.

Place all mailboxes on In-Place Hold

You can use the EAC to select up to 500 mailboxes and place them on hold. For details, see Create or remove an In-Place Hold.

To place more than 500 users on In-Place Hold, use the Shell.

More Info

- Only mailboxes that exist at the time you run the command are placed on hold. If you create new mailboxes later, run the command again to place them on hold. If you frequently create new mailboxes, you can run the command as a scheduled task as frequently as required.
- Placing mailboxes on hold preserves data by preventing deletion before the specified period and saving the original version of a message before it's modified. It does not automatically delete messages after the specified period. Combine Litigation Hold or In-Place Hold with a Retention Policy, which can automatically delete messages after the specified period, to meet your organization's email retention requirements. See Retention tags and retention policies for details.

In-Place eDiscovery

Exchange Server 2013 > Messaging policy and compliance >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-26*

If your organization adheres to legal discovery requirements (related to organizational policy, compliance, or lawsuits), In-Place eDiscovery in Microsoft Exchange Server 2013 and Exchange Online can help you perform discovery searches for relevant content within mailboxes. Exchange 2013 and Exchange Online also offer federated search capability and integration with Microsoft SharePoint 2013 and Microsoft SharePoint Online. Using the eDiscovery Center in SharePoint, you can search for and hold all content related to a case, including SharePoint 2013 and SharePoint Online websites, documents, file shares indexed by SharePoint (SharePoint 2013 only), mailbox content in Exchange, and archived Lync 2013 content. You can also use In-Place eDiscovery in an Exchange hybrid environment to search on-premises and cloud-based mailboxes in the same search.

Important:
In-Place eDiscovery is a powerful feature that allows a user with the correct permissions to potentially gain access to all messaging records stored throughout the Exchange 2013 or Exchange Online organization. It's important to control and monitor discovery activities, including addition of members to the Discovery Management role group, assignment of the Mailbox Search management role, and assignment of mailbox access permission to discovery mailboxes.

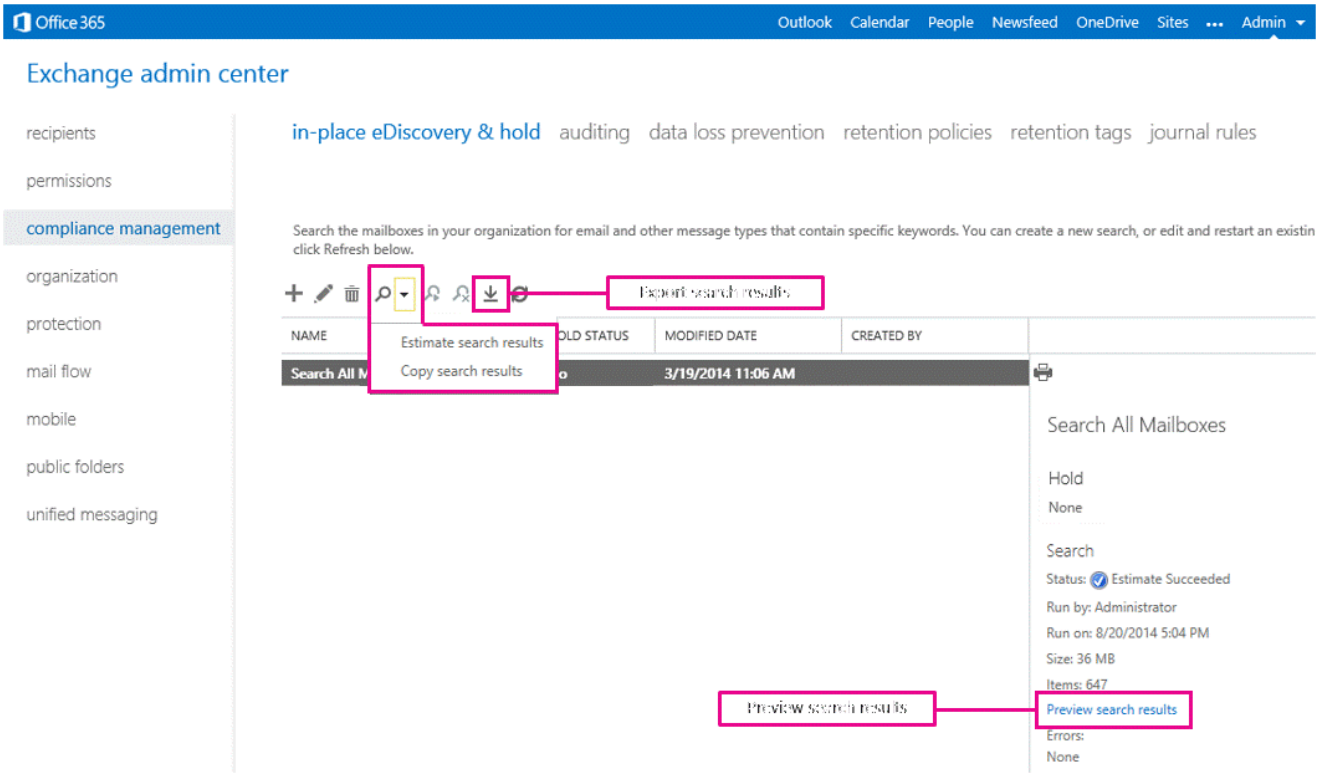
Contents

How In-Place eDiscovery works

In-Place eDiscovery uses the content indexes created by Exchange Search. Role Based Access Control (RBAC) provides the Discovery Management role group to delegate discovery tasks to non-technical personnel, without the need to provide elevated privileges that may allow a user to make any operational changes to Exchange configuration. The Exchange admin center (EAC) provides an easy-to-use search interface for non-technical personnel such as legal and compliance officers, records managers, and human resources (HR) professionals.

Authorized users can perform an In-Place eDiscovery search by selecting the mailboxes, and then specifying search criteria such as keywords, start and end dates, sender and recipient addresses, and message types. After the search is complete, authorized users can then select one of the following actions:

- **Estimate search results** This option returns an estimate of the total size and number of items that will be returned by the search based on the criteria you specified.
- **Preview search results** This option provides a preview of the results. Messages returned from each mailbox searched are displayed.
- **Copy search results** This option lets you copy messages to a discovery mailbox.
- **Export search results** After search results are copied to a discovery mailbox, you can export them to a PST file.



Exchange Search

In-Place eDiscovery uses the content indexes created by Exchange Search. Exchange Search has been retooled to use Microsoft Search Foundation, a rich search platform that comes with significantly improved indexing and querying performance and improved search functionality. Because the Microsoft Search Foundation is also used by other Office products, including SharePoint 2013, it offers greater interoperability and similar query syntax across these products.

With a single content indexing engine, no additional resources are used to crawl and index mailbox databases for In-Place eDiscovery when eDiscovery requests are received by IT departments.

In-Place eDiscovery uses Keyword Query Language (KQL), a querying syntax similar to the Advanced Query Syntax (AQS) used by Instant Search in Microsoft Outlook and Outlook Web App. Users familiar with KQL can easily construct powerful search queries to search content indexes.

For more information about the file formats indexed by Exchange search, see File formats indexed by Exchange Search.

Discovery Management role group and management roles

For authorized users to perform In-Place eDiscovery searches, you must add them to the Discovery Management role group. This role group consists of two management roles: the Mailbox Search role, which allows a user to perform an In-Place eDiscovery search, and the Legal Hold role, which allows a user to place a mailbox on In-Place Hold or litigation hold.

By default, permissions to perform In-Place eDiscovery-related tasks aren't assigned to any user or Exchange administrators. Exchange administrators who are members of the Organization Management role group can add users to the Discovery Management role group and create

custom role groups to narrow the scope of a discovery manager to a subset of users. To learn more about adding users to the Discovery Management role group, see Add a user to the Discovery Management role group.

Important:
If a user hasn't been added to the Discovery Management role group or isn't assigned the Mailbox Search role, the **In-Place eDiscovery & Hold** user interface isn't displayed in the EAC, and the In-Place eDiscovery cmdlets aren't available in the Exchange Management Shell.

Auditing of RBAC role changes, which is enabled by default, makes sure that adequate records are kept to track assignment of the Discovery Management role group. You can use the administrator role group report to search for changes made to administrator role groups. For more information, see Search the role group changes or administrator audit logs.

Return to top

Custom management scopes for In-Place eDiscovery

You can use a custom management scope to let specific people or groups use In-Place eDiscovery to search a subset of mailboxes in your Exchange 2013 or Exchange Online organization. For example, you might want to let a discovery manager search only the mailboxes of users in a specific location or department. You do this by creating a custom management scope that uses a custom recipient filter to control which mailboxes can be searched. Recipient filter scopes use filters to target specific recipients based on recipient type or other recipient properties.

For In-Place eDiscovery, the only property on a user mailbox that you can use to create a recipient filter for a custom scope is distribution group membership. If you use other properties, such as *CustomAttributeN*, *Department*, or *PostalCode*, the search fails when it's run by a member of the role group that's assigned the custom scope. For more information, see Create a custom management scope for In-Place eDiscovery searches.

Integration with SharePoint Server 2013 and SharePoint Online

Exchange 2013 and Exchange Online offer integration with SharePoint 2013 and SharePoint Online, allowing a discovery manager to use eDiscovery Center in SharePoint to perform the following tasks:

- **Search and preserve content from a single location** An authorized discovery manager can search and preserve content across SharePoint and Exchange, including Lync content such as instant messaging conversations and shared meeting documents archived in Exchange mailboxes.
- **Case management** eDiscovery Center uses a case management approach to eDiscovery, allowing you to create cases and search and preserve content across different content repositories for each case.

- **Export search results** A discovery manager can use eDiscovery Center to export search results. Mailbox content included in search results is exported to a PST file.

SharePoint also uses Microsoft Search Foundation for content indexing and querying. Regardless of whether a discovery manager uses the EAC or the eDiscovery Center to search Exchange content, the same mailbox content is returned.

In on-premises deployments, before you can use eDiscovery Center in SharePoint to search Exchange mailboxes, you must establish trust between the two applications. In Exchange 2013 and SharePoint 2013, this is done using OAuth authentication. For details, see Configure Exchange for SharePoint eDiscovery Center. eDiscovery searches performed from SharePoint are authorized by Exchange using RBAC. For a SharePoint user to be able to perform an eDiscovery search of Exchange mailboxes, they must be assigned delegated Discovery Management permission in Exchange. To be able to preview mailbox content returned in an eDiscovery search performed using SharePoint eDiscovery Center, the discovery manager must have a mailbox in the same Exchange organization.

For step-by step instructions for setting up an eDiscovery Center in an Office 365 organization, see Set up an eDiscovery Center in SharePoint Online.

eDiscovery in an Exchange hybrid deployment

To successfully perform cross-premises eDiscovery searches in an Exchange 2013 hybrid organization, you will have to configure OAuth (Open Authorization) authentication between your Exchange on-premises and Exchange Online organizations so that you can use In-Place eDiscovery to search on-premises and cloud-based mailboxes. OAuth authentication is a server-to-server authentication protocol that allows applications to authenticate to each other.

OAuth authentication supports the following eDiscovery scenarios in an Exchange hybrid deployment:

- Search on-premises mailboxes that use Exchange Online Archiving for cloud-based archive mailboxes.
- Search on-premises and cloud-based mailboxes in the same eDiscovery search.
- Search on-premises mailboxes by using the eDiscovery Center in SharePoint Online.

For more information about the eDiscovery scenarios that require OAuth authentication to be configured in an Exchange hybrid deployment, see Using OAuth authentication to support eDiscovery in an Exchange hybrid deployment. For step-by-step instructions for configuring OAuth authentication to support eDiscovery, see Configure OAuth authentication between Exchange and Exchange Online organizations.

Discovery mailboxes

After you create an In-Place eDiscovery search, you can copy the search results to a target mailbox. The EAC allows you to select a discovery mailbox as the target mailbox. A discovery mailbox is a

special type of mailbox that provides the following functionality:

- **Easier and secure target mailbox selection** When you use the EAC to copy In-Place eDiscovery search results, only discovery mailboxes are made available as a repository in which to store search results. You don't need to sort through a potentially long list of mailboxes available in the organization. This also eliminates the possibility of a discovery manager accidentally selecting another user's mailbox or an unsecured mailbox in which to store potentially sensitive messages.
- **Large mailbox storage quota** The target mailbox should be able to store a large amount of message data that may be returned by an In-Place eDiscovery search. By default, discovery mailboxes have a mailbox storage quota of 50 gigabytes (GB). This storage quota can’t be increased.
- **More secure by default** Like all mailbox types, a discovery mailbox has an associated Active Directory user account. However, this account is disabled by default. Only users explicitly authorized to access a discovery mailbox have access to it. Members of the Discovery Management role group are assigned Full Access permissions to the default discovery mailbox. Any additional discovery mailboxes you create don't have mailbox access permissions assigned to any user.
- **Email delivery disabled** Although visible in Exchange address lists, users can't send email to a discovery mailbox. Email delivery to discovery mailboxes is prohibited by using delivery restrictions. This preserves the integrity of search results copied to a discovery mailbox.

Exchange 2013 Setup creates one discovery mailbox with the display name **Discovery Search Mailbox**. You can use the Shell to create additional discovery mailboxes. By default, the discovery mailboxes you create won't have any mailbox access permissions assigned. You can assign Full Access permissions for a discovery manager to access messages copied to a discovery mailbox. For details, see Create a discovery mailbox.

In-Place eDiscovery also uses a system mailbox with the display name **SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9}** to hold In-Place eDiscovery metadata. System mailboxes aren't visible in the EAC or in Exchange address lists. In on-premises organizations, before removing a mailbox database where the In-Place eDiscovery system mailbox is located, you must move the mailbox to another mailbox database. If the mailbox is removed or corrupted, your discovery managers are unable to perform eDiscovery searches until you re-create the mailbox. For details, see Re-create the Discovery system mailbox.

Return to top

Using In-Place eDiscovery

Users who have been added to the Discovery Management role group can perform In-Place eDiscovery searches. You can perform a search using the web-based interface in the EAC. This makes it easier for non-technical users such as records managers, compliance officers, or legal and HR professionals to use In-Place eDiscovery. You can also use the Shell to perform a search. For more information, see Create an In-Place eDiscovery search

Note:

In on-premises organizations, you can use In-Place eDiscovery to search mailboxes located on Exchange 2013 Mailbox servers. To search mailboxes located on Exchange 2010 Mailbox servers, use Multi-Mailbox Search on an Exchange 2010 server.
In a hybrid deployment, which is an environment where some mailboxes exist on your on-premises Mailbox servers and some mailboxes exist in a cloud-based organization, you can perform In-Place eDiscovery searches of your cloud-based mailboxes using the EAC in your on-premises organization. If you intend to copy messages to a discovery mailbox, you must select an on-premises discovery mailbox. Messages from cloud-based mailboxes that are returned in search results are copied to the specified on-premises discovery mailbox. To learn more about hybrid deployments, see **Exchange Server 2013 Hybrid Deployments**.

The **In-Place eDiscovery & Hold** wizard in the EAC allows you to create an In-Place eDiscovery search and also use In-Place Hold to place search results on hold. When you create an In-Place eDiscovery search, a search object is created in the In-Place eDiscovery system mailbox. This object can be manipulated to start, stop, modify, and remove the search. After you create the search, you can choose to get an estimate of search results, which includes keyword statistics that help you determine query effectiveness. You can also do a live preview of items returned in the search, allowing you to view message content, the number of messages returned from each source mailbox and the total number of messages. You can use this information to further fine-tune your query if required.

When satisfied with the search results, you can copy them to a discovery mailbox. You can also use the EAC or Outlook to export a discovery mailbox or some of its content to a PST file.

When creating an In-Place eDiscovery search, you must specify the following parameters:

- **Name** The search name is used to identify the search. When you copy search results to a discovery mailbox, a folder is created in the discovery mailbox using the search name and the timestamp to uniquely identify search results in a discovery mailbox.
- **Mailboxes** You can choose to search all mailboxes in your Exchange 2013 organization or specify the mailboxes to search. If you also want to use the same search to place items on hold, you must specify the mailboxes. You can specify a distribution group to include mailbox users who are members of that group. Membership of the group is calculated once when creating the search and subsequent changes to group membership are not automatically reflected in the search. A user's primary and archive mailboxes are included in the search.
- **Search query** You can either include all mailbox content from the specified mailboxes or use a search query to return items that are more relevant to the case or investigation. You can specify the following parameters in a search query:
  - **Keywords** You can specify keywords and phrases to search message content. You can also use the logical operators **AND**, **OR**, and **NOT**. Additionally, Exchange 2013 also supports the **NEAR** operator, allowing you to search for a word or phrase that's in proximity to another word or phrase.

To search for an exact match of a multiple word phrase, you must enclose the phrase in quotation marks. For example, searching for the phrase “plan and competition” returns messages that contain an exact match of the phrase, whereas specifying **plan AND competition** returns messages that

contain the words **plan** and **competition** anywhere in the message.

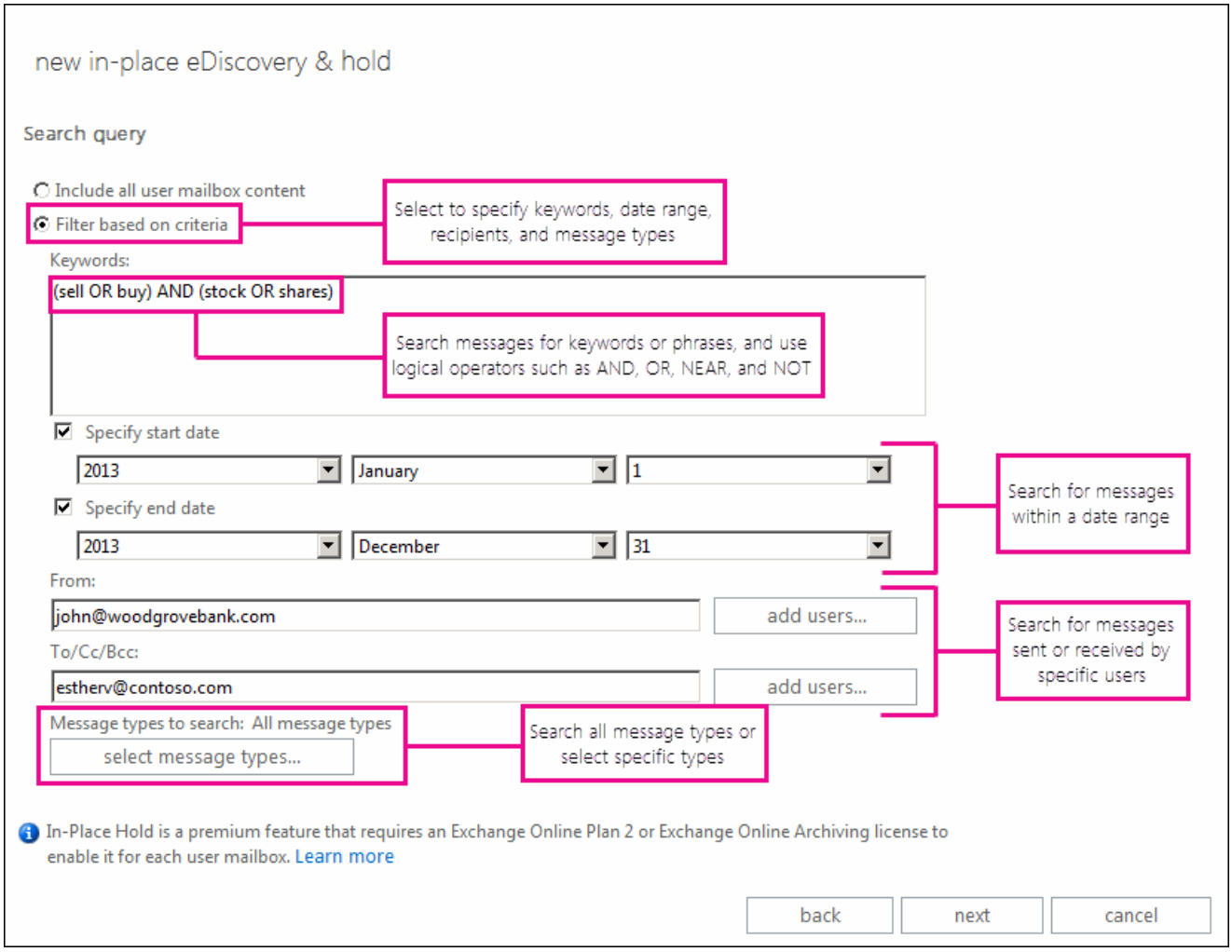
Exchange 2013 also supports the Keyword Query Language (KQL) syntax for In-Place eDiscovery searches.

Note:
In-Place eDiscovery does not support regular expressions.

You must capitalize logical operators such as **AND** and **OR** for them to be treated as operators instead of keywords. We recommend that you use explicit parenthesis for any query that mixes multiple logical operators to avoid mistakes or misinterpretations. For example, if you want to search for messages that contain either WordA or WordB AND either WordC or WordD, you must use **(WordA OR WordB) AND (WordC OR WordD)**.

- **Start and End dates** By default, In-Place eDiscovery doesn't limit searches by a date range. To search messages sent during a specific date range, you can narrow the search by specifying the start and end dates. If you don't specify an end date, the search will return the latest results every time you restart it.
- **Senders and recipients** To narrow down the search, you can specify the senders or recipients of messages. You can use email addresses, display names, or the name of a domain to search for items sent to or from everyone in the domain. For example, to find email sent by or sent to anyone at Contoso, Ltd, specify **@contoso.com** in the **From** or the **To/cc** field in the EAC. You can also specify **@contoso.com** in the *Senders* or *Recipients* parameters in the Shell.
- **Message types** By default, all message types are searched. You can restrict the search by selecting specific message types such as email, contacts, documents, journal, meetings, notes and Lync content.

The following screenshot shows an example of a search query in the EAC.



When using In-Place eDiscovery, also consider the following:

- **Attachments** In-Place eDiscovery searches attachments supported by Exchange Search. For details, see File formats indexed by Exchange Search. In on-premises deployments, you can add support for additional file types by installing search filters (also known as an iFilter) for the file type on Mailbox servers.
- **Unsearchable items** Unsearchable items are mailbox items that can't be indexed by Exchange Search. Reasons they can't be indexed include the lack of an installed search filter for an attached file, a filter error, and encrypted messages. For a successful eDiscovery search, your organization may be required to include such items for review. When copying search results to a discovery mailbox or exporting them to a PST file, you can include unsearchable items. For more information, see Unsearchable items in Exchange eDiscovery.
- **Encrypted items** Because messages encrypted using S/MIME aren't indexed by Exchange Search, In-Place eDiscovery doesn't search these messages. If you select the option to include unsearchable items in search results, these S/MIME encrypted messages are copied to the discovery mailbox.
- **IRM-protected items** Messages protected using Information Rights Management (IRM) are indexed by Exchange Search and therefore included in the search results if they match query parameters. Messages must be protected by using an Active Directory Rights Management Services (AD RMS) cluster in the same Active Directory forest as the Mailbox server. For more information, see Information Rights Management.

Important:

When Exchange Search fails to index an IRM-protected message, either due to a decryption failure or because IRM is disabled, the protected message isn't added to the list of failed items. If you select the option to include unsearchable items in search results, the results may not include IRM-protected messages that could not be decrypted.

To include IRM-protected messages in a search, you can create another search to include messages with .rpmsg attachments. You can use the query string `attachment:rpmsg` to search all IRM-protected messages in the specified mailboxes, whether successfully indexed or not. This may result in some duplication of search results in scenarios where one search returns messages that match the search criteria, including IRM-protected messages that have been indexed successfully. The search doesn't return IRM-protected messages that couldn't be indexed.

Performing a second search for all IRM-protected messages also includes the IRM-protected messages that were successfully indexed and returned in the first search. Additionally, the IRM-protected messages returned by the second search may not match the search criteria such as keywords used for the first search.

- **De-duplication** When copying search results to a discovery mailbox, you can enable *de-duplication* of search results to copy only one instance of a unique message to the discovery mailbox. De-duplication has the following benefits:
  - Lower storage requirement and smaller discovery mailbox size due to reduced number of messages copied.
  - Reduced workload for discovery managers, legal counsel, or others involved in reviewing search results.
  - Reduced cost of eDiscovery, depending on the number of duplicate items excluded from search results.

Return to top

Estimate, preview, and copy search results

After an In-Place eDiscovery search is completed, you can view search result estimates in the Details pane in the EAC. The estimate includes number of items returned and total size of those items. You can also view keyword statistics, which returns details about number of items returned for each keyword used in the search query. This information is helpful in determining query effectiveness. If the query is too broad, it may return a much bigger data set, which could require more resources to review and raise eDiscovery costs. If the query is too narrow, it may significantly reduce the number of records returned or return no records at all. You can use the estimates and keyword statistics to fine-tune the query to meet your requirements.

Note:

In Exchange 2013, keyword statistics also include statistics for non-keyword properties such as dates, message types, and senders/recipients specified in a search query.

You can also preview the search results to further ensure that messages returned contain the content you're searching for and further fine-tune the query if required. eDiscovery Search Preview

displays the number of messages returned from each mailbox searched and the total number of messages returned by the search. The preview is generated quickly without requiring you to copy messages to a discovery mailbox.

After you're satisfied with the quantity and quality of search results, you can copy them to a discovery mailbox. When copying messages, you have the following options:

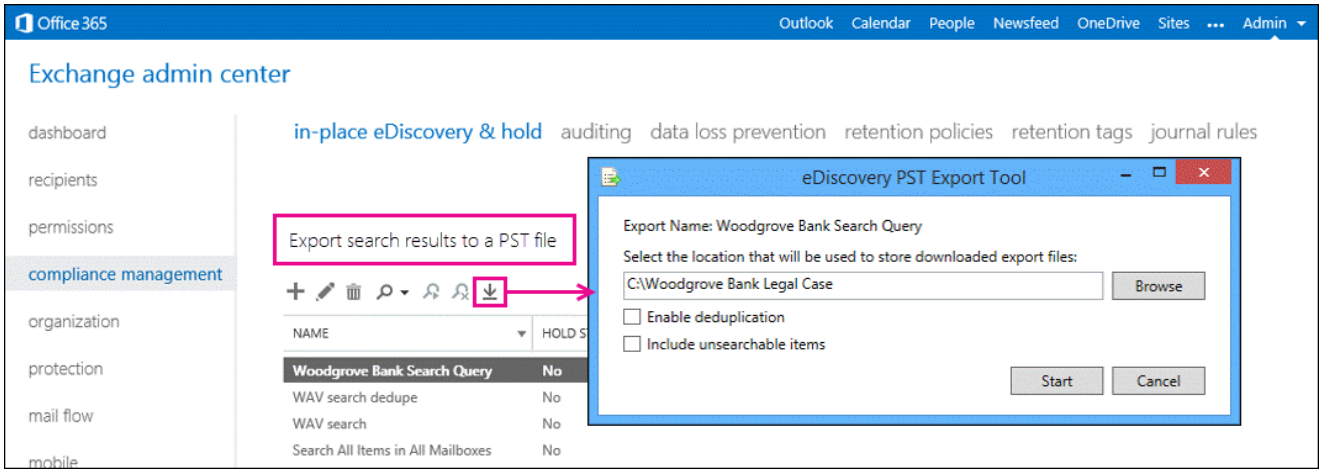
- **Include unsearchable items** For details about the types of items that are considered unsearchable, see the eDiscovery search considerations in the previous section.
- **Enable de-duplication** De-duplication reduces the dataset by only including a single instance of a unique record if multiple instances are found in one or more mailboxes searched.
- **Enable full logging** By default, only basic logging is enabled when copying items. You can select full logging to include information about all records returned by the search.
- **Send me mail when the copy is completed** An In-Place eDiscovery search can potentially return a large number of records. Copying the messages returned to a discovery mailbox can take a long time. Use this option to get an email notification when the copying process is completed. For easier access using Outlook Web App, the notification includes a link to the location in a discovery mailbox where the messages are copied.

For more information, see Copy eDiscovery search results to a discovery mailbox.

Return to top

Export search results to a PST file

After search results are copied to a discovery mailbox, you can export the search results to a PST file.



After search results are exported to a PST file, you or other users can open them in Outlook to review or print messages returned in the search results. For more information, see Export eDiscovery search results to a PST file.

Different search results

Because In-Place eDiscovery performs searches on live data, it's possible that two searches of the

same content sources and using the same search query can return different results. Estimated search results can also be different from the actual search results that are copied to a discovery mailbox. This can happen even when rerunning the same search within a short amount of time. There are several factors that can affect the consistency of search results:

- The continual indexing of incoming email because Exchange Search continuously crawls and indexes your organization's mailbox databases and transport pipeline.
- Deletion of email by users or automated processes.
- Bulk importing large amounts of email, which takes time to index.

If you do experience dissimilar results for the same search, consider placing mailboxes on hold to preserve content, running searches during off-peak hours, and allowing time for indexing after importing large amounts of email.

Logging for In-Place eDiscovery searches

There are two types of logging available for In-Place eDiscovery searches:

- **Basic logging** Basic logging is enabled by default for all In-Place eDiscovery searches. It includes information about the search and who performed it. Information captured about basic logging appears in the body of the email message sent to the mailbox where the search results are stored. The message is located in the folder created to store search results.
- **Full logging** Full logging includes information about all messages returned by the search. This information is provided in a comma-separated value (.csv) file attached to the email message that contains the basic logging information. The name of the search is used for the .csv file name. This information may be required for compliance or record-keeping purposes. To enable full logging, you must select the **Enable full logging** option when copying search results to a discovery mailbox in the EAC. If you're using the Shell, specify the full logging option using the *LogLevel* parameter.

Note:
When using the Shell to create or modify an In-Place eDiscovery search, you can also disable logging.

Besides the search log included when copying search results to a discovery mailbox, Exchange also logs cmdlets used by the EAC or the Shell to create, modify or remove In-Place eDiscovery searches. This information is logged in the admin audit log entries. For details, see Administrator audit logging.

Return to top

In-Place eDiscovery and In-Place Hold

As part of eDiscovery requests, you may be required to preserve mailbox content until a lawsuit or investigation is disposed. Messages deleted or altered by the mailbox user or any processes must also be preserved. In Exchange 2013, this is accomplished by using In-Place Hold. For details, see

In-Place Hold.

In Exchange 2013, you can use the new **In-Place eDiscovery & Hold** wizard to search items and preserve them for as long as they're required for eDiscovery or to meet other business requirements. When using the same search for both In-Place eDiscovery and In-Place Hold, be aware of the following:

- You can't use the option to search all mailboxes. You must select the mailboxes or distribution groups.
- You can't remove an In-Place eDiscovery search if the search is also used for In-Place Hold. You must first disable the In-Place Hold option in a search and then remove the search.

Preserving mailboxes for In-Place eDiscovery

When an employee leaves an organization, it's a common practice to disable or remove the mailbox. After you disable a mailbox, it is disconnected from the user account but remains in the mailbox for a certain period, 30 days by default. The Managed Folder Assistant does not process disconnected mailboxes and any retention policies are not applied during this period. You can't search content of a disconnected mailbox. Upon reaching the deleted mailbox retention period configured for the mailbox database, the mailbox is purged from the mailbox database.

Important:
In Exchange Online, In-Place eDiscovery can search content in inactive mailboxes. Inactive mailboxes are mailboxes that are placed on In-Place Hold or litigation hold and then removed. Inactive mailboxes are preserved as long as they’re placed on hold. When an inactive mailbox is removed from In-Place Hold or when litigation hold is disabled, it is permanently deleted. For details, see **Manage inactive mailboxes in Exchange Online**.

In on-premises deployments, if your organization requires that retention settings be applied to messages of employees who are no longer in the organization or if you may need to retain an ex-employee's mailbox for an ongoing or future eDiscovery search, do not disable or remove the mailbox. You can take the following steps to ensure the mailbox can't be accessed and no new messages are delivered to it.

1. Disable the Active Directory user account using **Active Directory Users & Computers** or other Active Directory or account provisioning tools or scripts. This prevents mailbox logon using the associated user account.

Important:
Users with Full Access mailbox permission will still be able to access the mailbox. To prevent access by others, you must remove their Full Access permission from the mailbox. For information about how to remove Full Access mailbox permissions on a mailbox, see Manage Permissions for Recipients.

2. Set the message size limit for messages that can be sent from or received by the mailbox user to a very low value, 1 KB for example. This prevents delivery of new mail to and from the mailbox. For details, see Configure message size limits for a mailbox.
3. Configure delivery restrictions for the mailbox so nobody can send messages to it. For details,

see Configure message delivery restrictions for a mailbox.

Important:
You must take the above steps along with any other account management processes required by your organization, but without disabling or removing the mailbox or removing the associated user account.

When planning to implement mailbox retention for messaging retention management (MRM) or In-Place eDiscovery, you must take employee turnover into consideration. Long-term retention of ex-employee mailboxes will require additional storage on Mailbox servers and also result in an increase in Active Directory database because it requires that the associated user account be retained for the same duration. Additionally, it may also require changes to your organization's account provisioning and management processes.

Return to top

In-Place eDiscovery limits and throttling policies

In Exchange 2013 and Exchange Online, the resources In-Place eDiscovery can consume are controlled using throttling policies.

The default throttling policy contains the following throttling parameters.

Parameter	Description	Default value
DiscoveryMaxConcurrency	The maximum number of In-Place eDiscovery searches a user can perform concurrently.	2
DiscoveryMaxMailboxes	The maximum number of mailboxes that can be searched in a single In-Place eDiscovery search.	Exchange Online: 10,000[1] Exchange 2013: 5,000
DiscoveryMaxStatsSearchMailboxes	The maximum number of mailboxes that can be searched in a single In-Place eDiscovery search that still allows you to view keyword statistics.	100 **Note:** After you run an eDiscovery search estimate, you can view keyword statistics. These statistics show details about the number of items returned for each keyword used in the search query. If more than 100 source mailboxes are included in the search, an error will be

		returned if you try to view keyword statistics.
DiscoveryMaxKeywords	The maximum number of keywords that can be specified in a single In-Place eDiscovery search.	500
DiscoveryMaxSearchResultsPageSize	The maximum number of items displayed on a single page when previewing In-Place eDiscovery search results.	200
DiscoverySearchTimeoutPeriod	The number of minutes that an In-Place eDiscovery search will run before it times out.	10 minutes

Note:

[1] If you initiate an eDiscovery search from the eDiscovery Center in SharePoint Online in an Office 365 organization, you can search a maximum of 1,500 mailboxes in a single search.

In Exchange Server 2013, you can change the default values for these parameters to suit your requirements or create additional throttling policies and assign them to users with delegated Discovery Management permission. In Exchange Online, the default values for these throttling parameters can't be changed.

In-Place eDiscovery documentation

The following table contains links to topics that will help you learn about and manage In-Place eDiscovery.

Topic	Description
Add a user to the Discovery Management role group	Learn how to give a user access to use In-Place eDiscovery in the EAC to search Exchange mailboxes. Adding a user to the Discovery Management role group also allows the person to use the eDiscovery Center in SharePoint 2013 and SharePoint Online to search Exchange mailboxes.

Create a discovery mailbox	Learn how to use the Shell to create a discovery mailbox and assign access permissions.
Create an In-Place eDiscovery search	Learn how to create an In-Place eDiscovery search, and how to estimate and preview eDiscovery search results.
Message properties and search operators for In-Place eDiscovery	Learn which email message properties can be searched using In-Place eDiscovery. The topic provides syntax examples for each property, information about search operators such as **AND** and **OR**, and information about other search query techniques such as using double quotation marks (" ") and prefix wildcards.
Search limits for In-Place eDiscovery in Exchange Online	Learn In-Place eDiscovery limits in Exchange Online that help maintain the health and quality of eDiscovery services for Office 365 organizations.
Start or stop an In-Place eDiscovery search	Learn how to start, stop, and restart eDiscovery searches.
Modify an In-Place eDiscovery search	Learn how to modify an existing eDiscovery search.
Copy eDiscovery search results to a discovery mailbox	Learn how to copy the results of an eDiscovery search to a discovery mailbox.
Export eDiscovery search results to a PST file	Learn how to export the results of an eDiscovery search to a PST file.
Create a custom management scope for In-Place eDiscovery searches	Learn how to use custom management scopes to limit the mailboxes that a discovery manager can search.

Remove an In-Place eDiscovery search	Learn how to delete an eDiscovery search.
Search and delete messages	Learn how to use the **Search-Mailbox** cmdlet to search for and then delete email messages.
Reduce the size of a discovery mailbox in Exchange	Use this process to reduce the size of a discovery mailbox that's larger than 50 GB.
Delete and re-create the default discovery mailbox in Exchange	Learn how to delete the default discovery mailbox, re-create it, and then reassign permissions to it. Use this procedure if this mailbox has exceeded the 50 GB limit and you don't need the search results.
Re-create the Discovery system mailbox	Learn how to recreate the discovery system mailbox. This task is applicable only to Exchange 2013 organizations.
Using OAuth authentication to support eDiscovery in an Exchange hybrid deployment	Learn about the eDiscovery scenarios in an Exchange hybrid deployment that require you to configure OAuth authentication.
Configure Exchange for SharePoint eDiscovery Center	Learn how to configure Exchange 2013 so that you can use the eDiscovery Center in SharePoint 2013 to search Exchange mailboxes.
Unsearchable items in Exchange eDiscovery	Learn about mailbox items that can't be indexed by Exchange Search and are returned in eDiscovery search results as unsearchable items.

For more information about eDiscovery in Office 365, Exchange 2013, SharePoint 2013, and Lync 2013, see the eDiscovery FAQ.

Return to top

Message properties and search operators for In-Place eDiscovery

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-05*

This topic describes the properties of Exchange Exchange email messages that you can search by using In-Place eDiscovery & Hold in Exchange Server 2013 and Exchange Online. The topic also describes Boolean search operators and other search query techniques that you can use to refine eDiscovery search results.

In-Place eDiscovery uses Keyword Query Language (KQL). For more details, see Keyword Query Language syntax reference.

Searchable properties in Exchange

The following table lists email message properties that can be searched using an In-Place eDiscovery search or by using the **New-MailboxSearch** or the **Set-MailboxSearch** cmdlet. The table includes an example of the *property:value* syntax for each property and a description of the search results returned by the examples.

Property	Property description	Examples	Search results returned by the examples
Attachment	The names of files attached to an email message.	attachment:annualreport.ppt attachment:annual*	Messages that have an attached file named annualreport.ppt. In the second example, using the wildcard returns messages with the word "annual" in the file name of an attachment.
Bcc	The BCC field of an email message.[1]	bcc:pilarp@contoso.com	All examples return messages with Pilar

		bcc:pilarp bcc:"Pilar Pinilla"	Pinilla included in the Bcc field.
Body	Text in the body of an email message.	body:"Northwind Traders" body:north*	Messages with the exact phrase "Northwind Traders" in the body of the message. The second example returns any message that contains words that begin with the string "north", such as north, northwind, or northern.
Category	The categories to search. Categories can be defined by users by using Outlook or Outlook Web App. The possible values are: • blue • green • orange • purple • red • yellow	category:"Red Category"	Messages that have been assigned the red category in the source mailboxes.
Cc	The CC field of an email message.[1]	cc:pilarp@contoso.com cc:"Pilar Pinilla"	In both examples, messages with Pilar Pinilla specified in the CC field.
From	The sender of an email message.[1]	from:pilarp@contoso.com	Messages sent by the specified user or sent

		from:contoso.com	from a specified domain.
Importance	The importance of an email message, which a sender can specify when sending a message. By default, messages are sent with normal importance, unless the sender sets the importance as **high** or **low**.	importance:high importance:medium importance:low	Messages that are marked as high importance, medium importance, or low importance.
Kind	The message type to search. Possible values: • contacts • docs • email • faxes • im • journals • meetings • notes • posts • rssfeeds • tasks • voicemail	kind:email kind:email OR kind:im OR kind:voicemail	Email messages that meet the search criteria. The second example returns email messages, instant messaging conversations, and voice messages that meet the search criteria.
Participants	All the people fields in an email message; these fields are From, To, CC, and BCC.[1]	participants:garthf@contoso.com participants:contoso.com	Messages sent by or sent to garthf@contoso.com. The second example returns all messages sent by or sent to a

			user in the contoso.com domain.
Received	The date that an email message was received by a recipient.	received:04/15/2014 received>=01/01/2014 AND received<=03/31/2014	Messages that were received on April 15, 2014. The second example returns all messages received between January 1, 2014 and March 31, 2014.
Recipients	All recipient fields in an email message; these fields are To, CC, and BCC.[1]	recipients:garthf@contoso.com recipients:contoso.com	Messages sent to garthf@contoso.com. The second example returns messages sent to any recipient in the contoso.com domain.
Sent	The date that an email message was sent by the sender.	sent:07/01/2014 sent>=06/01/2014 AND sent<=07/01/2014	Messages that were sent on the specified date or sent within the specified date range.
Size	The size of an item, in bytes.	size>2621440 Size:1..50000	Messages larger than 25 MB. The second example returns messages from 1 through 50,000 bytes in size.
Subject	The text in the subject line of an email message.	subject:"Quarterly Financials" subject:northwind	Messages that contain the exact phrase "Quarterly Financials" in the subject line.

			The second example returns all messages that contain the word northwind in the subject line.
To	The To field of an email message.[1]	to:annb@contoso.com to:annb to:"Ann Beebe"	All examples return messages where Ann Beebe is specified in the To: line.

Note:

[1] For the value of a recipient property, you can use the SMTP address, display name, or alias to specify a user. For example, you can use annb@contoso.com, annb, or "Ann Beebe" to specify the user Ann Beebe.

Supported search operators

Boolean search operators, such as **AND**, **OR**, and **NOT**, help you define more-precise mailbox searches by including or excluding specific words in the search query. Other techniques, such as using property operators (such as >= or ..), quotation marks, parentheses, and wildcards, help you refine eDiscovery search queries. The following table lists the operators that you can use to narrow or broaden search results.

Operator	Usage	Description
AND	keyword1 AND keyword2	Returns messages that include all of the specified keywords or `property:value` expressions.
+	keyword1 + keyword2	The same as the **AND** operator.
OR	keyword1 OR keyword2	Returns messages that include one or more of the specified keywords or `property:value` expressions.
NOT	keyword1 NOT keyword2	Excludes messages specified by

	NOT from:"Ann Beebe"	a keyword or a `property:value` expression. For example, `NOT from:"Ann Beebe"` excludes messages sent by Ann Beebe.
-	keyword1 -keyword2	The same as the **NOT** operator.
NEAR	keyword1 NEAR(n) keyword2	Returns messages with words that are near each other, where n equals the number of words apart. For example, `best NEAR(5) worst` returns messages where the word "worst" is within five words of "best". If no number is specified, the default distance is eight words.
:	property:value	The colon (:) in the `property:value` syntax specifies that the property value being searched for equals the specified value. For example, `recipients:garthf@contoso.com` returns any message sent to garthf@contoso.com.
=	property=value	The same as using the `property:value` syntax.
<	property<value	Denotes that the property being searched is less than the specified value. [1]

>	property>value	Denotes that the property being searched is greater than the specified value.[1]
<=	property<=value	Denotes that the property being searched is less than or equal to a specific value.[1]
>=	property>=value	Denotes that the property being searched is greater than or equal to a specific value.[1]
..	property:value1..value2	Denotes that the property being searched is greater than or equal to value1 and less than or equal to value2.[1]
" "	"fair value" subject:"Quarterly Financials"	Use double quotation marks (" ") to search for an exact phrase or term in keyword and `property:value` search queries.
*	cat* subject:set*	Prefix wildcard searches (where the asterisk is placed at the end of a word) match for zero or more characters in keywords or `property:value` queries. For example, `subject:set*` returns messages that contain the word set, setup, and setting (and other words that start with "set") in the subject line.
( )	(fair OR free) AND	Parentheses group together

	from:contoso.com (IPO OR initial) AND (stock OR shares) (quarterly financials)	Boolean phrases, `property:value` items, and keywords. For example, `(quarterly financials)` returns items that contain the words quarterly and financials.

Note:

[1] Use this operator for properties that have date or numeric values.

Search tips and tricks

- Keyword searches are not case sensitive. For example, **cat** and **CAT** return the same results.
- The Boolean operators **AND**, **OR**, **NOT**, and **NEAR** must be uppercase.
- A space between two keywords or two `property:value` expressions is the same as using **AND**. For example, `from:"Sara Davis" subject:reorganization` returns all messages sent by Sara Davis that contain the word **reorganization** in the subject line.
- Use syntax that matches the `property:value` format. Values are not case-sensitive, and they can't have a space after the operator. If there is a space, your intended value will just be full-text searched. For example **to: pilarp** searches for "pilarp" as a keyword, rather than for messages that were sent to pilarp.
- When searching a recipient property, such as To, From, Cc, or Recipients, you can use an SMTP address, alias, or display name to denote a recipient. For example, you can use pilarp@contoso.com, pilarp, or "Pilar Pinilla".
- You can use only prefix wildcard searches—for example, **cat*** or **set***. Suffix wildcard searches (*cat) or substring wildcard searches (*cat*) aren't supported.
- When searching a property, use double quotation marks (" ") if the search value consists of multiple words. For example **subject:budget Q1** returns messages that contain **budget** in the in the subject line and that contain **Q1** anywhere in the message or in any of the message properties. Using **subject:"budget Q1"** returns all messages that contain **budget Q1** in the subject line.

Add a user to the Discovery Management role group

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-12*

If you want users to be able to use Microsoft Exchange Server 2013 In-Place eDiscovery, you must first authorize them by adding them to the Discovery Management role group. Members of the Discovery Management role group have Full Access mailbox permissions for the Discovery mailbox that's created by Exchange Setup.

Caution:

Members of the Discovery Management role group can access sensitive message content. Specifically, these members can use In-Place eDiscovery to search all mailboxes in your Exchange organization, preview messages (and other mailbox items), copy them to a Discovery mailbox and export the copied messages to a .pst file. In most organizations, this permission is granted to legal, compliance, or Human Resources personnel.

To learn more about the Discovery Management role group, see Discovery Management. To learn more about Role Based Access Control (RBAC), see Understanding Role Based Access Control.

Interested in scenarios where this procedure is used? See the following topics:

- Create an In-Place eDiscovery search
- Create or remove an In-Place Hold

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.
- By default, the Discovery Management role group doesn't contain any members. Administrators with the Organization Management role are also unable to create or manage discovery searches without being added to the Discovery Management role group.
- In Exchange 2013, members of the Organization Management role group can create an In-Place Hold to place all mailbox content on hold. However, to create a query-based In-Place Hold, the user must be a member of the Discovery Management role group or have the Mailbox Search role assigned.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Use the EAC to add a user to the Discovery Management role group

1. Navigate to **Permissions** > **Admin roles**.
2. In the list view, select **Discovery Management** and then click **Edit**
3. In **Role Group**, under **Members**, click **Add +**.
4. In **Select Members**, select one or more users, click **Add**, and then click **OK**.

5. In **Role Group**, click **Save**.

Use the Shell to add a user to the Discovery Management role group

This example adds the user Bsuneja to the Discovery Management role group.

```
Add-RoleGroupMember -Identity "Discovery Management" -
Member Bsuneja
```

For detailed syntax and parameter information, see Add-RoleGroupMember.

How do you know this worked?

To verify that you've added the user to the Discovery Management role group, do the following:

1. Navigate to **Permissions** > **Admin roles**.
2. In the list view, select **Discovery Management**.
3. In the details pane, verify that the user is listed under **Members**.

You can also run this command to list the members of the Discovery Management role group.

```
Get-RoleGroupMember -Identity "Discovery Management"
```

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Create a discovery mailbox

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-21*

Microsoft Exchange Server 2013 Setup creates a discovery mailbox by default. In Exchange Online, a discovery mailbox is also created by default. Discovery mailboxes are used as target mailboxes for In-Place eDiscovery searches in the Exchange Admin Center (EAC). You can create additional discovery mailboxes as required. After you create a new discovery mailbox, you will have to assign Full Access permissions to the appropriate users so they can access eDiscovery search results that are copied to the discovery mailbox.

Caution:

After a discovery manager copies the results of an eDiscovery search to a discovery mailbox, the mailbox can potentially contain sensitive information. You should control access to discovery mailboxes and make sure only authorized users can access them.

For more information, see Discovery mailboxes.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Creating discovery mailboxes" entry in Messaging policy and compliance permissions topic.
- Discovery mailboxes have a mailbox storage quota of 50 gigabytes (GB). This storage quota can't be increased.
- You can't use the EAC to create a discovery mailbox or assign permissions to access it. You have to use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a discovery mailbox

This example creates a discovery mailbox named SearchResults.

```
New-Mailbox -Name SearchResults -Discovery
```

For detailed syntax and parameter information, see New-Mailbox.

To display a list of all discovery mailboxes in an Exchange organization, run the following command:

```
Get-Mailbox -Resultsize unlimited -Filter
{RecipientTypeDetails -eq "DiscoveryMailbox"}
```

For detailed syntax and parameter information, see Get-Mailbox.

Assign permissions to a discovery mailbox

You have to explicitly assign users or groups the necessary permissions to open a discovery mailbox

that you've created. Run the following command to assign a user or group permissions to open a discovery mailbox and view search results:

```
Add-MailboxPermission <Name of the discovery mailbox> -User
<Name of user or group> -AccessRights FullAccess -
InheritanceType all
```

For example, the following command assigns the Full Access permission to the Litigation Managers group, so members of the group can open the Fabrikam Litigation discovery mailbox.

```
Add-MailboxPermission "Fabrikam Litigation" -User
"Litigation Managers" -AccessRights FullAccess -
InheritanceType all
```

For detailed syntax and parameter information, see Add-MailboxPermission.

More information

- By default, members of the Discovery Management role group only have Full Access permission to the default Discovery Search Mailbox. You will have to explicitly assign the Full Access permission to the Discovery Management role group if you want members to open a discovery mailbox that you've created.
- Although visible in Exchange address lists, users can't send email to a discovery mailbox. Email delivery to discovery mailboxes is prohibited with delivery restrictions. This preserves the integrity of search results copied to a discovery mailbox.
- A discovery mailbox can't be repurposed or converted to another type of mailbox.
- You can remove a discovery mailbox as you would any other type of mailbox.

Create an In-Place eDiscovery search

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-27*

Use In-Place eDiscovery to search across all mailbox content, including deleted items and original versions of modified items for users placed on In-Place Hold.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.

- To create eDiscovery searches, you have to have an SMTP address in the organization that you're creating the searches in. So in Exchange Online, you must have a licensed Exchange Online mailbox to create eDiscovery searches. In an Exchange hybrid organization, your on-premises Exchange mailbox must have a corresponding mail user account in your Office 365 organization so that you can search Exchange Online mailboxes. Or, if you sign in with an account that only exists in Office 365, such as the tenant administrator account, that account must be assigned an Exchange Online license.
- Exchange 2013 Setup creates a Discovery mailbox called **Discovery Search Mailbox** to copy search results. The Discovery Search Mailbox is also created by default in Exchange Online. You can create additional Discovery mailboxes. For details, see Create a discovery mailbox.
- When you create an In-Place eDiscovery search, messages returned in search results aren't copied automatically to a discovery mailbox. After you create the search, you can use the Exchange Admin Center (EAC) to estimate and preview search results or copy them to a discovery mailbox. For details, see:
  - Estimate or preview search results (later in this topic)
  - Copy eDiscovery search results to a discovery mailbox
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create an In-Place eDiscovery search

Use the EAC to create an In-Place eDiscovery search

As previously explained, to create eDiscovery searches, you have to sign in to a user account that has an SMTP address in your organization.

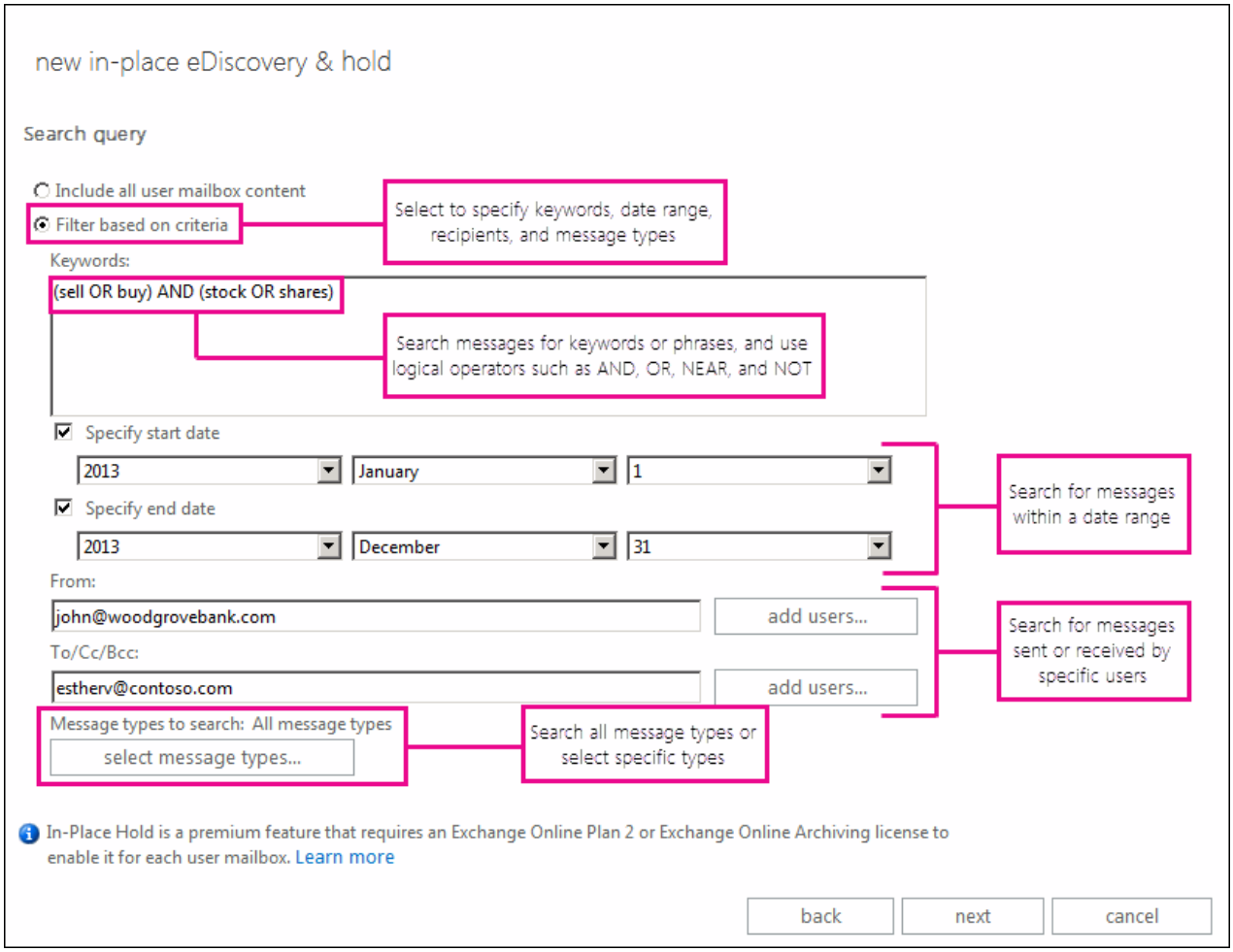
1. Go to **Compliance management** > **In-place eDiscovery & hold**.
2. Click **New +**.
3. In **In-Place eDiscovery & Hold**, on the **Name and description** page, type a name for the search, add an optional description, and then click **Next**.
4. On the **Mailboxes** page, select the mailboxes to search. You can search across all mailboxes or select specific ones to search.

Important:

You can't use the **Search all mailboxes** option to place all mailboxes on hold. To create an In-Place Hold, you must select **Specify mailboxes to search**. For more details, see Create or

remove an In-Place Hold.

5. On the **Search query** page, complete the following fields:
   - **Include all user mailbox content** Select this option to place all content in the selected mailboxes on hold. If you select this option, you can't specify additional search criteria.
   - **Filter based on criteria** Select this option to specify search criteria, including keywords, start and end dates, sender and recipient addresses, and message types.



6. On the **In-place hold settings** page, you can select the **Place content matching the search query in selected mailboxes on hold** check box, and then select one of the following options to place items on In-Place Hold:
   - **Hold indefinitely** Select this option to place the returned items on an indefinite hold. Items on hold will be preserved until you remove the mailbox from the search or remove the search.
   - **Specify number of days to hold items relative to their received date** Use this option to hold items for a specific period. For example, you can use this option if your organization requires that all messages be retained for at least seven years. You can use a *time-based* In-Place Hold along with a retention policy to make sure items are deleted in seven years.

Important:

When placing mailboxes or items on In-Place Hold for legal purposes, it is generally recommended to hold items indefinitely and remove the hold when the case or investigation is completed.

7. Click **Finish** to save the search and return an estimate of the total size and number of items that will be returned by the search based on the criteria you specified. Estimates are displayed in the

details pane. Click **Refresh** to update the information displayed in the details pane.

Use the Shell to create an In-Place eDiscovery search

This example creates the In-Place eDiscovery search Discovery-CaseId012 for items containing the keywords Contoso and ProjectA that also meet the following criteria:

- Start date: 1/1/2009
- End date: 12/31/2011
- Source mailbox: DG-Finance
- Target mailbox: Discovery Search Mailbox
- Message types: Email
- Log level: Full

Important:

If you don't specify additional search parameters when running an In-Place eDiscovery search, all items in the specified source mailboxes are returned in the results. If you don't specify mailboxes to search, all mailboxes in your Exchange or Exchange Online organization are searched.

```
New-MailboxSearch "Discovery-CaseId012" -StartDate
"1/1/2009" -EndDate "12/31/2011" -SourceMailboxes "DG-
Finance" -TargetMailbox "Discovery Search Mailbox" -
SearchQuery '"Contoso" AND "Project A"' -MessageTypes Email
-IncludeUnsearchableItems -LogLevel Full
```

Note:

When using the *StartDate* and *EndDate* parameters, you have to use the date format of mm/dd/yyyy, even if your local machine settings are configured to use a different date format, such as dd/mm/yyyy. For example, to search for messages sent between April 1, 2013 and July 1, 2013, you would use **04/01/2013** and **07/01/2013** for the start and end dates.

After using the Shell to create an In-Place eDiscovery search, you have to start the search by using the **Start-MailboxSearch** cmdlet to copy messages to the discovery mailbox specified in the *TargetMailbox* parameter. For details, see Copy eDiscovery search results to a discovery mailbox.

For detailed syntax and parameter information, see New-MailboxSearch.

Estimate or preview search results

After you create an In-Place eDiscovery search, you can use the EAC to get an estimate and preview of the search results. If you created a new search using the **New-MailboxSearch** cmdlet, you can use the Shell to start the search to get an estimate of the search results. You can't use the Shell to preview messages returned in search results.

Use the EAC to estimate or preview search results

1. Navigate to **Compliance management** > **In-place eDiscovery & hold**.

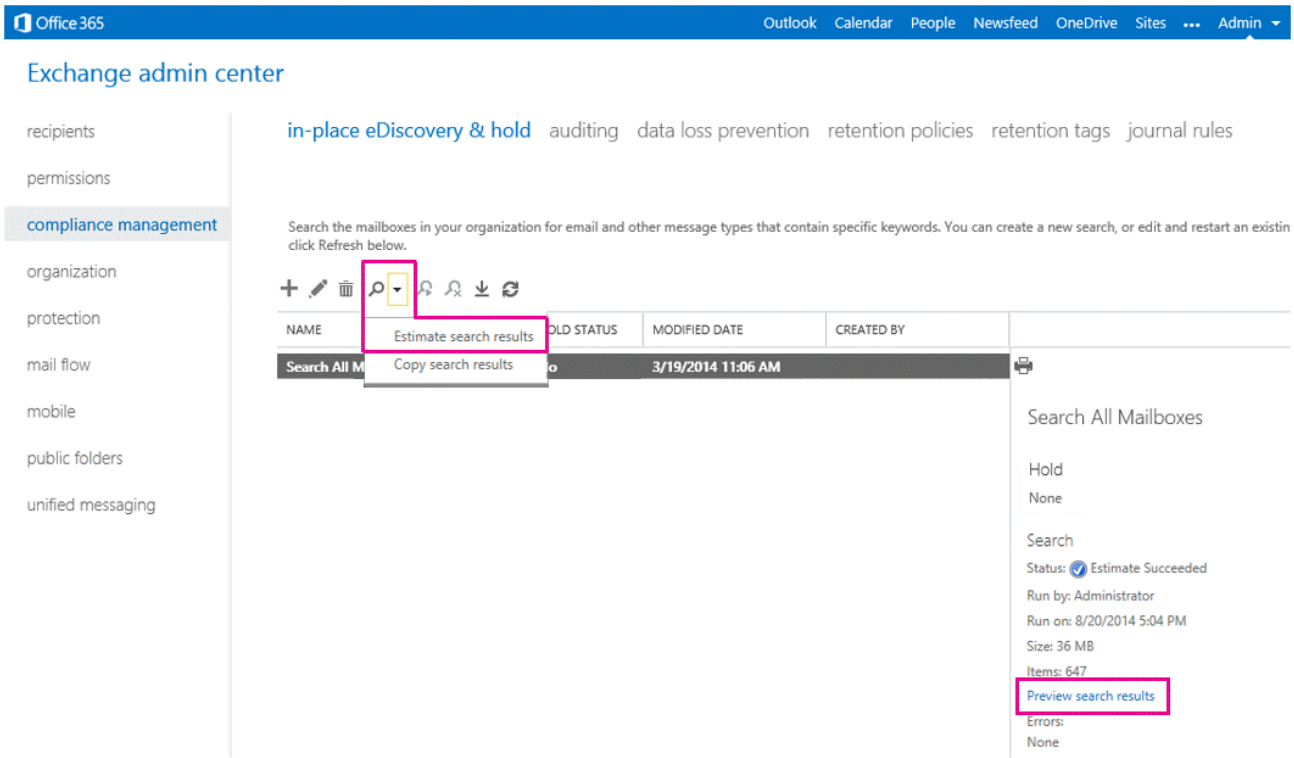
2. In the list view, select the In-Place eDiscovery search, and then do one of the following:
   - Click **Search** > **Estimate search results** to return an estimate of the total size and number of items that will be returned by the search based on the criteria you specified. Selecting this option restarts the search and performs an estimate.

Search Estimates are displayed in the details pane. Click **Refresh** to update the information displayed in the details pane.

   - Click **Preview search results** in the details pane to preview the results after the search estimate is completed. Selecting this option opens the **eDiscovery search preview** window. All messages returned from the mailboxes that were searched are displayed.

Note:

The mailboxes that were searched are listed in the right pane in the **eDiscovery search preview** window. For each mailbox, the number of items returned and the total size of these items is also displayed. All items returned by the search are listed in the right pane, and can be sorted by newest or oldest date. Items from each mailbox can't be displayed in the right pane by clicking a mailbox in the left pane. To view the items returned from a specific mailbox, you can copy the search results and view the items in the discovery mailbox.



Use the Shell to estimate search results

You can use the *EstimateOnly* switch to return only get an estimate of the search results and not copy the results to a discovery mailbox. You have to start an estimate-only search with the **Start-MailboxSearch** cmdlet. Then you can retrieve the estimated search results by using the **Get-MailboxSearch** cmdlet.

For example, you would run the following commands to create a new eDiscovery search and then display an estimate of the search results:

```
New-MailboxSearch "FY13 Q2 Financial Results" -StartDate
```

```
"04/01/2013" -EndDate "06/30/2013" -SourceMailboxes "DG-
Finance" -SearchQuery '"Financial" AND "Fabrikam"' -
EstimateOnly -IncludeKeywordStatistics

Start-MailboxSearch "FY13 Q2 Financial Results"

Get-MailboxSearch "FY13 Q2 Financial Results"
```

To display specific information about the estimated search results from the previous example, you could run the following command:

```
Get-MailboxSearch "FY13 Q2 Financial Results" | FL
Name,Status,LastRunBy,LastStartTime,LastEndTime,Sources,Sea
rchQuery,ResultSizeEstimate,ResultNumberEstimate,Errors,Key
wordHits
```

More information about eDiscovery searches

- After you create a new eDiscovery search, you can copy search results to the discovery mailbox and export those search results to a PST file. For more information, see:
  - Copy eDiscovery search results to a discovery mailbox
  - Export eDiscovery search results to a PST file
- After you run an eDiscovery search estimate (that includes keywords in the search criteria), you can view keyword statistics by clicking **View keyword statistics** in the details pane for the selected search. These statistics show details about the number of items returned for each keyword used in the search query. However, if more than 100 source mailboxes are included in the search, an error will be returned if you try to view keyword statistics. To view keyword statistics, no more than 100 source mailboxes can be included in the search.
- If you use **Get-MailboxSearch** in Exchange Online to retrieve information about an eDiscovery search, you have to specify the name of a search to return a complete list of the search properties; for example, `Get-MailboxSearch "Contoso Legal Case"`. If you run the **Get-MailboxSearch** cmdlet without using any parameters, the following properties aren't returned:
  - SourceMailboxes
  - Sources
  - SearchQuery
  - ResultsLink
  - PreviewResultsLink
  - Errors

The reason is that it requires a lot of resources to return these properties for all eDiscovery searches in your organization.

Start or stop an In-Place eDiscovery search

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-14*

You can stop or restart an In-Place eDiscovery search at any time. For example, if you want to modify search properties such as keywords or mailboxes searched, you must first stop a search. You can then restart the search after making the required changes.

Caution:

If you restart an In-Place eDiscovery search, search results copied to the Discovery mailbox specified in the search are removed.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place eDiscovery" entry in Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to start or stop an In-Place eDiscovery search

1. Navigate to **Compliance management** > **In-place eDiscovery & hold**.
2. To stop a search that's in progress, select the search, and then click **Stop search**.
3. To start a search that was stopped, select the search, and then click **Resume search**.

Use the Shell to start or stop an In-Place eDiscovery search

For an example of how to start an In-Place eDiscovery search, see "Example 1" in Start-MailboxSearch.

For an example of how to stop an In-Place eDiscovery search, see “Example 1” in Stop-MailboxSearch.

Modify an In-Place eDiscovery search

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-27*

After you create an In-Place eDiscovery search, you can modify it to change the search parameters. For example, you can change the mailboxes to be searched, date ranges, key words, logging options, or you can specify a different Discovery mailbox to store search results. Any changes you make to the search properties will be used when you restart the search.

Caution:

If an In-Place eDiscovery search is running, you must stop it before modifying it. When you restart the search, the results from the last time the search was run are removed from the Discovery mailbox. However, the logs from previous searches are saved.

What do you need to know before you begin?

- Estimated time to complete: 2-5 minutes.
- An In-Place eDiscovery search has been created and isn't running.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to modify an In-Place eDiscovery search

1. Navigate to **Compliance management** > **In-place eDiscovery & hold**.
2. In the list view, select the In-Place eDiscovery search you want to modify, and then click **Edit** ✎.
3. In **In-Place eDiscovery & Hold**, you can modify the following settings:
   - On **Name** page, modify the name for the search and the optional description.
   - On the **Mailboxes** page, modify the mailboxes to search. You can search across all mailboxes or select specific ones to search.

Important:

You can't use the **Search all mailboxes** option to place all mailboxes on Exchange 2013 Mailbox servers on hold. To create an In-Place Hold, you must select **Specify mailboxes to search**. For more details, see Create or remove an In-Place Hold.

- On the **Search query** page, modify the following fields:
  - **Include all user mailbox content** Select this option to place all content in the selected mailboxes on hold.
  - **Filter based on criteria** Select this option to specify search criteria, including keywords, start and end dates, sender and recipient addresses, and message types.
- On the **In-Place Hold** page, select the **Place content matching the search query in selected mailboxes on hold** check box, and then select one of the following options to place items on In-Place Hold:
  - **Hold indefinitely** Select this option to place the returned items on an indefinite hold. Items on hold will be preserved until you remove the mailbox from the search or remove the search.
  - **Specify number of days to hold items relative to their received date** Use this option to hold items for a specific period. For example, you can use this option if your organization requires that all messages be retained for at least seven years. You can use a *time-based* In-Place Hold along with a retention policy to make sure items are deleted in seven years.

Important:

When placing mailboxes or items on In-Place Hold for legal purposes, it is generally recommended to hold items indefinitely and remove the hold when the case or investigation is completed.

4. Click **Save**.

Use the Shell to modify an In-Place eDiscovery search

This example modifies the In-Place eDiscovery search Search-Project Contoso to search mailboxes belonging to members of the DG-ProjectManagers distribution group.

```
Set-MailboxSearch -Identity "Search-Project Contoso" -
SourceMailboxes "DG-ProjectManagers"
```

How do you know this worked?

To verify that you have successfully modified an In-Place eDiscovery search, do one of the following:

- Use the EAC to check properties of the search.
- Use the **Get-MailboxSearch** cmdlet from the Shell to check the properties of the search. For examples of how to check the properties of a mailbox search, see the "Examples" section in Get-MailboxSearch.

Note:

If you use **Get-MailboxSearch** in Exchange Online to retrieve information about an eDiscovery search, you have to specify the name of a search to return a complete list of the search properties; for example, `Get-MailboxSearch "Contoso Legal Case"`. If you run the **Get-MailboxSearch** cmdlet without using any parameters, the following properties aren't returned:

- SourceMailboxes
- Sources
- SearchQuery
- ResultsLink
- PreviewResultsLink
- Errors

The reason is that it requires a lot of resources to return these properties for all eDiscovery searches in your organization.

Copy eDiscovery search results to a discovery mailbox

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Topic Last Modified: *2014-02-24*

After you create an In-Place eDiscovery search, you can use the EAC to copy the results to a discovery mailbox. You can also use the Shell to start an eDiscovery search that was created using the **New-MailboxSearch** cmdlet, which will copy the results to the discovery mailbox that was specified when you created the search.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes or longer depending on the number of mailbox items returned in the search results
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.
- An eDiscovery search has to be created, by using the EAC or the Shell, before you can copy the search results. For details, see Create an In-Place eDiscovery search.
- Exchange 2013 Setup creates a discovery mailbox called **Discovery Search Mailbox** to copy search results. The Discovery Search Mailbox is also created by default in Exchange Online. You can create additional discovery mailboxes. For details, see Create a discovery mailbox.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

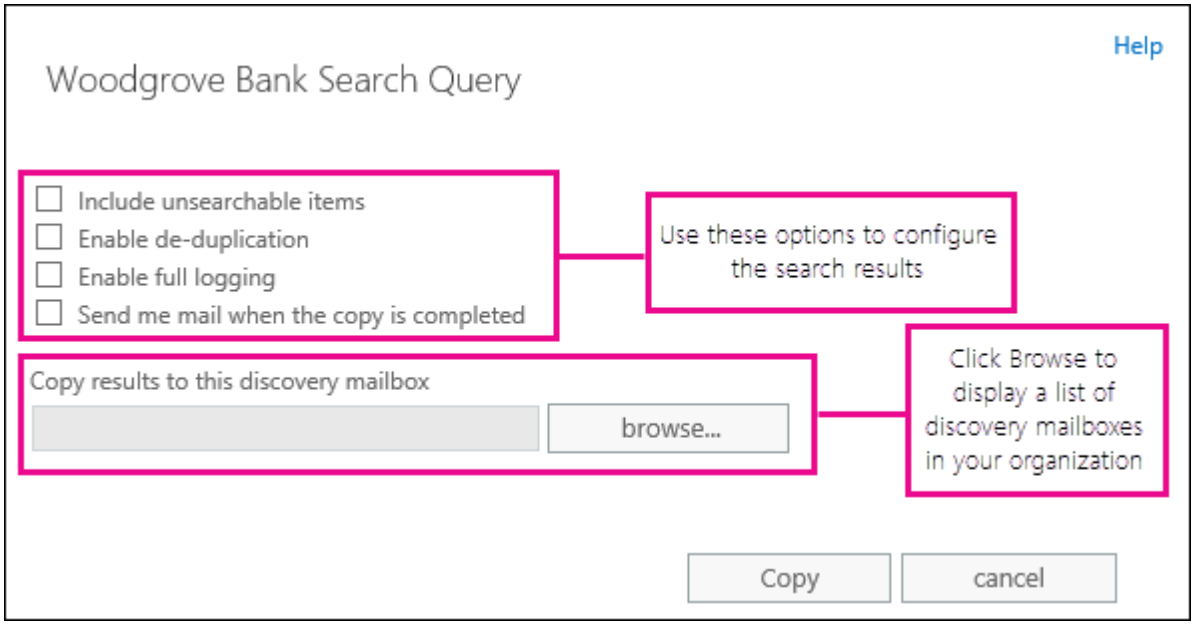
Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to copy search results

1. In the EAC, go to **Compliance management** > **In-place eDiscovery & hold**.
2. In the list view, select an eDiscovery search.
3. Click **Search**, and then click **Copy search results** from the drop-down list.
4. In **Copy Search Results**, select from the following options:
   - **Include unsearchable items** Select this check box to include mailbox items that couldn't be searched (for example, messages with attachments of file types that couldn't be indexed by Exchange Search). For more information, see Unsearchable items in Exchange eDiscovery.
   - **Enable de-duplication** Select this check box to exclude duplicate messages. Only a single instance of a message will be copied to the discovery mailbox.
   - **Enable full logging** Select this check box to include a full log in search results.
   - **Send me mail when the copy is completed** Select this check box to get an email notification when the search is completed.
   - **Copy results to this discovery mailbox** Click **Browse** to select the discovery mailbox where you want the search results copied to.



5. Click **Copy** to start the process to copy the search results to the specified discovery mailbox.
6. Click **Refresh** to update the information about the copying status that is displayed in the details pane.
7. When copying is complete, click **Open** to open the discovery mailbox to view the search results.

Use the Shell to copy search results

After using the **New-MailboxSearch** cmdlet to create an In-Place eDiscovery search, you must start the search to copy messages to the discovery mailbox you specified in the *TargetMailbox* parameter. For information about creating eDiscovery searches using the Shell, see:

- Use the Shell to create an In-Place eDiscovery search
- New-MailboxSearch

For example, you would run the following command to start an eDiscovery search named *Fabrikam Investigation* to copy the search results to the specified discovery mailbox.

```
Start-MailboxSearch "Fabrikam Investigation"
```

If you used the *EstimateOnly* switch to get an estimate of the search results, you have to remove the switch before you can copy the search results. You also have to specify a discovery mailbox to copy to search results to. For example, say you created an estimate-only search by using the following command:

```
New-MailboxSearch "FY13 Q2 Financial Results" -StartDate
"04/01/2013" -EndDate "06/30/2013" -SourceMailboxes "DG-
Finance" -SearchQuery '"Financial" AND "Fabrikam"' -
EstimateOnly -IncludeUnsearchableItems
```

To copy the results of this search to a discovery mailbox, you would run the following commands:

```
Set-MailboxSearch "FY13 Q2 Financial Results" -EstimateOnly
$false -TargetMailbox "Discovery Search Mailbox"

Start-MailboxSearch "FY13 Q2 Financial Results"
```

More information about copying search results

- After you copy search results to the discovery mailbox, you can export those search results to a PST file. For more information, see Export eDiscovery search results to a PST file.
- For more information about unsearchable items, see Unsearchable items in Exchange eDiscovery.
- If you're copying all mailbox content within a specific date range (by not specifying any keywords in the search criteria), then all unsearchable items within that date range will be automatically included in the search results. Therefore, don't select the **Include unsearchable items** checkbox when copying search results. Otherwise, a duplicate copy of all unsearchable items will be copied to the discovery mailbox.
- In addition to copying the search results to a discovery mailbox, you can also estimate or preview the search results for a selected search.
  - **Estimate search results** This option returns an estimate of the total size and number of items that will be returned by the search based on the criteria you specified. Estimates are displayed in the details pane in the EAC.

- **Preview search results** This option lets you preview the search results returned by the search instead of having to copy them to a discovery mailbox to view. This lets you quickly determine whether the search results are relevant. After you preview the results, you can revise your search query to narrow the search results and rerun the search. Items in the preview page are read-only versions of the actual search results, so you can't move, edit, delete or forward on the preview page.

For more information, see Estimate or preview search results.

Export eDiscovery search results to a PST file

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-25*

You can use the eDiscovery Export tool in the EAC to export the results of an In-Place eDiscovery search to an Outlook Data File, which is also called a PST file. This lets you distribute the results to other people within your organization, such as a human resources manager or records manager, or to opposing counsel in a legal case. After search results are exported to a PST file, you or other users can open them in Outlook to review or print messages returned in the search results. PST files can also be opened in third-party eDiscovery and reporting applications.

What do you need to know before you begin?

- Estimated time to complete: Time will vary based on the amount and size of the search results that will be exported.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.
- The computer you use to export search results to a PST file has to meet the following system requirements:
  - 32- and 64-bit versions of Windows 7 and later versions
  - Microsoft .NET Framework 4.5
  - A supported browser:
    - Internet Explorer 8 and later versions

OR

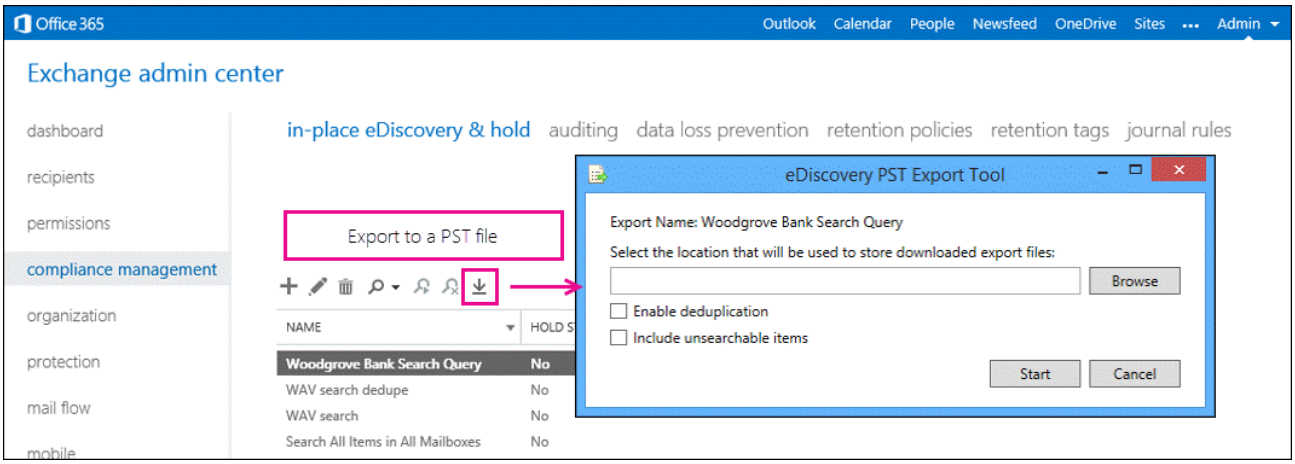
    - Mozilla Firefox or Google Chrome, with the ClickOnce add-in installed
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to export search results

1. Go to **Compliance management** > **In-place eDiscovery & hold**.
2. In the list view, select the In-Place eDiscovery search you want to export the results of, and then click **Export to a PST file**.



3. In the **eDiscovery PST Export Tool** window, do the following:
   - Click **Browse** to specify the location where you want to download the PST file.
   - Click the **Enable deduplication** checkbox to exclude duplicate messages. Only a single instance of a message will be included in the PST file.
   - Click the **Include unsearchable items** checkbox to include mailbox items that couldn't be searched (for example, messages with attachments of file types that couldn't be indexed by Exchange Search). Unsearchable items are exported to a separate PST file.

Important:

Including unsearchable items when you export eDiscovery search results takes longer when mailboxes contain a lot of unsearchable items. To reduce the time it takes to export search results and prevent large PST export files, consider the following recommendations:

- Create multiple eDiscovery searches that each search a fewer number of source mailboxes.
- If you're exporting all mailbox content within a specific date range (by not specifying any keywords in the search criteria), then all unsearchable items within that date range will be automatically included in the search results. Therefore, don't select the **Include unsearchable items** checkbox.

1. Click **Start** to export the search results to a PST file.

A window is displayed that contains status information about the export process.

More information

- Another way to reduce the size of PST export files is to export only the unsearchable items for an

eDiscovery search. To do this, create or edit a search, specify a start date in the future, and then remove any keywords from the **Keywords** box. This will result in no search results being returned. When you copy or export the search results and select the **Include unsearchable items** checkbox, only the unsearchable items will be copied to the discovery mailbox or exported to a PST file.

- If you enable de-duplication, all search results are exported in a single PST file. If you don't enable de-duplication, a separate PST file is exported for each mailbox included in the search. And as previously stated, unsearchable items are exported to a separate PST file.
- In addition to the PST files that contain the search results, two other files are also exported:
  - A configuration file (.txt file format) that contains information about the PST export request, such as the name of the eDiscovery search that was exported, the date and time of the export, whether de-duplication and unsearchable items were enabled, the search query, and the source mailboxes that were searched.
  - A search results log (.csv file format) that contains an entry for each message returned in the search results. Each entry identifies the source mailbox where the message is located. If you've enabled de-duplication, this helps you identify all mailboxes that contain a duplicate message.
- The name of the search is the first part of the filename for each file that is exported. Also, the date and time of the export request is appended to the filename of each PST file and the results log.
- For more information about de-duplication and unsearchable items, see:
  - Estimate, preview, and copy search results
  - Unsearchable items in Exchange eDiscovery
- To export eDiscovery search results from the eDiscovery Center in SharePoint or SharePoint Online, see Export eDiscovery content and create reports.

Create a custom management scope for In-Place eDiscovery searches

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-14*

You can use a custom management scope to let specific people or groups use In-Place eDiscovery to search a subset of mailboxes in your Exchange 2013 or Exchange Online organization. For example, you might want to let a discovery manager search only the mailboxes of users in a specific location or department. You can do this by creating a custom management scope. This custom management scope uses a recipient filter to control which mailboxes can be searched. Recipient filter scopes use filters to target specific recipients based on recipient type or other recipient properties.

For In-Place eDiscovery, the only property on a user mailbox that you can use to create a recipient

filter for a custom scope is distribution group membership (the actual property name is *MemberOfGroup*). If you use other properties, such as *CustomAttributeN*, *Department*, or *PostalCode*, the search fails when it's run by a member of the role group that's assigned the custom scope.

To learn more about management scopes, see:

- Understanding management role scopes
- Understanding management role scope filters

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- As previously stated, you can only use group membership as the recipient filter to create a custom recipient filter scope that is intended to be used for eDiscovery. Any other recipient properties can't be used to create a custom scope for eDiscovery searches. Note that membership in a dynamic distribution group can't be used either.
- Perform steps 1 through 3 to let a discovery manager export the search results for an eDiscovery search that uses a custom management scope.
- If your discovery manager doesn't need to preview the search results, you can skip step 4.
- If your discovery manager doesn't need to copy the search results, you can skip step 5.

Step 1: Organize users into distribution groups for eDiscovery

To search a subset of mailboxes in your organization or to narrow the scope of source mailboxes that a discovery manager can search, you'll need to group the subset of mailboxes into one or more distribution groups. When you create a custom management scope in step 2, you'll use these distribution groups as the recipient filter to create a custom management scope. This allows a discovery manager to search only the mailboxes of the users who are members of a specified group.

You might be able to use existing distribution groups for eDiscovery purposes, or you can create new ones. See More information at the end of this topic for tips on how to create distribution groups that can be used to scope eDiscovery searches.

Step 2: Create a custom management scope

Now you'll create a custom management scope that's defined by the membership of a distribution group (using the *MemberOfGroup* recipient filter). When this scope is applied to a role group used for eDiscovery, members of the role group can search the mailboxes of users who are members of the distribution group that was used to create the custom management scope.

This procedure uses Exchange Management Shell commands to create a custom scope named Ottawa Users eDiscovery Scope. It specifies the distribution group named Ottawa Users for the

recipient filter of the custom scope.

1. Run this command to get and save the properties of the Ottawa Users group to a variable, which is used in the next command.

```
$DG = Get-DistributionGroup -Identity "Ottawa Users"
```

2. Run this command to create a custom management scope based on the membership of the Ottawa Users distribution group.

```
New-ManagementScope "Ottawa Users eDiscovery Scope" -
RecipientRestrictionFilter "MemberOfGroup -eq
'$($DG.DistinguishedName)'"
```

The distinguished name of the distribution group, which is contained in the variable **$DG**, is used to create the recipient filter for the new management scope.

Step 3: Create a management role group

In this step, you create a new management role group and assign the custom scope that you created in step 2. Add the Legal Hold and Mailbox Search roles so that role group members can perform In-Place eDiscovery searches and place mailboxes on In-Place Hold or Litigation Hold. You can also add members to this role group so they can search the mailboxes of the members of the distribution group used to create the custom scope in step 2.

In the following examples, the Ottawa Users eDiscovery Managers security group will be added as members this role group. You can use either the Shell or the EAC for this step.

Use the Shell to create a management role group

Run this command to create a new role group that uses the custom scope created in step 2. The command also adds the Legal Hold and Mailbox Search roles, and adds the Ottawa Users eDiscovery Managers security group as members of the new role group.

```
New-RoleGroup "Ottawa Discovery Management" -Roles "Mailbox
Search","Legal Hold" -CustomRecipientWriteScope "Ottawa
Users eDiscovery Scope" -Members "Ottawa Users eDiscovery
Managers"
```

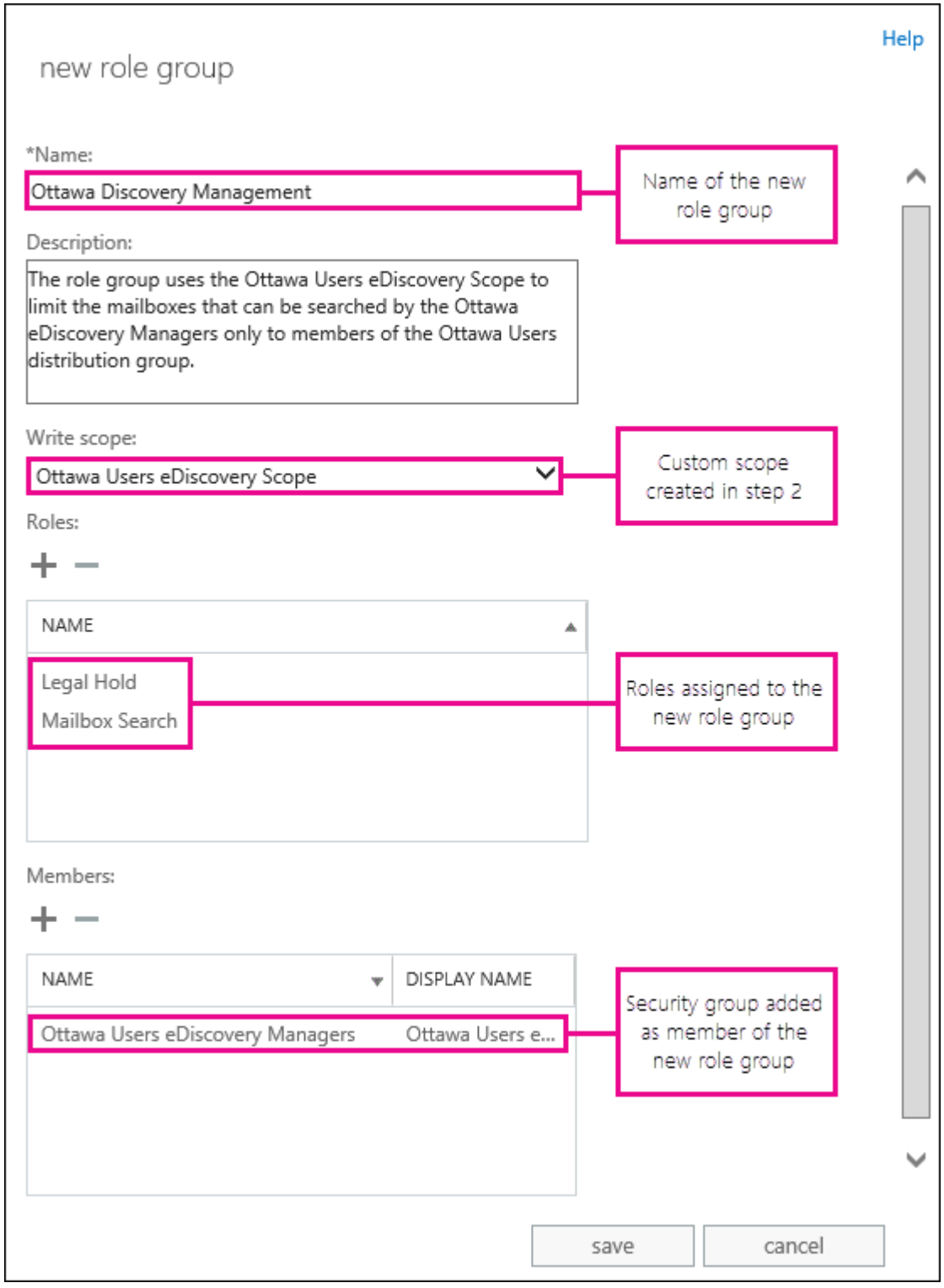
Use the EAC to create a management role group

1. In the EAC, go to **Permissions** > **Admin roles**, and then click **New +**.
2. In **New role group**, provide the following information:
   - **Name** Provide a descriptive name for the new role group. For this example, you'd use **Ottawa Discovery Management**.

- **Write scope** Select the custom management scope that you created in step 2. This scope will be applied to the new role group.
- **Roles** Click **Add +**, and add the **Legal Hold** and **Mailbox Search** roles to the new role group.
- **Members** Click **Add +**, and select the users, security group, or role groups that you want add as members of the new role group. For this example, the members of the **Ottawa Users eDiscovery Managers** security group will be able to search only the mailboxes of users who are members of the **Ottawa Users** distribution group.

3. Click **Save** to create the role group.

Here's an example of what the **New role group** window will look like when you're done.



(Optional) Step 4: Add discovery managers as members of the distribution group used to create the custom management scope

You only need to perform this step if you want to let a discovery manager preview eDiscovery search results.

Run this command to add the Ottawa Users eDiscovery Managers security group as a member of the Ottawa Users distribution group.

```
Add-DistributionGroupMember -Identity "Ottawa Users" -
Member "Ottawa Users eDiscovery Managers"
```

You can also use the EAC to add members to a distribution group. For more information, see Manage Distribution Groups.

(Optional) Step 5: Add a discovery mailbox as a member of the distribution group used to create the custom management scope

You only need to perform this step if you want to let a discovery manager copy eDiscovery search results.

Run this command to add a discovery mailbox named Ottawa Discovery Mailbox as a member of the Ottawa Users distribution group.

```
Add-DistributionGroupMember -Identity "Ottawa Users" -
Member "Ottawa Discovery Mailbox"
```

Note:

To open a discovery mailbox and view the search results, discovery managers must be assigned Full Access permissions for the discovery mailbox. For more information, see Create a discovery mailbox.

How do you know this worked?

Here are some ways to verify if you've successfully implemented custom management scopes for eDiscovery. When you verify, be sure that the user running the eDiscovery searches is a member of the role group that uses the custom management scope.

- Create an eDiscovery search, and select the distribution group that was used to create the custom management scope as the source of mailboxes to be searched. All mailboxes should be successfully searched.
- Create an eDiscovery search, and search the mailboxes of any users who aren't members of the distribution group that was used to create the custom management scope. The search should fail because the discovery manager can only search mailboxes for users who are members of the distribution group that was used to create the custom management scope. In this case, an error such as "Unable to search mailbox *<name of mailbox>* because the current user does not have permissions to access the mailbox" will be returned.
- Create an eDiscovery search, and search the mailboxes of users who are members of the distribution group that was used to create the custom management scope. In the same search, include the mailboxes of users who aren't members. The search should partially succeed. The mailboxes of members of the distribution group used to create the custom management scope should be successfully searched. The search of mailboxes for users who aren't members of the group should fail.

More information

- Because distribution groups are used in this scenario to scope eDiscovery searches and not for message delivery, consider the following when you create and configure distribution groups for eDiscovery:
  - Create distribution groups with a closed membership so that members can be added to or removed from the group only by the group owners. If you're creating the group in the Shell, use the syntax `MemberJoinRestriction closed` and `MemberDepartRestriction closed`.
  - Enable group moderation so that any message sent to the group is first sent to the group moderators who can approve or reject the message accordingly. If you're creating the group in the Shell, use the syntax `ModerationEnabled $true`. If you're using the EAC, you can enable moderation after the group is created.
  - Hide the distribution group from the organization's shared address book. Use the EAC or the **Set-DistributionGroup** cmdlet after the group is created. If you're using the Shell, use the syntax `HiddenFromAddressListsEnabled $true`.

In the following example, the first command creates a distribution group with closed membership and moderation enabled. The second command hides the group from the shared address book.

```
New-DistributionGroup -Name "Vancouver Users eDiscovery
Scope" -Alias VancouverUserseDiscovery -
MemberJoinRestriction closed -MemberDepartRestriction
closed -ModerationEnabled $true

Set-DistributionGroup "Vancouver Users eDiscovery Scope" -
HiddenFromAddressListsEnabled $true
```

For more information about creating and managing distribution groups, see Manage Distribution

Groups.

- Though you can use only distribution group membership as the recipient filter for a custom management scope used for eDiscovery, you can use other recipient properties to add users to that distribution group. Here are some examples of using the **Get-Mailbox** and **Get-Recipient** cmdlets to return a specific group of users based on common user or mailbox attributes.

```
Get-Recipient -RecipientTypeDetails UserMailbox -ResultSize
unlimited -Filter 'Department -eq "HR"'

Get-Mailbox -RecipientTypeDetails UserMailbox -ResultSize
unlimited -Filter 'CustomAttribute15 -eq
"VancouverSubsidiary"'

Get-Recipient -RecipientTypeDetails UserMailbox -ResultSize
unlimited -Filter 'PostalCode -eq "98052"'

Get-Recipient -RecipientTypeDetails UserMailbox -ResultSize
unlimited -Filter 'StateOrProvince -eq "WA"'

Get-Mailbox -RecipientTypeDetails UserMailbox -ResultSize
unlimited -OrganizationalUnit
"namsr01a002.sdf.exchangelabs.com/Microsoft Exchange Hosted
Organizations/contoso.onmicrosoft.com"
```

- You can then use the examples from the previous bullet to create a variable that can be used with the **Add-DistributionGroupMember** cmdlet to add a group of users to a distribution group. In the following example, the first command creates a variable that contains all user mailboxes that have the value **Vancouver** for the *Department* property in their user account. The second command adds these users to the Vancouver Users distribution group.

```
$members = Get-Recipient -RecipientTypeDetails UserMailbox
-ResultSize unlimited -Filter 'Department -eq "Vancouver"'

$members | ForEach {Add-DistributionGroupMember "Ottawa
Users" -Member $_.Name}
```

- You can use the **Add-RoleGroupMember** cmdlet to add a member to an existing role group that's used to scope eDiscovery searches. For example, the following command adds the user admin@ottawa.contoso.com to the Ottawa Discovery Management role group.

```
Add-RoleGroupMember "Vancouver Discovery Management" -
Member paralegal@vancouver.contoso.com
```

You can also use the EAC to add members to a role group. For more information, see Add members to a role group.

Remove an In-Place eDiscovery search

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-14*

In Microsoft Exchange Server 2013, you can use In-Place eDiscovery to search mailbox content. You can remove an In-Place eDiscovery search at any time. When you remove an In-Place eDiscovery search, search results are removed from the Discovery mailbox.

Caution:
Deleting an In-Place eDiscovery search will remove any search results copied to a Discovery mailbox.

What do you need to know before you begin?

- Estimated time to completion: 2-5 minutes.
- To remove an In-Place eDiscovery search that has In-Place Hold enabled, you must first remove the In-Place Hold from the search. For details, see Create or remove an In-Place Hold.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to remove an In-Place eDiscovery search

1. Navigate to **Compliance management** > **In-place eDiscovery & hold**.
2. In the list view, select the In-Place eDiscovery search you want to remove, and then click **Delete** .

Use the Shell to remove an In-Place eDiscovery search

For an example of how to remove an In-Place eDiscovery search, see the “Examples” section in Remove-MailboxSearch.

How do you know this worked?

To verify that you have successfully removed an In-Place eDiscovery search, do one of the following:

- Use the EAC to verify that the search is no longer displayed in the list view of the **In-place eDiscovery & hold** tab.
- Use the **Get-MailboxSearch** cmdlet to retrieve In-Place eDiscovery searches. For an example of how to retrieve In-Place eDiscovery searches, see the "Examples" section in Get-MailboxSearch.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Search and delete messages

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-14*

You can use the **Search-Mailbox** cmdlet to search and delete messages from a mailbox.

To search and delete messages in one step, run the **Search-Mailbox** cmdlet with the *DeleteContent* switch. However, when you do this, you can't preview search results or generate a log of messages that will be returned by the search, and you may inadvertently delete messages that you didn't intend to. To preview a log of the messages found in the search before they're deleted, run the **Search-Mailbox** cmdlet with the *LogOnly* switch.

As an additional safeguard, you can first copy the messages to another mailbox by using the *TargetMailbox* and *TargetFolder* parameters. By doing this, you retain a copy of the deleted messages in case you need to access them again.

What do I need to know before I begin?

- Estimated time to complete: 10 minutes. The actual time may vary depending on the size of the mailbox and the search query.
- You need to be assigned the following management roles to search for and delete messages in users' mailboxes:
  - **Mailbox Search** This role allows you to search for messages across multiple mailboxes in your organization. Administrators aren't assigned this role by default. To assign yourself this role so that you can search mailboxes, add yourself as a member of the Discovery Management role group. See Add a user to the Discovery Management role group.

- **Mailbox Import Export** This role allows you to delete messages from a user's mailbox. By default, this role isn't assigned to any role group. To delete messages from users' mailboxes, you can add the Mailbox Import Export role to the Organization Management role group. For more information, see Add a role to a role group.

- If the mailbox from which you want to delete messages has single item recovery enabled, you must first disable the feature.
- If the mailbox from which you want to delete messages is placed on litigation hold, we recommend that you check with your records management or legal department before removing the hold and deleting the mailbox content. After you obtain approval, follow the steps listed in the topic Clean up the Recoverable Items folder.
- You can't use the Exchange admin center (EAC) to perform these procedures. You must use the Shell.

What do you want to do?

Search messages and log the search results

This example searches April Stewart's mailbox for messages that contain the phrase "Your bank statement" in the Subject field and logs the search results in the SearchAndDeleteLog folder of the administrator's mailbox. Messages aren't copied to or deleted from the target mailbox.

```
Search-Mailbox -Identity "April Stewart" -SearchQuery
"Subject:'Your bank statement'" -TargetMailbox
"administrator" -TargetFolder "SearchAndDeleteLog" -LogOnly
-LogLevel Full
```

For detailed syntax and parameter information, see Search-Mailbox.

Search and delete messages

This example searches April Stewart's mailbox for messages that contain the phrase "Your bank statement" in the Subject field and deletes the messages from the source mailbox without copying the search results to another folder. As previously explained, you need to be assigned the Mailbox Import Export management role to delete messages from a user's mailbox.

Caution:

When you use the **Search-Mailbox** cmdlet with the *DeleteContent* switch, messages are permanently deleted from the source mailbox. Before you permanently delete messages, we recommend that you either use the *LogOnly* switch to generate a log of the messages found in the search before they're deleted or copy the messages to another mailbox before deleting them from the source mailbox.

```
Search-Mailbox -Identity "April Stewart" -SearchQuery
```

```
"Subject:'Your bank statement'" -DeleteContent
```

This example searches April Stewart's mailbox for messages that contain the phrase "Your bank statement" in the Subject field, copies the search results to the folder AprilStewart-DeletedMessages in the mailbox BackupMailbox, and deletes the messages from April's mailbox.

```
Search-Mailbox -Identity "April Stewart" -SearchQuery
"Subject:'Your bank statement'" -TargetMailbox
"BackupMailbox" -TargetFolder "AprilStewart-
DeletedMessages" -LogLevel Full -DeleteContent
```

For detailed syntax and parameter information, see Search-Mailbox.

Reduce the size of a discovery mailbox in Exchange

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-30*

Have a discovery mailbox that's exceeded the 50 GB limit? You can fix this issue by creating new discovery mailboxes and copying the search results from the large discovery mailbox to the new ones.

Why would I want to do this?

In Exchange Server 2013 and Exchange Online, the maximum size of discovery mailboxes, which are used to store In-Place eDiscovery search results, is 50 GB. Prior to the current size limit, you were able to increase the storage quota to more than 50 GB, which resulted in having discovery mailboxes much larger than 50 GB. There are three issues with discovery mailboxes that are larger than 50 GB:

- They're not supported.
- They can't be migrated to Office 365.
- If they're discovery mailboxes in Exchange Server 2010, they can't be upgraded to Exchange Server 2013.

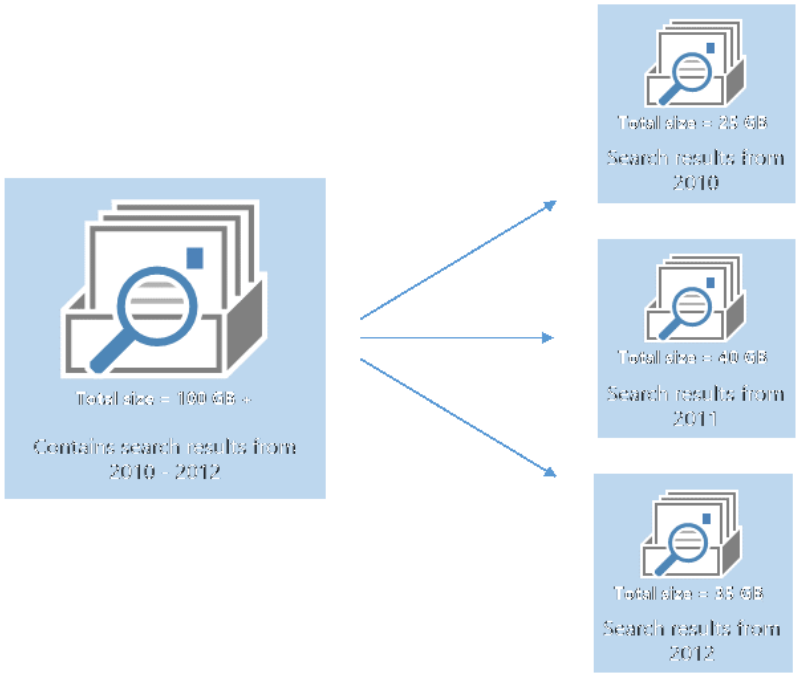
The process at a glance

Here's a quick look at what you'll need to do to reduce the size of a discovery mailbox that's

exceeded the 50 GB limit:

1. Create additional discovery mailboxes to distribute the search results to.
2. Copy the search results from the existing discovery mailbox to one or more of the new discovery mailboxes.
3. Delete eDiscovery searches from the original discovery mailbox to reduce its size.

The strategy presented here groups the search results from the original discovery mailbox into separate eDiscovery searches that are based on date ranges. This is a quick way to copy many search results to a new discovery mailbox. The following graphic illustrates this approach.



What do you need to know before you begin?

- Estimated time to complete this task: Time will vary based on the amount and size of the search results that will be copied to different discovery mailboxes.
- Run the following command to determine the size of the discovery mailboxes in your organization.

```
Get-Mailbox -RecipientTypeDetails DiscoveryMailbox | Get-MailboxStatistics | FL DisplayName,TotalItemSize
```

- Determine if you need to keep some or all of the search results from the discovery mailbox that's exceeded the 50 GB limit. Follow the steps in this topic to retain search results by copying them to a different discovery mailbox. If you don't need to keep the results of a specific eDiscovery search, you can delete the search, as explained in step 3. Deleting a search will delete the search results from the discovery mailbox.

- If you don't need any of the search results from a discovery mailbox that's exceeded the 50 GB limit, you can delete it. If this is the default discovery mailbox that was created when your Exchange organization was provisioned, you can re-create it. For more information, see Delete and re-create the default discovery mailbox in Exchange.
- For current legal cases, you might want to export the results of selected eDiscovery searches to .pst files. Doing this keeps the results from a specific search intact. In addition to the .pst files that contain the search results, a search results log (.csv file format) that contains an entry for each message returned in the search results is also exported. Each entry in this file identifies the source mailbox where the message is located. For more information, see Export eDiscovery search results to a PST file.

After you export search results to .pst files, you'll need to use Outlook if you want to import them to a new discovery mailbox.

Step 1: Create discovery mailboxes

The first step is to create additional discovery mailboxes so that you can copy the search results from the discovery mailbox that's exceeded the size limit. Based on the 50 GB size limit for discovery mailboxes, determine how many additional discovery mailboxes you'll need and create them. You'll then need to assign users or groups the necessary permissions to open these new discovery mailboxes.

1. Run the following command to create a new discovery mailbox.

```
New-Mailbox -Name <discovery mailbox name> -Discovery
```

2. Run the following command to assign a user or group permissions to open the discovery mailbox and view search results.

```
Add-MailboxPermission <discovery mailbox name> -User <name
of user or group> -AccessRights FullAccess -InheritanceType
all
```

Step 2: Copy search results to a discovery mailbox

The next step is to use the **New-MailboxSearch** cmdlet to copy the search results from the existing discovery mailbox to a new discovery mailbox that you created in the previous step. This procedure uses the *StartDate* and *EndDate* parameters to scope the search results into batches that are no larger than 50 GB. This may require some testing (by estimating the search results) to size the search results appropriately.

1. Run the following command to create a new eDiscovery search.

```
New-MailboxSearch -Name "Search results from 2010" -
SourceMailboxes "Discovery Search Mailbox" -StartDate
```

```
"01/01/2010" -EndDate "12/31/2010" -TargetMailbox
"Discovery Mailbox Backup 01" -EstimateOnly -
StatusMailRecipients admin@contoso.com
```

This example uses the following parameters:

- *Name* This parameter specifies the name of the new eDiscovery search. Because the search is scoped by sent and received dates, it's useful that the name of the search includes the date range.
- *SourceMailboxes* This parameter specifies the default discovery mailbox. You can also specify the name of another discovery mailbox that's exceeded the size limit.
- *StartDate* and *EndDate* These parameters specify the date range of the search results in the default discovery mailbox to include in the search results.

Note:

For dates, use the short date format, mm/dd/yyyy, even if the Regional Options settings on the local computer are configured with a different format, such as dd/mm/yyyy. For example, use **03/01/2014** to specify March 1, 2014.

- *TargetMailbox* This parameter specifies that search results should be copied to the discovery mailbox named "Discovery Mailbox Backup 01".
- *EstimateOnly* This switch specifies that only an estimate of the number of items that will be returned is provided when the search is started. If you don't include this switch, messages are copied to the target mailbox when the search is started. Using this switch lets you adjust the date ranges if necessary to increase or decrease the number of search results.
- *StatusMailRecipients* This parameter specifies that the status message should be sent to the specified recipient.

2. After the search is created, start it by using the Shell or the Exchange admin center (EAC).
   - **Using the Shell:** Run the following command to start the search created in the previous step. Because the *EstimateOnly* switch was included when the search was created, the search results won't be copied to the target discovery mailbox.

```
Start-MailboxSearch "Search results from 2010"
```

   - **Using the EAC:** Go to **Compliance management** > **In-Place eDiscovery & hold**. Select the search created in the previous step, click **Search**, and then click **Estimate search results**.
3. If necessary, adjust the date range to increase or decrease the amount of search results that are returned. If you change the date range, run the search again to get a new estimate of the results. Consider changing the name of the search to reflect the new date range.
4. When you're finished testing the search, use the Shell or the EAC to copy the search results to the target discovery mailbox.
   - **Using the Shell:** Run the following commands to copy the search results. You have to remove the *EstimateOnly* switch before you can copy the search results.

```
Set-MailboxSearch "Search results from 2010" -EstimateOnly
$false
```

```
Start-MailboxSearch "Search results from 2010"
```

- **Using the EAC:** Go to **Compliance management** > **In-Place eDiscovery & hold**. Select the search, click **Search** 🔍, and then click **Copy search results**.

For more information, see Copy eDiscovery search results to a discovery mailbox.

5. Repeat steps 1 through 4 to create new searches for additional date ranges. Include the date range in the name of the new search to indicate the range of the results. To make sure none of the discovery mailboxes exceeds the 50 GB limit, use different discovery mailboxes as the target mailbox.

Step 3: Delete eDiscovery searches

After you've copied search results from the original discovery mailbox to another discovery mailbox, you can delete the original eDiscovery searches. Deleting an eDiscovery search will delete the search results from the discovery mailbox where those search results are stored.

Before deleting a search, you can run the following command to identify the size of the search results that have been copied to a discovery mailbox for all searches in your organization.

```
Get-MailboxSearch | FL Name,TargetMailbox,ResultSizeCopied
```

You can use the Shell or the EAC to delete an eDiscovery search.

- **Using the Shell:** Run the following command.

```
Remove-MailboxSearch -Identity <name of search>
```

- **Using the EAC:** Go to **Compliance management** > **In-Place eDiscovery & hold**. Select the search that you want to delete, and then click **Delete** 🗑.

How do you know this worked?

After you've deleted the eDiscovery searches to remove the results from the discovery mailbox where they were stored, run the following command to display the size of a selected discovery mailbox.

```
Get-Mailbox <name of discovery mailbox> | Get-
MailboxStatistics | FL TotalItemSize
```

Delete and re-create the default discovery mailbox in Exchange

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-23*

You can use the Exchange Management Shell to delete the default discovery mailbox, re-create it, and then assign permissions to it.

Why would I want to do this?

In Exchange Server 2013 and Exchange Online, the maximum size of the default discovery mailbox is 50 GB. It's used to store In-Place eDiscovery search results. Before the size limit was changed, organizations could increase the storage quota to more than 50 GB. As a result, discovery mailboxes could grow to more than 50 GB. There are three issues with a default discovery mailbox that is larger than 50 GB:

- It's not supported.
- It can't be migrated to Office 365.
- If it's the default discovery mailbox in Exchange Server 2010, it can't be upgraded to Exchange Server 2013.

How you resolve this depends on whether you want to save the search results from a default discovery mailbox that's exceeded 50 GB.

Do you want to save the search results?	Do this
No	Follow the steps in this topic to delete, and then re-create the default discovery mailbox.
Yes	Follow the steps in Reduce the size of a discovery mailbox in Exchange.

Use the Shell to delete and re-create the default discovery mailbox

Note:
You can't use the Exchange admin center (EAC) because discovery mailboxes aren't displayed in the EAC.

1. Run the following command to delete the default discovery mailbox.

```
Remove-Mailbox "DiscoverySearchMailbox{D919BA05-46A6-415f-80AD-7E09334BB852}"
```

2. In the message asking you to confirm that you want to delete the mailbox and the corresponding

Active Directory user object, type **Y**, and then press Enter.

A new user object is created in Active Directory when you create the discovery mailbox in the next step.

3. Run the following command to re-create the default discovery mailbox.

```
New-Mailbox -Name "DiscoverySearchMailbox{D919BA05-46A6-415f-80AD-7E09334BB852}" -Alias "DiscoverySearchMailbox{D919BA05-46A6-415f-80AD-7E09334BB852}" -DisplayName "Discovery Search Mailbox" -Discovery
```

4. Run the following command to assign the Discovery Management role group permissions to open the default discovery mailbox and view search results.

```
Add-MailboxPermission "DiscoverySearchMailbox{D919BA05-46A6-415f-80AD-7E09334BB852}" -User "Discovery Management" -AccessRights FullAccess -InheritanceType all
```

Re-create the Discovery system mailbox

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-14*

In-Place eDiscovery uses a system mailbox to store In-Place eDiscovery search metadata. This Discovery system mailbox has the display name **SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9}**. Because system mailboxes aren't visible in the Exchange Administration Center (EAC) or in Exchange address lists, they are rarely deleted inadvertently.

However, if the Discovery system mailbox is deleted accidentally, discovery managers will be unable to perform In-Place eDiscovery searches or manage existing searches. In this case, to enable eDiscovery functionality, you must re-create the Discovery system mailbox.

What you should know before you begin

- Estimated time to complete: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Use the Shell to re-create the Discovery system mailbox

1. Delete the SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9} user account from Active Directory, if it exists. By default, Exchange Server 2013 Setup creates the mailbox in the Users container in Active Directory. For details about how to delete a user account from Active Directory, see Delete a User Account.
2. Prepare Active Directory by running Microsoft Exchange 2013 Setup with the **/PrepareAD** switch in the root domain of your Active Directory forest. For details, see Prepare Active Directory and domains.
3. Use the Shell to enable the Discovery system mailbox.

Note:
You can't use the EAC to enable the Discovery system mailbox.

This example enables the Discovery system mailbox. You must specify the fully qualified domain name (FQDN) of a global catalog server in the root domain of the Active Directory forest.

```
Enable-Mailbox -Arbitration -DomainController <FQDN of root global catalog server> -Identity "SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9}"
```

For detailed syntax and parameter information, see Enable-Mailbox.

How do I know this worked?

To verify that you have successfully re-created the Discovery system mailbox, use the **Get-Mailbox** cmdlet with the *Arbitration* switch to retrieve system mailboxes. View the results of the command to verify that the system mailbox `SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9}` has been re-created.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Using OAuth authentication to support eDiscovery in an Exchange hybrid deployment

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-07*

To successfully perform cross-premises eDiscovery searches in an Exchange 2013 hybrid organization, you will have to configure OAuth (Open Authorization) authentication between your Exchange on-premises and Exchange Online organizations so that you can use In-Place eDiscovery to search on-premises and cloud-based mailboxes. OAuth authentication supports the following eDiscovery scenarios in an Exchange hybrid deployment:

- Search on-premises mailboxes that use Exchange Online Archiving for cloud-based archive mailboxes.
- Search on-premises and cloud-based mailboxes in the same eDiscovery search.

For step-by-step instructions for configuring OAuth authentication to support eDiscovery, see Configure OAuth authentication between Exchange and Exchange Online organizations.

What is OAuth authentication?

OAuth authentication is a server-to-server authentication protocol that allows applications to authenticate to each other. With OAuth authentication, user credentials and passwords are not passed from one computer to another. Instead, authentication and authorization is based on the exchange of security tokens, which grant access to a specific set of resources for a specific amount of time.

OAuth authentication typically involves three parties: a single authorization server and the two realms that need to communicate with one another. Security tokens are issued by the authorization server (also known as a security token server) to the two realms that need to communicate; these tokens verify that communications originating from one realm should be trusted by the other realm. When using OAuth authentication between an on-premises Exchange organization and Exchange Online, the function of the authorization server is provided by Microsoft Azure Active Directory Access Control Services (ACS) in your Office 365 organization. For example, during a cross-premises eDiscovery search, Azure ACS issues tokens that verify that an administrator or compliance officer from the Exchange on-premises organization is able to access mailboxes in the Exchange Online organization, and vice-versa.

eDiscovery scenarios in a hybrid deployment

The follow table identifies the eDiscovery scenarios in an Exchange hybrid deployment that require OAuth authentication.

eDiscovery scenario	Requires OAuth authentication?
Search Exchange on-premises mailboxes and Exchange Online mailboxes in the same eDiscovery search initiated from the Exchange	Yes

on-premises organization. For example, searching all mailboxes in the organization in a single eDiscovery search.	
Search Exchange on-premises mailboxes that use Exchange Online Archiving for cloud-based archive mailboxes. When you use In-Place eDiscovery, both the primary and archive mailboxes are searched.	Yes
Search Exchange Online mailboxes from an eDiscovery search initiated from the Exchange on-premises organization by an administrator or compliance officer.	Yes
Search on-premises mailboxes using an eDiscovery search initiated from the Exchange on-premises organization by an administrator or compliance officer.	No **Note:** As previously discussed, OAuth authentication would be required if the on-premises mailboxes were configured with cloud-based archive mailboxes.
Search Exchange Online mailboxes from an eDiscovery search initiated from Exchange Online or the eDiscovery Center in SharePoint Online by an Office 365 tenant administrator or a compliance officer signed in to an Office 365 user account.	No

Configuring OAuth authentication to support eDiscovery

As previously stated, see Configure OAuth authentication between Exchange and Exchange Online organizations for instructions to configure OAuth authentication to support eDiscovery in an Exchange hybrid deployment.

If OAuth isn't configured for your Exchange hybrid deployment, you can't use eDiscovery to search Exchange on-premises and Exchange Online mailboxes in the same eDiscovery search. You will have to search on-premises mailboxes from an eDiscovery search initiated from your on-premises

organization. Similarly, you can only search Exchange Online mailboxes from an eDiscovery search initiated from your Exchange Online organization or by using the eDiscovery Center in SharePoint Online. Additionally, you won't be able to search primary on-premises mailboxes if their corresponding archive mailbox resides in Exchange Online or in an Exchange Online Archiving organization.

More information

- You can also use OAuth authentication to allow other applications, such as SharePoint 2013 and Lync Server 2013, to authenticate to Exchange 2013. For more information, see Configure OAuth authentication with SharePoint 2013 and Lync 2013.
- You can configure server-to-server authentication between Exchange 2013 and SharePoint 2013 so administrators and compliance officers can use the eDiscovery Center in SharePoint 2013 to search Exchange 2013 mailboxes. For more information, see Configure Exchange for SharePoint eDiscovery Center.
- You can configure an Exchange hybrid deployment using the Hybrid Configuration Wizard in Exchange 2013. For a customized, step-by-step hybrid deployment configuration checklist, see the Exchange Server Deployment Assistant.

Configure Exchange for SharePoint eDiscovery Center

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-09-06*

Microsoft Exchange Server 2013 includes features that work with Microsoft SharePoint Server 2013 and Microsoft Lync Server 2013, known as *partner applications*. To make sure these partner applications can access each other's resources, you need to configure server-to-server authentication.

This topic shows you how to configure server-to-server authentication between Exchange 2013 and SharePoint 2013 so users can use the eDiscovery Center in SharePoint 2013 to search Exchange Server 2013 mailbox content. To fully enable this functionality, you must complete additional steps in SharePoint 2013. For details, see Configure eDiscovery in SharePoint 2013 .

What do you need to know before you begin?

- Estimated time to complete this task: 30 minutes.
- Procedures in this topic require specific permissions. See each procedure for its permissions

information.

- It's supported to install Exchange 2013 and SharePoint 2013 in different domains or forests. A Windows trust relationship between Exchange and SharePoint forests isn't required, because in that circumstance, Exchange and SharePoint will rely on the OAuth 2.0 protocol to trust one another.
- The SharePoint 2013 site must be configured to use Secure Sockets Layer (SSL).
- The Exchange Web Services Managed API must be installed on every server that is running SharePoint 2013. Reset Internet Information Server (IIS) after installation.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Configure server-to-server authentication for Exchange 2013 on a server running SharePoint Server 2013

Run the following command to create Exchange 2013 as a trusted security token issuer in SharePoint 2013.

```
New-SPTrustedSecurityTokenIssuer -Name Exchange -
MetadataEndPoint https://<Exchange Server Name or FQDN>/
autodiscover/metadata/json/1
```

Step 2: Configure server-to-server authentication for SharePoint 2013 on a server running Exchange 2013

Perform this step on an Exchange 2013 server. You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.

Run this command to configure the SharePoint partner application.

```
cd c:\'Program Files'\Microsoft\'Exchange Server'\V15
\Scripts
.\Configure-EnterprisePartnerApplication.ps1 -
AuthMetadataUrl <path to SharePoint AuthMetadataUrl> -
```

```
ApplicationType SharePoint
```

Step 3: Add authorized users to the Discovery Management role group

Add users who need to perform an eDiscovery search using SharePoint 2013 to the Discovery Management role group in Exchange 2013. For details, see Add a user to the Discovery Management role group.

Caution:
Adding users to the Discovery Management role group allows them to use In-Place eDiscovery to search all Exchange 2013 mailboxes and access potentially sensitive email content in user mailboxes. By default, this permission isn't assigned to any user, including members of the Organization Management role group. Check with your organization's legal or HR departments before assigning this permission to any user.

Unsearchable items in Exchange eDiscovery

Exchange Server 2013 > Messaging policy and compliance > In-Place eDiscovery >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-01-29*

In In-Place eDiscovery for Exchange Server 2013 and Exchange Online, unsearchable items are mailbox items that can't be indexed by Exchange Search or that have been only partially indexed. An unsearchable item typically contains a file, which can't be indexed, attached to an email message. Here are a few reasons why files can't be indexed for search and are returned as an unsearchable item when attached to an email message:

- The file type is unsupported for indexing because a search filter (also known as an *IFilter*) to index the file format isn't installed.
- The file type is disabled for indexing.
- The file type is supported for indexing but an indexing error occurred for a specific file.
- A file is encrypted with non-Microsoft technologies.
- A file is password-protected.

For a successful eDiscovery search, your organization may be required to review unsearchable items. You can specify whether to include unsearchable items when you copy eDiscovery search results to a discovery mailbox or export results to a PST file.

File types not supported for search

Certain types of files, such as Bitmap or MP3 files, don't contain content that can be indexed. As a result, Exchange Search doesn't perform full-text indexing on these types of files. These types of files are considered as unsupported file types. There are also file types for which full-text indexing has been disabled, either by default or by an administrator. Unsupported and disabled file types are considered unsearchable items in eDiscovery searches. These types of files, which are typically attached to an email message, are included in the result set when you include unsearchable items when copying or exporting search results. For a list of supported and disabled file formats, see File formats indexed by Exchange Search. In Exchange Server 2013, administrators can disable indexing for a supported file format by using the Set-SearchDocumentFormat cmdlet. This cmdlet isn't available in Exchange Online.

To identify the unsearchable items in a specific mailbox, you can run the Get-FailedContentIndexDocuments cmdlet to get a list of items that would be copied or exported when you choose to include unsearchable items with the search results.

Messages with unsupported file types returned in search results

Not every message with an unsupported file attachment is automatically returned as an unsearchable item. That's because other file properties, such as the filename, are indexed and available to be searched. For example, a keyword search for "financial" will return a message with an unsupported file attachment if that keyword appears in the file name. If the keyword only appears in the body of the attached file, the message would be returned as an unsearchable item.

Similarly, messages with unsupported file attachments are included in search results when other properties of a mailbox item, which are indexed and searchable, meet the search criteria. Message properties that are indexed for search include sent and received dates, sender and recipient, the file name of an attachment, and text in the message body. So even though a message attachment may be unsearchable, the message will be included in the regular search results if the value of other message properties matches the search criteria. In fact, it's common that messages with unsearchable attachments are included in the regular search results.

For a list of email message properties indexed for search, see Message properties indexed by Exchange Search.

Including unsearchable items in the search results

Your organization may be required to identify and perform additional processing on unsearchable items to determine what they are, what they contain, and whether they're relevant. To include unsearchable items with the eDiscovery search results, you can use the unsearchable items option

when you copy or export search results. To include unsearchable items when using In-Place eDiscovery in Exchange or Exchange Online, select the **Include unsearchable items** option when copying search results to a discovery mailbox or exporting them to a PST file. To include unsearchable items when using the eDiscovery Center in SharePoint or SharePoint Online, select the **Include items that are encrypted or have an unrecognized format** option.

Keep the following in mind when copying or exporting unsearchable items:

- When you copy unsearchable items to a discovery mailbox, any unsearchable items are copied to a separate folder named **Unsearchable**, which is located under the folder that contains the search results. When you export search results and include unsearchable items, the unsearchable items are exported to a separate PST file.
- When you include unsearchable items in the search results, all unsearchable items in the mailboxes that are searched will be returned, regardless of the search criteria.
- If you choose to include all mailbox items in the search results or if a search query doesn't specify any keywords or only specifies a date range, unsearchable items may not be copied to the **Unsearchable** folder if you select the option to include unsearchable items. This is because all items, including any unsearchable items, will be automatically included in the regular search results.
- As previously stated, because message properties and metadata are indexed, a keyword search may return results if that keyword appears in the properties or metadata of a message with an unsearchable file attached to it. In this case, two copies of the same mailbox item will also be included in the search results. To prevent this type of duplication and only include one copy of the item in the regular search results, you can select the **Enable deduplication** option when you copy or export the search results.
- Including unsearchable items in the search results can also affect the estimated search results that are displayed. If you include unsearchable items when copying search results, the total estimated item count and the total estimated size will include the unsearchable items.

For more information about including unsearchable items in search results, see:

- Create an In-Place eDiscovery search
- Export eDiscovery search results to a PST file
- SharePoint: Export eDiscovery content and create reports

More information about unsearchable items

- Although a file type supported by Exchange Search and is full-text indexed, there can be indexing or search errors that will cause a file to be returned as an unsearchable item. For example, searching a very large Excel file might be partially successful, but will then fail as the size limit is exceeded. In this case, it's possible that the same file is returned with the search results and as an unsearchable item.
- Attached files encrypted with Microsoft technologies are indexed by Exchange Search and will be searched. Files encrypted with non-Microsoft technologies are returned as unsearchable.
- Email messages encrypted with S/MIME aren't indexed and are considered unsearchable items. This includes encrypted messages with or without file attachments.

- Messages protected using Information Rights Management (IRM) are indexed by Exchange Search and included in the search results if they match query parameters. For more information about IRM, see Information Rights Management.
- As previously stated, because message properties and metadata are indexed, a keyword search may return results if that keyword appears in the indexed metadata. However, that same keyword search may not return the same item if the keyword only appears in the content of an attached item with an unsupported file type. In this case, the item would be returned only as an unsearchable item.
- In eDiscovery in Exchange 2010, there is the concept of a *safelist*. These are file types that contain content that isn't searchable and so aren't indexed by Exchange Search; for example, Windows Media Video (.wmv) and Waveform Audio (.wav) files. Because these types of files don't contain searchable content, they aren't considered unsearchable items in Exchange 2010. Mailbox items containing these file types weren't returned as unsearchable items and weren't copied to a discovery mailbox.

There is no longer a safelist in Exchange 2013 or Exchange Online. File types are either enabled or disabled for indexing or they are unsupported. Disabled and unsupported file types are considered unsearchable items.

Messaging records management

Exchange Server 2013 > Messaging policy and compliance >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-20*

Users send and receive email every day. If left unmanaged, the volume of email generated and received each day can inundate users, impact user productivity, and expose your organization to risks. As a result, email lifecycle management is a critical component for most organizations.

Messaging records management (MRM) is the records management technology in Microsoft Exchange Server 2013 that helps organizations manage email lifecycle and reduce the legal risks associated with email. Deploying MRM can help your organization in several ways:

- **Meet business requirements** Depending on your organization's messaging policies, you may need to retain important email messages for a certain period. For example, a user's mailbox may contain critical messages related to business strategy, transactions, product development, or customer interactions.
- **Meet legal and regulatory requirements** Many organizations have a legal or regulatory requirement to store messages for a designated period and remove messages older than that period. Storing messages longer than necessary may increase your organization's legal or financial risks.
- **Increase user productivity** If left unmanaged, the ever-increasing volume of email in your users'

mailboxes can also impact their productivity. For example, although newsletter subscriptions and automated notifications may have informational value when they're received, users may not remove them after reading (often they're never read). Many of these types of messages don't have a retention value beyond a few days. Using MRM to remove such messages can help reduce information clutter in users' mailboxes, thereby increasing productivity.

- **Improve storage management** Due to expectations driven by free consumer email services, many users keep old messages for a long period or never remove them. Maintaining large mailboxes is increasingly becoming a standard practice, and users shouldn't be forced to change their work habits based on restrictive mailbox quotas. However, retaining messages beyond the period that's necessary for business, legal, or regulatory reasons also increases storage costs.

Looking for management tasks related to MRM? See Messaging Records Management Procedures.

MRM in Exchange 2013

In Exchange 2013 (and also in Exchange 2010), MRM is accomplished through the use of retention tags and retention policies. Retention tags are used to apply retention settings to an entire mailbox and default mailbox folders such as Inbox and Deleted Items. You can also create and deploy retention tags that Outlook 2010 and later and Outlook Web App users can use to apply to folders or individual messages. After they're created, you add retention tags to a retention policy and then apply the policy to users. The Managed Folder Assistant, a mailbox assistant that runs on Exchange 2013 Mailbox servers, processes mailboxes and applies retention settings in the user's retention policy. To learn more about retention policies, see Retention tags and retention policies.

When a message reaches its retention age specified in the applicable retention tag, the Managed Folder Assistant takes the retention action specified by the tag. Messages can then be deleted permanently or deleted with the ability to recover them. If an archive has been provisioned for the user, you can also use retention tags to move items to the user's In-Place Archive. To learn more about In-Place Archiving in Exchange 2013, see In-Place Archiving.

Note:

Managed folders, the MRM feature available in Exchange 2007 and deprecated in Exchange 2010, isn't available in Exchange 2013. You must port your managed folder policy settings to retention policies.

MRM strategies

You can use retention policies to enforce basic message retention for an entire mailbox or for specific default folders. Although there are several strategies for deploying MRM, here are some of the most common:

Remove all messages after a specified period. In this strategy, you implement a single MRM policy that removes all messages after a certain period. In this strategy, there's no classification of messages. You can implement this policy by creating a single default policy tag (DPT) for the mailbox. However, this doesn't ensure that messages are retained for the specified period. Users

can still delete messages before retention period is reached.

Remove messages based on folder location. In this strategy, you implement MRM policies based on email location. For example, you can specify that messages in the Inbox are retained for one year and messages in the Junk Email folder are retained for 60 days. You can implement this policy by using a combination of retention policy tags (RPTs) for each default folder you want to configure and a DPT for the entire mailbox. The DPT applies to all custom folders and all default folders that don't have an RPT applied.

Note:
In Exchange 2013, you can create RPTs for the Calendar and Tasks folders. If you don't want items in these folders or other default folders to expire, you can create a disabled retention tag for that default folder.

Allow users to classify messages. In this strategy, you implement MRM policies that include a baseline retention setting for all messages but allow users to classify messages based on business or regulatory requirements. In this case, users become an important part of your records management strategy - often they have the best understanding of a message's retention value.

In Exchange 2013, users can apply different retention settings to messages that need to be retained for a longer or shorter period. You can implement this policy using a combination of the following:

- A DPT for the mailbox
- Personal tags that users can apply to custom folders or individual messages
- (Optional) Additional RPTs to expire items in specific default folders

For example, you can use a retention policy with personal tags that have a shorter retention period (such as two days, one week, or one month), as well as personal tags that have a longer retention period (such as one, two, or five years). Users can apply personal tags with the shorter retention periods for items such as newsletter subscriptions that may lose their value within days of receiving them, and apply the tags with longer periods to preserve items that have a high business value. They can also automate the process by using Inbox rules in Outlook and Outlook Web App.

Retain messages for eDiscovery purposes. In this strategy, you implement MRM policies that remove messages from mailboxes after a specified period but also retain them in the Recoverable Items folder for In-Place eDiscovery purposes, even if the messages were deleted by the user or another process.

In Exchange 2013, you can meet this requirement by using a combination of retention policies and In-Place Hold. Retention policies remove messages from the mailbox after the specified period. A time-based In-Place Hold preserves messages that were deleted or modified before that period. For example, to retain messages for seven years, you can create a retention policy with a DPT that deletes messages in seven years and an In-Place Hold to hold messages for seven years. Messages that aren't removed by users will be deleted after seven years; messages deleted by users before the seven year period will be retained in the Recoverable Items folder for seven years. To learn more about this folder, see Recoverable Items folder.

Optionally, you can use RPTs and personal tags to allow users to clean up their mailboxes. However,

In-Place Hold continues to retain the deleted messages until the hold period expires.

Note:
A time-based In-Place Hold is similar to what was informally referred to as a *rolling legal hold* in Exchange 2010. Rolling legal hold was implemented by configuring the deleted item retention period for a mailbox database or individual mailbox. However, deleted item retention retains deleted and modified items based on the date deleted. In-Place Hold preserves items based on the date they're received or created. This ensures that messages are preserved for at least the specified period.

Move messages to archive mailboxes. In this strategy, you implement MRM policies that move items to the user's archive mailbox. An archive mailbox provides additional storage for users to maintain old and infrequently accessed content. Retention tags that move items are also known as *archive policies*. Within the same retention policy, you can combine a DPT and personal tags to move items, and a DPT, RPTs, and personal tags to delete items. To learn more about archiving policies, see In-Place Archiving.

In Exchange 2013, MRM provides the flexibility to implement the records management policy that best meets your organization's requirements. With a good understanding of MRM, In-Place Archiving, and In-Place Hold, you can help meet your goals of managing mailbox storage and meeting regulatory retention requirements.

For more information

Messaging records management terminology in Exchange 2013

Retention tags and retention policies

Messaging records management terminology in Exchange 2013

Exchange Server 2013 > Messaging policy and compliance > Messaging records management >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-30*

This topic defines the core components associated with messaging records management (MRM) in Microsoft Exchange Server 2013. MRM is a records management technology in Exchange 2013 that helps organizations reduce the risks associated with e-mail and other communications. MRM makes it easier to keep messages needed to comply with company policy, government regulations, or legal needs, and to remove content that has no legal or business value.

default policy tag (DPT)

A DPT is a retention tag that applies to all items in a mailbox that don't already have a retention tag applied. You can have only one DPT in a retention policy.

filer user or filer

A user who regularly files mailbox items into folders.

Compare to piler user or piler.

journaling

The ability to record communications, including e-mail communications, in an organization for use in the organization's e-mail retention or archival strategy. In MRM, journaling commonly refers to the process of sending a journal report about messages in a managed folder to the specified SMTP address. This process is performed by the Managed Folder Assistant.

managed content settings

The retention information created for managed folders, the MRM feature available in Exchange Server 2007 and Exchange Server 2010. A managed folder can have multiple content settings for different message types such as e-mail messages, voice mail, and calendar items. Message retention settings defined in content settings for a managed folder apply to messages in that managed folder.

managed folder

Used for the MRM functionality in Exchange Server 2007 and Exchange Server 2010, a managed folder is an Active Directory object that represents a mailbox folder in order to apply managed content settings to it. In a mailbox, a managed folder is a folder to which managed content settings have been applied.

Managed Folder Assistant

One of the Microsoft Exchange Mailbox Assistants in Exchange 2013. The Managed Folder Assistant is responsible for archiving, message expiration, and compliance. It processes mailboxes and applies retention policies.

managed folder mailbox policy

A logical grouping of managed folders. In Exchange 2010 and Exchange 2007, when a managed folder mailbox policy is applied to a user's mailbox, all managed folders linked to the policy are deployed in a single operation.

personal tag

A personal tag is a retention tag available to Outlook Web App and Outlook 2010 and later users for applying retention settings to custom folders and individual items, such as e-mail messages.

piler user or piler

A user who doesn't file mailbox items regularly. Piler users tend to have a large Inbox and rely on search to find messages.

Compare to filer user or filer.

policy

See *retention policy* or *managed folder mailbox policy*.

retention policy tag (RPT)

A RPT is a retention tag that's applied to default folders such as Inbox and Deleted Items.

retention policy

A retention policy is logical grouping of retention tags. When a retention policy is applied to a user's mailbox, all retention tags linked to the policy are deployed in a single operation.

retention tag

Retention tags are used to apply retention settings to messages and folders in user mailboxes. There are three types of retention tags:
- Default policy tags (DPTs)
- Retention policy tags (RPTs)
- Personal tags

Retention tags and retention policies

Exchange Server 2013 > Messaging policy and compliance > Messaging records management >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-09-03*

In Microsoft Exchange Server 2013 and Exchange Online, Messaging records management (MRM) helps organizations to manage email lifecycle and reduce legal risks associated with e-mail and other communications. MRM makes it easier to keep messages needed to comply with company policy, government regulations, or legal needs, and to remove content that has no legal or business value.

Looking for management tasks related to MRM? See Messaging Records Management Procedures.

Contents

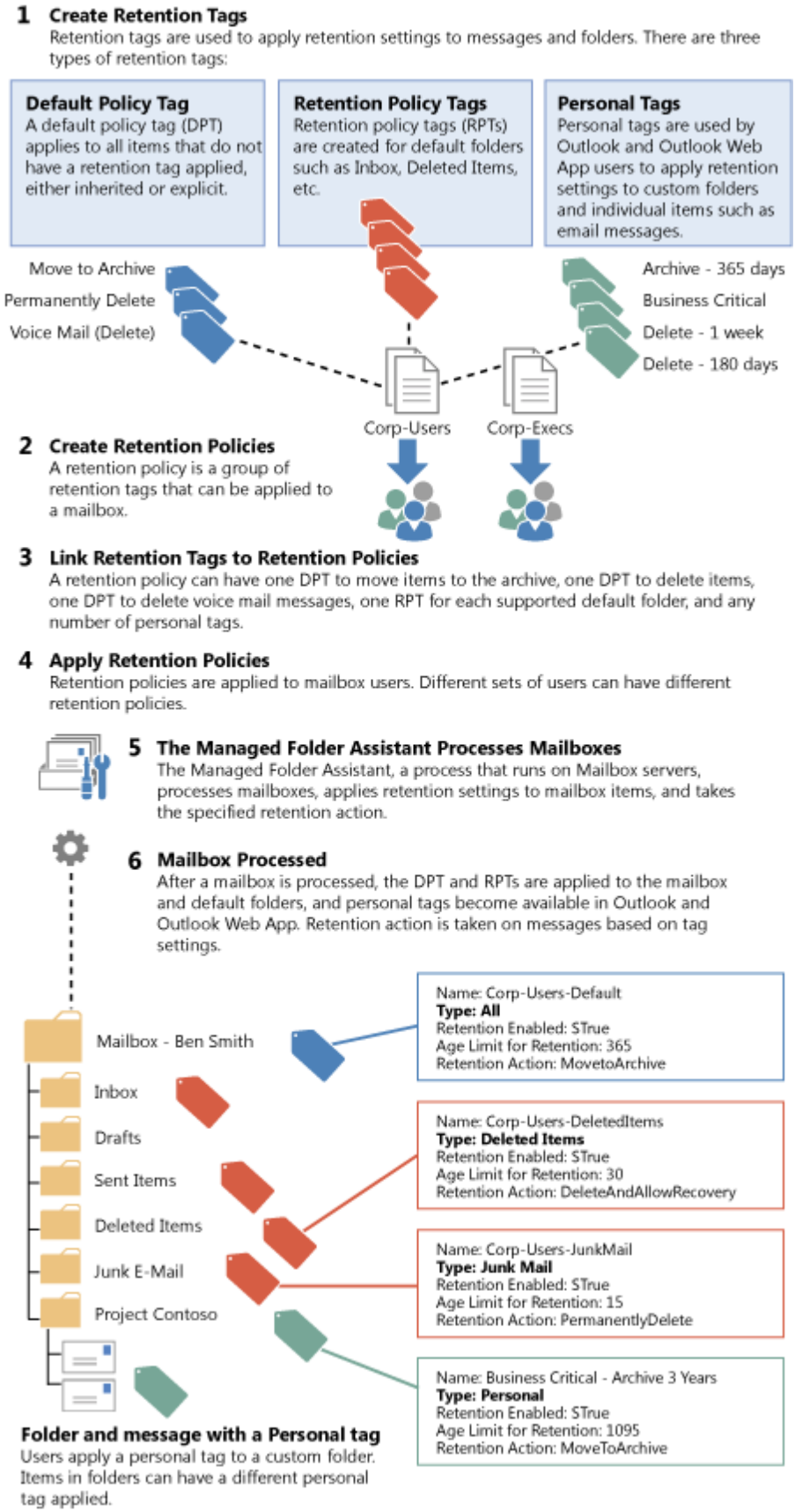
Messaging Records Management strategy

MRM in Exchange 2013 and Exchange Online is accomplished by using *retention tags* and *retention policies*. Before discussing the details about each of these retention features, it's important to learn how the features are used in the overall MRM strategy. This strategy is based on:

- Assigning *retention policy tags* (RPTs) to default folders, such as the Inbox and Deleted Items.
- Applying *default policy tags* (DPTs) to mailboxes to manage the retention of all untagged items.
- Allowing the user to assign *personal tags* to custom folders and individual items.
- Separating MRM functionality from users' Inbox management and filing habits. Users aren't required to file messages in managed folders based on retention requirements. Individual messages can have a different retention tag than the one applied to the folder in which they're located.

The following figure illustrates the tasks involved in implementing this strategy.

Messaging records management strategy



Return to top

Requirements

The following table shows requirements for the Mailbox server and client application.

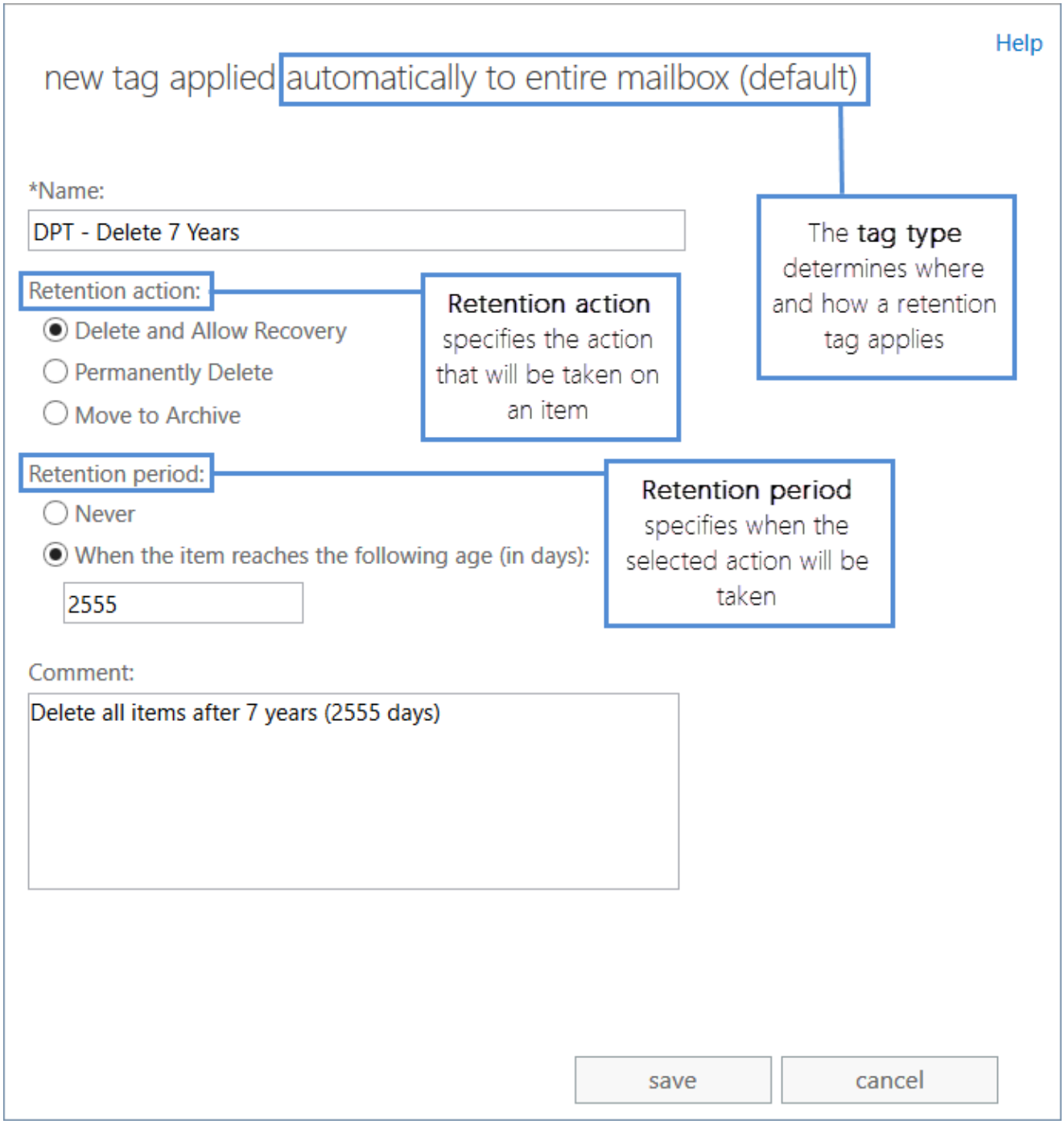
Retention tag and retention policy requirements

Location	Requirement
Mailbox server	Exchange 2010 or later is required.
Client application (to view retention tags and apply personal tags)	Only Microsoft Outlook 2010 and later, and Microsoft Office Outlook Web App users apply personal tags and view the retention tags applied to their mailbox folders or items. **Note:** Because retention policies are processed on Mailbox servers, they're independent of the Outlook version used by the users in your organization. You can still apply retention policies to user mailboxes running Microsoft Office Outlook 2007 and earlier. In these cases, RPTs in the policy apply to the default folders in their mailbox, and the DPT applies to untagged mailbox items.

Return to top

Retention tags

As illustrated in the preceding figure, retention tags are used to apply retention settings to folders and individual items such as e-mail messages and voice mail. These settings specify how long a message remains in a mailbox and the action to be taken when the message reaches the specified retention age. When a message reaches its retention age, it's moved to the user's In-Place Archive or deleted.



Retention tags allow users to tag their own mailbox folders and individual items for retention. Users no longer have to file items in managed folders provisioned by an administrator based on message retention requirements.

Types of retention tags

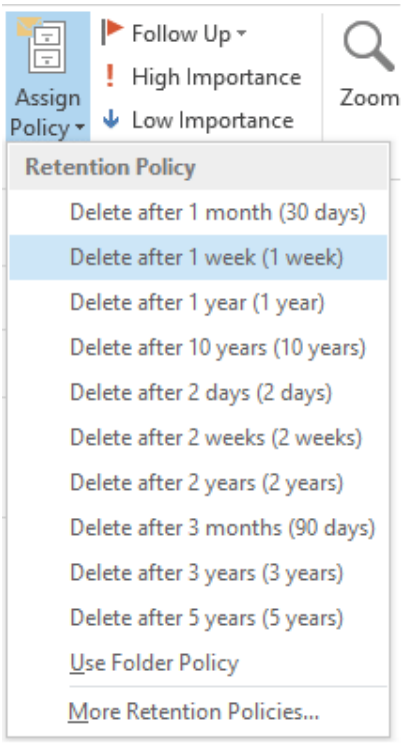
Retention tags are classified into the following three types based on who can apply them and where in a mailbox they can be applied.

Type of retention tag	Applied..	Applied by..	Action to take...	Details
Default Policy Tag (DPT)	Automatically to entire mailbox	Administrator	• Move to archive • Delete and allow	Users can't change DPTs

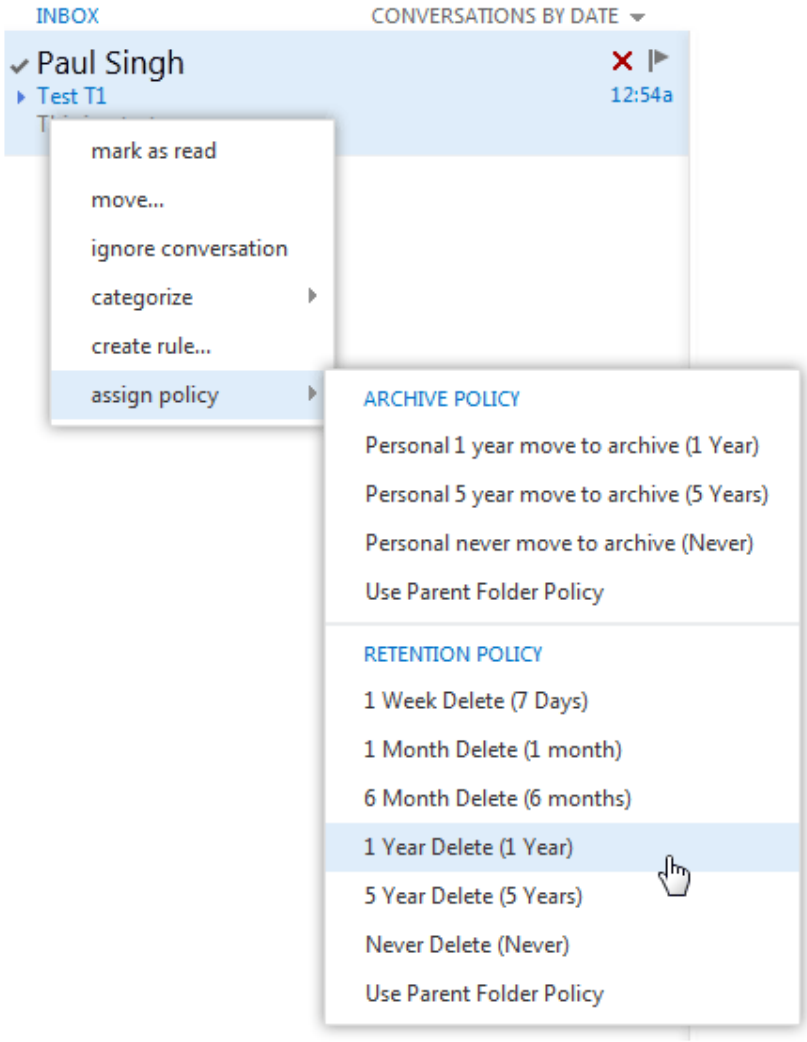
	A DPT applies to *untagged* items - items that don't have a retention tag applied directly or by inheritance from the folder.		recovery • Permanently delete	applied to a mailbox.
Retention Policy Tag (RPT)	Automatically to a default folder Default folders are folders created automatically in all mailboxes, for example: **Inbox**, **Deleted Items**, and **Sent Items**. See the list of supported default folders in Default folders that support Retention Policy Tags.	Administrator	• Delete and allow recovery • Permanently delete	Users can't change the RPT applied to a default folder.
Personal Tag	Manually to items and folders Users can automate tagging by using Inbox rules to either move a message	Users	• Move to archive • Delete and allow recovery • Permanently delete	Allow your users to determine how long an item should be retained. For example, the mailbox can have a DPT to delete

	to a folder that has a particular tag or to apply a personal tag to the message.			items in seven years, but a user can create an exception for an item or items in a folder by applying a personal tag that deletes an item such as newsletters and automated notifications in three days.

Personal tags Personal tags are available to Outlook 2010 and Outlook Web App users as part of their retention policy. Users can apply personal tags to folders they create or to individual items, even if those items already have a different tag applied. In Outlook 2010 and Outlook Web App, personal tags with the **Move to Archive** action appear as **Archive Policy**, and personal tags with the **Delete and Allow Recovery** or **Permanently Delete** actions appear as **Retention Policy**, as shown in the following figure.



Outlook 2013



OWA 2013

Messages that have a personal tag applied are always processed based on the personal tag's settings. Users can apply a personal tag to a message so that it's moved or deleted sooner or later than the settings specified in the DPT or RPTs applied to that user's mailbox. You can also create personal tags with retention disabled. This allows users to tag items so they're never moved to an archive or never expire.

Note:

Users can apply archive policies to default folders, user-created folders or subfolders, and individual items. Users can apply a retention policy to user-created folders or subfolders and individual items (including subfolders and items in a default folder), but not to default folders.

Users can also use the Exchange Administration Center (EAC) to select additional personal tags that aren't linked to their retention policy. The selected tags then become available in Outlook 2010 and Outlook Web App. To enable users to select additional tags from the EAC, you must add the MyRetentionPolicies role to the user's role assignment policy. To learn more about role assignment policies for users, see Understanding management role assignment policies. If you allow users to select additional personal tags, all personal tags in your Exchange organization become available to them.

Note:

Personal tags are a premium feature. Mailboxes with policies that contain these tags (or as a result of users adding the tags to their mailbox) require an Exchange Enterprise client access license (CAL).

Retention age limit and retention actions

When you enable a retention tag, you must specify a retention age for the tag. This age indicates the number of days to retain a message after it arrives in the user's mailbox.

The retention age for non-recurring items (such as email messages) is calculated differently than items that have an end date or recurring items (such as meetings and tasks). To learn how retention age is calculated for different types of items, see How retention age is calculated.

You can also create retention tags with retention disabled or disable tags after they're created. Because messages that have a disabled tag applied aren't processed by the Managed Folder Assistant, no retention action is taken. As a result, users can use a disabled personal tag as a **Never Move** tag or a **Never Delete** tag to override a DPT or RPT that would otherwise apply to the message.

When creating or configuring an RPT, you can select from one of the following actions to specify what retention action should be taken when a mailbox item reaches its retention age:

- **Move to Archive** This action moves a message to the user's archive mailbox. Tags that have this action applied are known as *archive tags*. Messages are moved to a folder in the archive mailbox that has the same name as the source folder in the user's primary mailbox. This allows users to easily locate messages in their archive mailbox. The **Move to Archive** action is available only for DPTs and personal tags. You can't create an RPT with the **Move to Archive** action. If the mailbox user doesn't have an archive mailbox, no action is taken. To learn more about archive mailboxes, see In-Place Archiving.
- **Delete and Allow Recovery** This action emulates the behavior when the Deleted Items folder is emptied. Tags that have this action applied are known as *deletion tags*. When this action occurs, and deleted item retention is configured for the mailbox database or the user, messages move to the Recoverable Items folder. The Recoverable Items folder (previously known as the dumpster) provides the user another chance to recover deleted messages. To do so, the user would access the **Recover Deleted Items** dialog box in Outlook 2010 or Outlook Web App. To learn more about recoverable items, see Recoverable Items folder.
- **Permanently Delete** This action permanently deletes a message. Like tags with the **Delete and Allow Recovery** action, tags that have this action applied are known as deletion tags. When this action is applied to a message, it's purged from the mailbox. This action is like a deleted message being removed from the Recoverable Items folder. After this happens, the user can no longer recover the message.
- **Mark as Past Retention Limit** This action isn't available in the Exchange Administration Center (EAC); you must use the Shell. This action marks a message as expired after it reaches its retention age. In Outlook 2010 or later, and Outlook Web App, expired items are displayed with the notification stating 'This item has expired' and 'This item will expire in 0 days'. In Outlook 2007,

items marked as expired are displayed by using strikethrough text.

Important:

If In-Place Hold or litigation hold is enabled for a mailbox user, permanently deleted items are retained in the Recoverable Items store until hold is removed. In-Place eDiscovery will still return permanently deleted messages in search results. To learn more, see In-Place Hold and In-Place eDiscovery.
If single item recovery is enabled for the mailbox, permanently deleted items are retained in the Recoverable Items store until the deleted item retention period for the mailbox database (or the deleted item retention period for the mailbox, if specified) is reached. To learn more, see Recoverable Items folder.

For details about how to create retention tags, see Create a Retention Policy.

Return to top

Retention policies

To apply one or more retention tags to a mailbox, you must add them to a retention policy and then apply the policy to mailboxes. A mailbox can't have more than one retention policy. Retention tags can be linked to or unlinked from a retention policy at any time, and the changes automatically take effect for all mailboxes that have the policy applied.

A retention policy can have the following retention tags:

- One RPT for each supported default folder

Note:

You can't link more than one RPT for a particular default folder (such as **Deleted Items**) to the same retention policy.

- One DPT with the **Move to Archive** action
- One DPT with the **Delete and Allow Recovery** or **Permanently Delete** actions
- One DPT for voice mail messages with the **Delete and Allow Recovery** or **Permanently Delete** actions
- Any number of personal tags

Although you can add any number of personal tags to a retention policy, having many personal tags with different retention settings can confuse users. We recommend linking no more than 10 personal tags to a retention policy.

Note:

Although a retention policy doesn't need to have any retention tags linked to it, we don't recommend using this scenario. If mailboxes with retention policies don't have retention tags linked to them, this may cause mailbox items to never expire.

A retention policy can contain both archive tags (tags that move items to the personal archive mailbox) and deletion tags (tags that delete items). A mailbox item can also have both types of tags applied. Archive mailboxes don't have a separate retention policy. The same retention policy is applied to the primary and archive mailbox.

When planning to create retention policies, you must consider whether they'll include both archive and deletion tags. As mentioned earlier, a retention policy can have one DPT that uses the **Move to Archive** action and one DPT that uses either the **Delete and Allow Recovery** or **Permanently Delete** action. The DPT with the **Move to Archive** action must have a lower retention age than the DPT with a deletion action. For example, you can use a DPT with the **Move to Archive** action to move items to the archive mailbox in two years, and a DPT with a deletion action to remove items from the mailbox in seven years. Items in both primary and archive mailboxes will be deleted after seven years.

For a list of management tasks related to retention policies, see Messaging Records Management Procedures.

Default Retention Policy

Exchange Setup creates the retention policy **Default MRM Policy**. The Default MRM Policy is applied automatically to new mailboxes in Exchange Online. In Exchange Server, the policy is applied automatically if you create an archive for the new user and don't specify a retention policy

You can modify tags included in the Default MRM Policy, for example by changing the retention age or retention action, disable a tag or modify the policy by adding or removing tags from it. The updated policy is applied to mailboxes the next time they're processed by the Managed Folder Assistant.

For more details, including a list of retention tags linked to the policy, see Default Retention Policy in Exchange Online and Exchange Server.

Note:
In Exchange 2010, the Default MRM Policy was named Default Archive and Retention Policy.

Return to top

Managed Folder Assistant

The Managed Folder Assistant, a mailbox assistant that runs on Mailbox servers, processes mailboxes that have a retention policy applied.

The Managed Folder Assistant applies the retention policy by inspecting items in the mailbox and determining whether they're subject to retention. It then stamps items subject to retention with the appropriate retention tags and takes the specified retention action on items past their retention age.

The Managed Folder Assistant is a throttle-based assistant. Throttle-based assistants are always running and don't need to be scheduled. The system resources they can consume are throttled. You can configure the Managed Folder Assistant to process all mailboxes on a Mailbox server within a certain period (known as a *work cycle*). Additionally, at a specified interval (known as the *work cycle checkpoint*), the assistant refreshes the list of mailboxes to be processed. During the refresh, the

assistant adds newly created or moved mailboxes to the queue. It also reprioritizes existing mailboxes that haven't been processed successfully due to failures and moves them higher in the queue so they can be processed during the same work cycle.

You can also use the Start-ManagedFolderAssistant cmdlet to manually trigger the assistant to process a specified mailbox. To learn more, see Configure the Managed Folder Assistant.

Note:
The Managed Folder Assistant doesn't take any action on messages that aren't subject to retention, specified by disabling the retention tag. You can also disable a retention tag to temporarily suspend items with that tag from being processed.

Moving items between folders

A mailbox item moved from one folder to another inherits any tags applied to the folder to which it's moved. If an item is moved to a folder that doesn't have a tag assigned, the DPT is applied to it. If the item has a tag explicitly assigned to it, the tag always takes precedence over any folder-level tags or the default tag.

Applying a retention tag to a folder in the archive

When the user applies a personal tag to a folder in the archive, if a folder with the same name exists in the primary mailbox and has a different tag, the tag on that folder in the archive changes to match the one in the primary mailbox. This is by design to avoid any confusion about items in a folder in the archive having a different expiry behavior than the same folder in the user's primary mailbox. For example, the user has a folder named Project Contoso in the primary mailbox with a *Delete – 3 years* tag and a Project Contoso folder also exists in the archive mailbox. If the user applies a *Delete – 1 year* personal tag to delete items in the folder after 1 year. When the mailbox is processed again, the folder reverts to the Delete – 3 Years tag.

Removing or deleting a retention tag from a retention policy

When a retention tag is removed from the retention policy applied to a mailbox, the tag is no longer available to the user and can't be applied to items in the mailbox.

Existing items that have been stamped with that tag continue to be processed by the Managed Folder Assistant based on those settings and any retention action specified in the tag is applied to those messages.

However, if you delete the tag, the tag definition stored in Active Directory is removed. This causes the Managed Folder Assistant to process all items in a mailbox and restamp the ones that have the removed tag applied. Depending on the number of mailboxes and messages, this process may

significantly consume resources on all Mailbox servers that contain mailboxes with retention policies that include the removed tag.

Important:

If a retention tag is removed from a retention policy, any existing mailbox items with the tag applied will continue to expire based on the tag's settings. To prevent the tag's settings from being applied to any items, you should delete the tag. Deleting a tag removes it from any retention policies in which it's included.

Disabling a retention tag

If you disable a retention tag, the Managed Folder Assistant ignores items that have that tag applied. Items that have a retention tag for which retention is disabled are either never moved or never deleted, depending on the specified retention action. Because these items are still considered tagged items, the DPT doesn't apply to them. For example, if you want to troubleshoot retention tag settings, you can temporarily disable a retention tag to stop the Managed Folder Assistant from processing messages with that tag.

Note:

The retention period for a disabled retention tag is displayed to the user as **Never**. If a user tags an item believing it will never be deleted, enabling the tag later may result in unintentional deletion of items the user didn't want to delete. The same is true for tags with the **Move to Archive** action.

Return to top

Retention hold

When users are temporarily away from work and don't have access to their e-mail, retention settings can be applied to new messages before they return to work or access their e-mail. Depending on the retention policy, messages may be deleted or moved to the user's personal archive. You can temporarily suspend retention policies from processing a mailbox for a specified period by placing the mailbox on retention hold. When you place a mailbox on retention hold, you can also specify a retention comment that informs the mailbox user (or another user authorized to access the mailbox) about the retention hold, including when the hold is scheduled to begin and end. Retention comments are displayed in supported Outlook clients. You can also localize the retention hold comment in the user's preferred language.

Note:

Placing a mailbox on retention hold doesn't affect how mailbox storage quotas are processed. Depending on the mailbox usage and applicable mailbox quotas, consider temporarily increasing the mailbox storage quota for users when they're on vacation or don't have access to e-mail for an extended period. For more information about mailbox storage quotas, see Configure storage quotas for a mailbox.

During long absences from work, users may accrue a large amount of e-mail. Depending on the

volume of e-mail and the length of absence, it may take these users several weeks to sort through their messages. In these cases, consider the additional time it may take the users to catch up on their mail before removing them from retention hold.

If your organization has never implemented MRM, and your users aren't familiar with its features, you can also use retention holds during the initial *warm up and training* phase of your MRM deployment. You can create and deploy retention policies and educate users about the policies without the risk of having items moved or deleted before users can tag them. A few days before the warm up and training period ends, you should remind users of the warm-up deadline. After the deadline, you can remove the retention hold from user mailboxes, allowing the Managed Folder Assistant to process mailbox items and take the specified retention action.

For details about how to place a mailbox on retention hold, see Place a mailbox on retention hold.

Return to top

Default Retention Policy in Exchange Online and Exchange Server

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-05*

Exchange creates the retention policy Default MRM Policy in your Exchange Online and on-premises Exchange organization. The policy is auomatically applied to new users in Exchange Online. In on-premises organizations, the policy is applied when you create an archive for the mailbox. You can change the retention policy applied to a user at any time.

You can modify tags included in the Default MRM Policy, for example by changing the retention age or retention actions, disable a tag, or modify the policy by adding or removing tags from it. The updated policy is applied to mailboxes the next time they're processed by the Managed Folder Assistant

Retention tags linked to the Default MRM Policy

The following table lists the default retention tags linked to the Default MRM Policy.

Name	Type	Retention age (days)	Retention action
Default 2 years move to archive	Default Policy Tag (DPT)	730	Move to Archive

Recoverable Items 14 days move to archive	Recoverable Items folder	14	Move to Archive
Personal 1 year move to archive	Personal tag	365	Move to Archive
Personal 5 year move to archive	Personal tag	1,825	Move to Archive
Personal never move to archive	Personal tag	Not applicable	Move to Archive
1 Week Delete	Personal tag	7	Delete and Allow Recovery
1 Month Delete	Personal tag	30	Delete and Allow Recovery
6 Month Delete	Personal tag	180	Delete and Allow Recovery
1 Year Delete	Personal tag	365	Delete and Allow Recovery
5 Year Delete	Personal tag	1,825	Delete and Allow Recovery
Never Delete	Personal tag	Not applicable	Delete and Allow Recovery

What you can do with the Default MRM Policy

You can...	In Exchange Online...	In Exchange Server...
Apply the Default MRM Policy automatically to new users	Yes, applied by default. No action is required.	Yes, applied by default if you also create an archive for the new user. If you create an archive for the

		user later, the policy is applied automatically only if the user doesn't have an existing Retention Policy.
Modify the retention age or retention action of a retention tag linked to the policy	Yes	Yes
Disable a retention tag linked to the policy	Yes	Yes
Add a retention tag to the policy	Yes	Yes
Remove a retention tag from the policy	Yes	Yes
Set another policy as the default retention policy to be applied automatically to new users	No	No

More Info

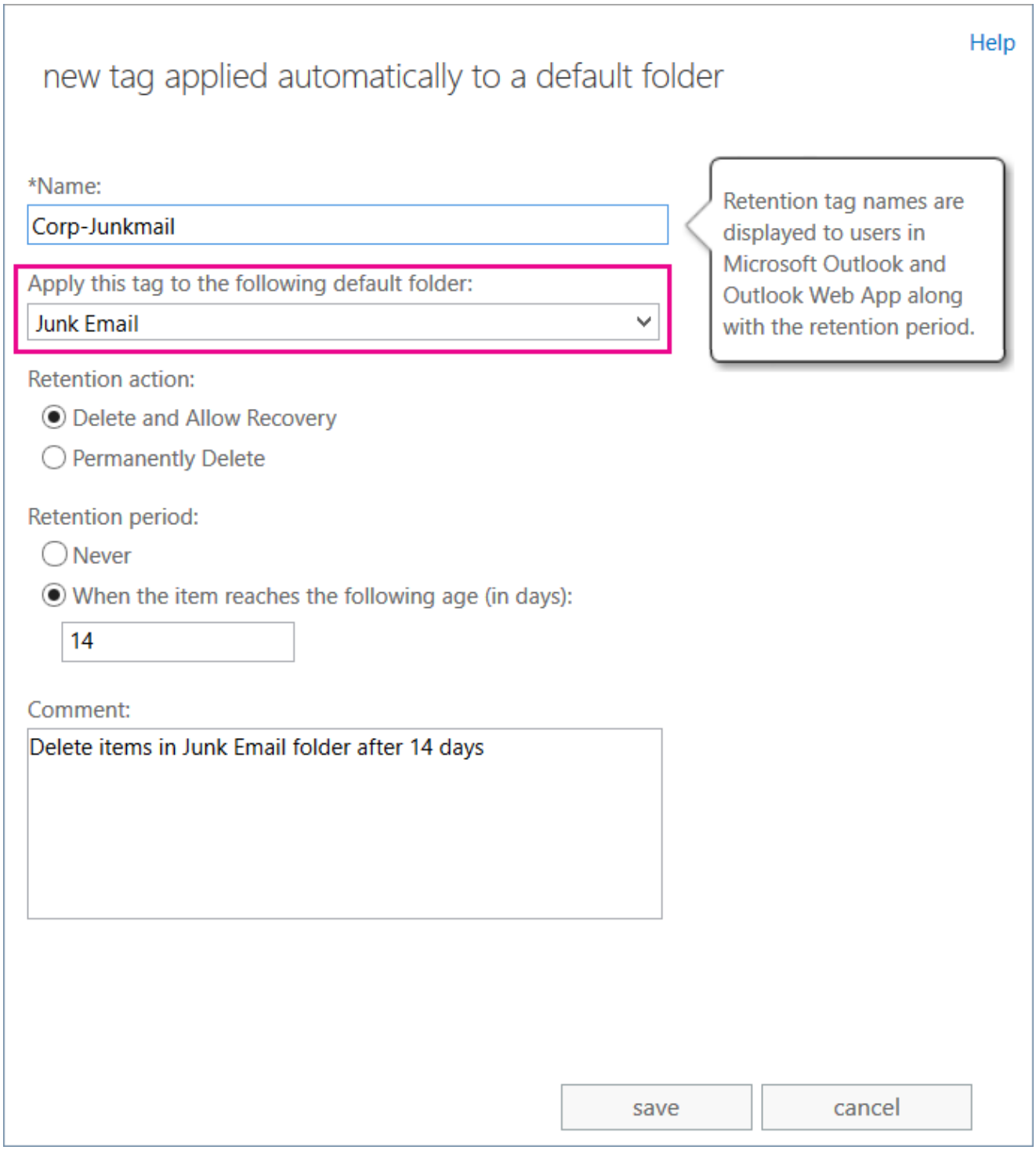
- A Retention Tag can be linked to more than one Retention Policy. For details about managing Retention tags and retention policies, see Messaging Records Management Procedures.
- The Default MRM Policy does not include a DPT to automatically delete items. If you want to automatically delete items after a specified period, create a DPT with the required delete action and add it to the policy. For details, see Create a Retention Policy and Add retention tags to or remove retention tags from a retention policy.
- Retention policies are applied to mailbox users. The same policy applies to the user's mailbox and archive.

Default folders that support Retention Policy Tags

Topic Last Modified: *2014-08-29*

You can use Retention tags and retention policies to manage email lifecycle. Retention Policies contain Retention Tags, which are settings you can use to specify when a message should be automatically moved to the archive or when it should be deleted.

A Retention Policy Tag (RPT) is a type of retention tag that you can apply to default folders in a mailbox, such as **Inbox** and **Deleted Items**.



Supported default folders

You can create RPTs for the default folders shown in the following table.

Folder name	Details
Calendar	This default folder is used to store meetings and appointments.
Conversation History	This folder is created by Microsoft Lync (previously Microsoft Office Communicator). Although not treated as a default folder by Outlook, it's treated as a special folder by Exchange and can have RPTs applied.
Deleted Items	This default folder is used to store items deleted from other folders in the mailbox. Outlook and Outlook Web App users can manually empty this folder. Users can also configure Outlook to empty the folder upon closing Outlook.
Drafts	This default folder is used to store draft messages that haven't been sent by the user. Outlook Web App also uses this folder to save messages that were sent by the user but not submitted to the Hub Transport server.
Inbox	This default folder is used to store messages delivered to a mailbox.
Journal	This default folder contains actions selected by the user. These actions are automatically recorded by Outlook and placed in a timeline view.
Junk E-mail	This default folder is used to save messages marked as junk e-mail by the content filter on an Exchange server or by the anti-spam filter in Outlook.
Notes	This folder contains notes created by users in Outlook. These notes are also visible in Outlook Web App.
Outbox	This default folder is used to temporarily store messages sent by the user until they're submitted to a Hub Transport server. A

	copy of sent messages is saved in the Sent Items default folder. Because messages usually remain in this folder for a brief period, it isn't necessary to create an RPT for this folder.	
RSS Feeds	This default folder contains RSS feeds.	
Recoverable Items	This is a hidden folder in the Non-IPM sub-tree. It contains the Deletions, Versions, Purges, and Audits sub-folders. Retention tags for this folder move items from the Recoverable Items folder in the user's primary mailbox to the Recoverable Items folder in the user's archive mailbox. You can assign only the **Move To Archive** retention action to tags for this folder. To learn more, see Recoverable Items folder.	
Sent Items	This default folder is used to store messages that have been submitted to a Hub Transport server.	
Sync Issues	This folder contains synchronization logs. To learn more, see Synchronization error folders.	
Tasks	This default folder is used to store tasks.	

More Info

- RPTs are retention tags for default folders. You can only select a delete action for RPTs – either **delete and allow recovery** or **permanently delete**.
- You can't create an RPT to move messages to the archive. To move old items to archive, you can create a *Default Policy Tag* (DPT), which applies to the entire mailbox, or *Personal Tags*, which are displayed in Outlook and Outlook Web App (OWA) as Archive Policies. Your users can apply them to folders or individual messages.
- You can't apply RPTs to the Contacts folder.
- You can only add one RPT for a particular default folder to a Retention Policy. For example, if a retention policy has an Inbox tag, you can't add another RPT of type Inbox to that retention policy.
- To learn how to create RPTs or other types of retention tags and add them to a retention policy, see Create a Retention Policy.
- In Exchange 2013 and Exchange Online, a DPT also applies to the **Calendar** and **Tasks** default folders. This may result in items being deleted or moved to the archive based on the DPT settings.

To prevent the DPT settings from deleting items in these folders , create RPTs with retention disabled. To prevent the DPT settings from moving items in a default folder, you can create a disabled Personal Tag with the move to archive action, add it to the retention policy, and then have users apply it to the default folder. For details, see Prevent archiving of items in a default folder in Exchange 2010.

How retention age is calculated

Messaging policy and compliance > Messaging records management > Retention tags and retention policies >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-06-16*

The Managed Folder Assistant (MFA) is one of many *mailbox assistant* processes that runs on mailbox servers. Its job is to process mailboxes that have a Retention Policy applied, add the Retention Tags included in the policy to the mailbox, and process items in the mailbox. If the items have a retention tag, the assistant tests the age of those items. If an item has exceeded its *retention age*, it takes the specified *retention action*. Retention actions include moving an item to the user's archive, deleting the item and allowing recovery, or deleting the item permanently.

See Retention tags and retention policies for more information.

Determining the age of different types of items

The retention age of mailbox items is calculated from the date of delivery or the date of creation for items such as drafts that are not delivered but created by the user. When the Managed Folder Assistant processes items in a mailbox, it stamps a start date and an expiration date for all items that have retention tags with the **Delete and Allow Recovery** or **Permanently Delete** retention action. Items that have an archive tag are also stamped with a move date.

Items in the Deleted Items folder and items which may have a start and end date, such as calendar items (meetings and appointments) and tasks, are handled differently as shown in this table.

If the item type is...	And the item is...	The retention age is calculated based on...
• Email message • Contact • Document • Fax • Journal item • Meeting request, response,	Not in the Deleted Items folder	• Delivery date or date of creation

or cancellation • Missed call		
• Email message • Contact • Document • Fax • Journal item • Meeting request, response, or cancellation • Missed call	In the Deleted Items folder	• Date of delivery or creation unless the item was deleted from a folder that does not have an inherited or implicit retention tag. • If an item is in a folder that doesn't have an inherited or implicit retention tag applied, the item isn't processed by the MFA and therefore doesn't have a start date stamped by it. When the user deletes such an item, and the MFA processes it for the first time in the Deleted Items folder, it stamps the current date as the start date.
Calendar	Not in the Deleted Items folder	• Non-recurring calendar items expire according to their end date. • Recurring calendar items expire according to the end date of their last occurrence. Recurring calendar items with no end date don't expire.
Calendar	In the Deleted Items folder	1. A calendar item expires according to its `message-received date`, if one exists. 2. If a calendar item doesn't have a `message-received date`, it expires according to its `message-creation date`. 3. If a calendar item has neither a `message-received date` nor a

		message-creation date, it doesn't expire.
Task	Not in the Deleted Items folder	• Non-recurring tasks: 1. A non-recurring task expires according to its message-received date, if one exists. 2. If a non-recurring task doesn't have a message-received date, it expires according to its message-creation date. 3. If a non-recurring task has neither a message-received date nor a message-creation date, it doesn't expire. • A recurring task expires according to the end date of its last occurrence. If a recurring task doesn't have an end date, it doesn't expire. • A regenerating task (which is a recurring task that regenerates a specified time after the preceding instance of the task is completed) doesn't expire.
Task	In the Deleted Items folder	1. A task expires according to its message-received date, if one exists. 2. If a task doesn't have a message-received date, it expires according to its message-creation date. 3. If a task has neither a message-received date nor a message-creation date, it doesn't expire.

Corrupted	Any folder	• Corrupted items are skipped by the Managed Folder Assistant and don't expire.

Examples

If the user..	The retention tags on folder...	The Managed Folder Assistant...
1. Receives a message in the Inbox on 01/26/2013. 2. Deletes the message on 2/27/2013.	• Inbox: Delete in 365 days • Deleted Items: Delete in 30 days	1. Processes the message in the Inbox on 1/26/2013, stamps it with a *start date* of 01/26/2013 and an *expiration date* of 01/26/2014. 2. Processes the message again in the Deleted Items folder on 2/27/2013. It recalculates the expiration date based on the same start date (01/26/2013). 3. Because the item is older than 30 days, it is expired immediately.
1. Receives a message in the Inbox on 01/26/2013. 2. Deletes the message on 2/27/2013.	• Inbox: None (inherited or implicit) • Deleted Items: Delete in 30 days	1. Processes the message in the Deleted Items folder on 02/27/2013 and determines the item doesn't have a start date. It stamps the current date as the start date, and 03/27/2013 as the expiration date. 2. The item is expired on 3/27/2013, which is 30 days after the user deleted or moved it to the Deleted Items

		folder.

More Info

- In Exchange Online, the Managed Folder Assistant may process a mailbox once in seven days. This may result in items being expired up to seven days after the expiration date stamped on the item.
- Items in mailboxes placed on Retention Hold are not removed until the hold is removed.
- If a mailbox is placed on In-Place Hold or Litigation Hold, expiring items are removed from the Inbox but preserved in the Recoverable Items folder until the mailbox is removed from In-Place Hold.
- In hybrid deployments, the same retention tags and retention policies must exist in your on-premises and Exchange Online organizations in order to consistently move and expire items across both organizations. See Export and import retention tags for more information.

Checklist: Deploying retention policies

Messaging policy and compliance > Messaging records management > Retention tags and retention policies >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-17*

Use this checklist to deploy retention policies in your Microsoft Exchange Server 2013 organization. Before you start working with this checklist, make sure you're familiar with the concepts in the following topics:

- Messaging records management
- Retention tags and retention policies

Checklist for deploying retention policies

Done?	Tasks	Topic
	Assess messaging records management (MRM) requirements for different sets of users.	Messaging records management
	Determine which Microsoft Outlook client versions are in use.	**Get-LogonStatistics**

	Create retention tags.	Create a Retention Policy
	Create retention policies.	Create a Retention Policy
	Add retention tags to retention policies.	Add retention tags to or remove retention tags from a retention policy
	Place mailboxes on retention hold.	Place a mailbox on retention hold
	Apply a retention policy to a single mailbox for testing purposes.	Apply a retention policy to mailboxes
	Optional: Implement client blocking to block legacy Outlook clients.	Configure Outlook Client Blocking for Messaging Records Management
	Begin user communication and training activities. Include a deadline when retention policies will be processed, and items moved or deleted.	Not applicable
	Apply retention policy to additional mailboxes.	Apply a retention policy to mailboxes
	A few days in advance, remind users about the deadline.	Not applicable
	At the deadline, remove the retention hold from mailboxes.	Place a mailbox on retention hold

Messaging Records Management

Procedures

Messaging policy and compliance > Messaging records management > Retention tags and retention policies >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-14*

Create a Retention Policy

Add retention tags to or remove retention tags from a retention policy

Apply a retention policy to mailboxes

Configure the Managed Folder Assistant

Place a mailbox on retention hold

Create a Retention Policy

Messaging records management > Retention tags and retention policies > Messaging Records Management Procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-11*

In Exchange Server 2013, you can use retention policies to manage email lifecycle. Retention policies are applied by creating retention tags, adding them to a retention policy, and applying the policy to mailbox users.

For additional management tasks related to retention policies, see Messaging Records Management Procedures.

What do you need to know before you begin?

- Estimated time to complete this task: 30 minutes.
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- Mailboxes to which you apply retention policies must reside on Exchange Server 2010 or later servers.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

How do you do this?

Step 1: Create a retention tag

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Use the EAC to create a retention tag

1. Navigate to **Compliance management** > **Retention tags**, and then click **Add +**
2. Select one of the following options:
   - **Applied automatically to entire mailbox (default)** Select this option to create a default policy tag (DPT). You can use DPTs to create a default deletion policy and a default archive policy, which applies to all items in the mailbox.

Note:
You can't use the EAC to create a DPT to delete voice mail items. For details about how to create a DPT to delete voice mail items, see the Shell example below.

   - **Applied automatically to a specific folder** Select this option to create a retention policy tag (RPT) for a default folder such as **Inbox** or **Deleted Items**.

Note:
You can only create RPTs with the **Delete and allow recovery** or **Permanently delete** actions.

   - **Applied by users to items and folders (Personal)** Select this option to create personal tags. These tags allow Outlook and Outlook Web App users to apply archive or deletion settings to a message or folders that are different from the settings applied to the parent folder or the entire mailbox.
3. The **New retention tag** page title and options will vary depending on the type of tag you selected. Complete the following fields:
   - **Name** Enter a name for the retention tag. The tag name is for display purposes and doesn't have any impact on the folder or item a tag is applied to. Consider that the personal tags you provision for users are available in Outlook and Outlook Web App.
   - **Apply this tag to the following default folder** This option is available only if you selected **Applied automatically to a specific folder**.
   - **Retention action** Select one of the following actions to be taken after the item reaches its retention period:
     - **Delete and Allow Recovery** Select this action to delete items but allow users to recover them using the **Recover Deleted Items** option in Outlook or Outlook Web App. Items are retained until the deleted item retention period configured for the mailbox database or the mailbox user is reached.
     - **Permanently Delete** Select this option to permanently delete the item from the mailbox database.

Important:
Mailboxes or items subject to In-Place Hold or litigation hold will be retained and returned in

In-Place eDiscovery searches. To learn more, see In-Place Hold.

- **Move to Archive** This action is available only if you're creating a DPT or a personal tag. Select this action to move items to the user's In-Place Archive.

- **Retention period** Select one of the following options:
  - **Never** Select this option to specify that items should never be deleted or moved to the archive.
  - **When the item reaches the following age (in days)** Select this option and specify the number of days to retain items before they're moved or deleted. The retention age for all supported items except Calendar and Tasks is calculated from the date an item is received or created. Retention age for Calendar and Tasks items is calculated from the end date.
- **Comment** User this optional field to enter any administrative notes or comments. The field isn't displayed to users.

Use the Shell to create a retention tag

Use the **New-RetentionPolicyTag** cmdlet to create a retention tag. Different options available in the cmdlet allow you to create different types of retention tags. Use the *Type* parameter to create a DPT (`All`), RPT (specify a default folder type, such as `Inbox`) or a personal tag (`Personal`).

This example creates a DPT to delete all messages in the mailbox after 7 years (2,556 days).

```
New-RetentionPolicyTag -Name "DPT-Corp-Delete" -Type All -
AgeLimitForRetention 2556 -RetentionAction
DeleteAndAllowRecovery
```

This example creates a DPT to move all messages to the In-Place Archive in 2 years (730 days).

```
New-RetentionPolicyTag -Name "DPT-Corp-Move" -Type All -
AgeLimitForRetention 730 -RetentionAction MoveToArchive
```

This example creates a DPT to delete voice mail messages after 20 days.

```
New-RetentionPolicyTag -Name "DPT-Corp-Voicemail" -Type All
-MessageClass Voicemail -AgeLimitForRetention 20 -
RetentionAction DeleteAndAllowRecovery
```

This example creates a RPT to permanently delete messages in the Junk EMail folder after 30 days.

```
New-RetentionPolicyTag -Name "RPT-Corp-JunkMail" -Type
JunkEmail -AgeLimitForRetention 30 -RetentionAction
PermanentlyDelete
```

This example creates a personal tag to never delete a message.

```
New-RetentionPolicyTag -Name "Never Delete" -Type Personal
```

```
-RetentionAction DeleteAndAllowRecovery -RetentionEnabled
$false
```

Step 2: Create a retention policy

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Use the EAC to create a retention policy

1. Navigate to **Compliance management** > **Retention policies**, and then click **Add +**
2. In **New Retention Policy**, complete the following fields:
   - **Name** Enter a name for the retention policy.
   - **Retention tags** Click **Add +** to select the tags you want to add to this retention policy.

A retention policy can contain the following tags:

- One DPT with the **Move to Archive** action
- One DPT with the **Delete and Allow Recovery** or **Permanently Delete** actions
- One DPT for voice mail messages with the **Delete and Allow Recovery** or **Permanently Delete** actions
- One RPT per default folder such as **Inbox** to delete items
- Any number of personal tags

Note:

Although you can add any number of personal tags to a retention policy, having many personal tags with different retention settings can confuse users. We recommend linking no more than ten personal tags to a retention policy.

You can create a retention policy without adding any retention tags to it, but items in the mailbox to which the policy is applied won't be moved or deleted. You can also add and remove retention tags from a retention policy after it's created.

Use the Shell to create a retention policy

This example creates the retention policy RetentionPolicy-Corp and uses the *RetentionPolicyTagLinks* parameter to associate five tags to the policy.

```
New-RetentionPolicy "RetentionPolicy-Corp"  -
RetentionPolicyTagLinks "DPT-Corp-Delete","DPT-Corp-
Move","DPT-Corp-Voicemail","RPT-Corp-JunkMail","Never
Delete"
```

For detailed syntax and parameter information, see New-RetentionPolicy.

Step 3: Apply a retention policy to mailbox users

After you create a retention policy, you must apply it to mailbox users. You can apply different retention policies to different set of users. For detailed instructions, see Apply a retention policy to mailboxes.

How do you know this task worked?

After you create retention tags, add them to a retention policy, and apply the policy to a mailbox user, the next time the MRM mailbox assistant processes the mailbox, messages are moved or deleted based on settings you configured in the retention tags.

To verify that you have applied the retention policy, do the following:
1. Run the following Shell command to run the MRM assistant manually against a single mailbox.

```
Start-ManagedFolderAssistant -Identity <mailbox identity>
```

2. Log on to the mailbox using Outlook or Outlook Web App and verify that messages are deleted or moved to an archive in accordance with the policy configuration.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Add retention tags to or remove retention tags from a retention policy

Messaging records management > Retention tags and retention policies > Messaging Records Management Procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-14*

You can add retention tags to a retention policy when the policy is created or any time thereafter. For details about how to create a retention policy, including how to simultaneously add retention tags, see Create a Retention Policy.

A retention policy can contain the following retention tags:
- One or more retention policy tags (RPTs) for supported default folders
- One default policy tag (DPT) with the **Move to Archive** action
- One DPT with the **Delete and Allow Recovery** or the **Permanently Delete** action
- One DPT for voice mail
- Any number of personal tags

For additional management tasks related to messaging records management (MRM), see

Messaging Records Management Procedures.

What do you need to know before you begin?

- Estimated time to completion: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Recipients Permissions topic.
- Retention tags aren't applied to a mailbox until they're linked to a retention policy and the Managed Folder Assistant processes the mailbox. To learn more about the Managed Folder Assistant, see Configure the Managed Folder Assistant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to add retention tags to or remove retention tags from a retention policy

1. Navigate to **Compliance management** >**Retention policies**.
2. In the list view, select the retention policy to which you want to add retention tags and then click **Edit** .
3. In **Retention Policy**, use the following settings:
   - **Add +** Click this button to add a retention tag to the policy.
   - **Remove −** Select a tag from the list, and then click this button to remove the tag from the policy.

Use the Shell to add retention tags to or remove retention tags from a retention policy

This example adds the retention tags VPs-Default, VPs-Inbox, and VPs-DeletedItems to the retention policy RetPolicy-VPs, which doesn't already have retention tags linked to it.

Caution:

If the policy has retention tags linked to it, this command replaces the existing tags.

```
Set-RetentionPolicy -Identity "RetPolicy-VPs" -
```

```
RetentionPolicyTagLinks "VPs-Default","VPs-Inbox","VPs-
DeletedItems"
```

This example adds the retention tag VPs-DeletedItems to the retention policy RetPolicy-VPs, which already has other retention tags linked to it.

```
$TagList = (Get-RetentionPolicy "RetPolicy-
VPs").RetentionPolicyTagLinks
$TagList.Add((Get-RetentionPolicyTag 'VPs-
DeletedItems').DistinguishedName)
Set-RetentionPolicy "RetPolicy-VPs" -
RetentionPolicyTagLinks $TagList
```

This example removes the retention tag VPs-Inbox from the retention policy RetPolicy-VPs.

```
$TagList = (Get-RetentionPolicy "RetPolicy-
VPs").RetentionPolicyTagLinks
$TagList.Remove((Get-RetentionPolicyTag 'VPs-
Inbox').DistinguishedName)
Set-RetentionPolicy "RetPolicy-VPs" -
RetentionPolicyTagLinks $TagList
```

For detailed syntax and parameter information, see Set-RetentionPolicy and Get-RetentionPolicy.

How do you know this worked?

To verify that you have successfully added or removed a retention tag from a retention policy, use the Get-RetentionPolicy cmdlet to verify the *RetentionPolicyTagLinks* property.

This example use the **Get-RetentionPolicy** cmdlet to retrieve retention tags added to the Default MRM Policy and pipes them to the **Format-Table** cmdlet to output only the name property of each tag.

```
(Get-RetentionPolicy "Default MRM
Policy").RetentionPolicyTagLinks | Format-Table name
```

Apply a retention policy to mailboxes

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-25*

You can use retention policies to group one or more retention tags and apply them to mailboxes to enforce message retention settings. A mailbox can't have more than one retention policy.

Caution:

Messages are expired based on settings defined in the retention tags linked to the policy. These settings include actions such moving messages to the archive or permanently deleting them. Before applying a retention policy to one or more mailboxes, we recommended that you test the policy and inspect each retention tag associated with it.

For additional management tasks related to messaging records management (MRM), see Messaging Records Management Procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Applying retention policies" entry in the Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to apply a retention policy to a single mailbox

1. Navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox to which you want to apply the retention policy, and then click **Edit** .
3. In **User Mailbox**, click **Mailbox features**.
4. In the **Retention policy** list, select the policy you want to apply to the mailbox, and then click **Save**.

Use the EAC to apply a retention policy to multiple mailboxes

1. Navigate to **Recipients** > **Mailboxes**.
2. In the list view, use the Shift or Ctrl keys to select multiple mailboxes.

3. In the details pane, click **More options**.
4. Under **Retention Policy**, click **Update**.
5. In **Bulk Assign Retention Policy**, select the retention policy you want to apply to the mailboxes, and then click **Save**.

Use the Shell to apply a retention policy to a single mailbox

This example applies the retention policy RP-Finance to Bharat's mailbox.

```
Set-Mailbox "Bharat" -RetentionPolicy "RP-Finance"
```

For detailed syntax and parameter information, see Set-Mailbox.

Use the Shell to apply a retention policy to multiple mailboxes

This example applies the new retention policy New-Retention-Policy to all mailboxes that have the old policy Old-Retention-Policy.

```
$OldPolicy={Get-RetentionPolicy "Old-Retention-
Policy"}.distinguishedName
Get-Mailbox -Filter {RetentionPolicy -eq $OldPolicy} -
Resultsize Unlimited | Set-Mailbox -RetentionPolicy "New-
Retention-Policy"
```

This example applies the retention policy RetentionPolicy-Corp to all mailboxes in the Exchange organization.

```
Get-Mailbox -ResultSize unlimited | Set-Mailbox -
RetentionPolicy "RetentionPolicy-Corp"
```

This example applies the retention policy RetentionPolicy-Finance to all mailboxes in the Finance organizational unit.

```
Get-Mailbox -OrganizationalUnit "Finance" -ResultSize
Unlimited | Set-Mailbox -RetentionPolicy "RetentionPolicy-
Finance"
```

For detailed syntax and parameter information, see Get-Mailbox and Set-Mailbox.

How do you know this worked?

To verify that you have applied the retention policy, run the Get-Mailbox cmdlet to retrieve the retention policy for the mailbox or mailboxes.

This example retrieves the retention policy for Bharat's mailbox.

```
Get-Mailbox Bharat | Select RetentionPolicy
```

This command retrieves all mailboxes that have the retention policy RP-Finance applied.

```
Get-Mailbox -ResultSize unlimited | Where-Object
{$_.RetentionPolicy -eq "RP-Finance"} | Format-Table
Name,RetentionPolicy -Auto
```

Configure the Managed Folder Assistant

Messaging records management > Retention tags and retention policies > Messaging Records Management Procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-14*

The *Managed Folder Assistant* is a Microsoft Exchange Mailbox Assistant that applies message retention settings configured in retention policies.

For additional management tasks related to messaging records management (MRM), see Messaging Records Management Procedures.

What do you need to know before you begin?

- Time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.
- You can't use the Exchange Administration Center (EAC) to configure the Managed Folder Assistant. You must use the Shell
- In Exchange 2013, the Managed Folder Assistant is a throttle-based assistant. Throttle-based assistants are always running and don't need to be scheduled. The system resources they can consume are throttled. You can configure the Managed Folder Assistant to process all mailboxes

on a Mailbox server within a certain period (known as a *work cycle)*. The work cycle is set to one day by default.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to configure the Managed Folder Assistant

This example configures the Managed Folder Assistant to process all mailboxes within one day.

```
Set-MailboxServer MyMailboxServer -ManagedFolderWorkCycle 1
```

For detailed syntax and parameter information, see Set-MailboxServer.

How do I know this worked?

To verify that you have successfully configured the Managed Folder Assistant, use the Get-MailboxServer cmdlet to check the *ManagedFolderWorkCycle* parameter.

This command retrieves all Mailbox servers in the organization and outputs the Managed Folder Assistant's workcycle properties from each server in a table format. The *Auto* switch is used to automatically fit column width.

```
Get-MailboxServer | Format-Table
Name,ManagedFolderWorkCycle* -Auto
```

Use the Shell to start the Managed Folder Assistant

This example triggers the Managed Folder Assistant to immediately process Bharat Suneja's mailbox.

```
Start-ManagedFolderAssistant -Identity
bharat.suneja@contoso.com
```

For detailed syntax and parameter information, see Start-ManagedFolderAssistant.

Place a mailbox on retention hold

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-14*

Placing a mailbox on retention hold suspends the processing of a retention policy or managed folder mailbox policy for that mailbox. Retention hold is designed for situations such as a user being on vacation or away temporarily.

During retention hold, users can log on to their mailbox and change or delete items. When you perform a mailbox search, deleted items that are past the deleted item retention period aren't returned in search results. To make sure items changed or deleted by users are preserved in legal hold scenarios, you must place a mailbox on legal hold. For more information, see Create or remove an In-Place Hold.

You can also include retention comments for mailboxes you place on retention hold. The comments are displayed in supported versions of Microsoft Outlook.

For additional management tasks related to messaging records management (MRM), see Messaging Records Management Procedures.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.
- You can't use the Exchange Administration Center (EAC) to place a mailbox on retention hold. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to place a mailbox on retention hold

This example places Michael Allen's mailbox on retention hold.

```
Set-Mailbox "Michael Allen" -RetentionHoldEnabled $true
```

For detailed syntax and parameter information, see Set-Mailbox.

Use the Shell to remove retention hold for a mailbox

This example removes the retention hold from Michael Allen's mailbox.

```
Set-Mailbox "Michael Allen" -RetentionHoldEnabled $false
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To verify that you have successfully placed a mailbox on retention hold, use the Get-Mailbox cmdlet to retrieve the *RetentionHoldEnabled* property of the mailbox.

This command retrieves the *RetentionHoldEnabled* property for Michael Allen's mailbox.

```
Get-Mailbox "Michael Allen" | Select RetentionHoldEnabled
```

This command retrieves all mailboxes in the Exchange organization, filters the mailboxes that are placed on retention hold, and lists them along with the retention policy applied to each.

Important:
Because *RetentionHoldEnabled* isn't a filterable property in Exchange 2013, you can't use the *Filter* parameter with the **Get-Mailbox** cmdlet to filter mailboxes that are placed on retention hold on the server-side. This command retrieves a list of all mailboxes and filters on the client running the Shell session. In large environments with thousands of mailboxes, this command may take a long time to complete.

```
Get-Mailbox -ResultSize unlimited | Where-Object
{$_.RetentionHoldEnabled -eq $true} | Format-Table
Name,RetentionPolicy,RetentionHoldEnabled -Auto
```

Export and import retention tags

Messaging records management > Retention tags and retention policies > Messaging Records Management Procedures >

Applies to: *Exchange Online*

Topic Last Modified: *2013-02-21*

There are several scenarios in which you may want to export or import retention tags, including:

- Applying the same retention policies across all servers in a multi-forest Exchange organization

- Applying the same retention policies in a hybrid deployment where some mailboxes reside in your on-premises Exchange organization and some reside in Exchange Online.
- Applying retention policies in an Exchange Online Archiving scenario, where users with on-premises Exchange 2010 or later mailboxes have a cloud-based archive.

In these scenarios, the Managed Folder Assistant can correctly process an item that has a retention tag applied after the item or the mailbox is moved to another organization.

Caution:

To keep retention tags and retention policies synchronized between two organizations, every time you make changes to a retention tag or policy in the source organization, you must perform this procedure to export retention tags and policies from the source organization and import them in the destination organization.
You can't select specific retention tags or policies to export. The Export-RetentionTags.ps1 script exports all retention tags and policies from an organization.

For additional management tasks related to Messaging Records Management, see Messaging Records Management Procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging Records Management" entry in the Messaging policy and compliance permissions topic.
- You can't select specific retention tags or policies to export or import. The Export-RetentionTags.ps1 script exports all retention tags and policies from an organization. The Import-RetentionTags.ps1 script imports all retention tags and policies in the XML file being imported, replacing all existing retention tags and policies in an Exchange organization.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Export retention tags from an on-premises Exchange organization

1. Run this Exchange Management Shell command to change directory to the **Scripts** subdirectory in your Exchange installation path.

```
Cd "<Exchange Server installation path>\Scripts"
```

2. Run the Export-RetentionTags.ps1 script to export retention tags to an XML file.

Important:
If you're importing or exporting retention tags and retention policies to Exchange Online, you must connect your Windows PowerShell session to Exchange Online. For details, see **Connect to Exchange Online using remote PowerShell**.

```
Export-RetentionTags.ps1 "c:\docs
\ExportedRetentionTags.xml"
```

How do you know this worked?

To verify that you have successfully exported retention tags and retention policies, do the following:

1. Navigate to the path you specified in the command to export and verify that the XML file with the name you specified has been created.
2. Optionally, you can open the XML file in a text editor to review its contents.

Import retention tags to an Exchange organization

1. Run this Shell command to change the directory to the **Scripts** subdirectory in your Exchange installation path.

```
Cd "<Exchange Server installation path>\Scripts"
```

2. Run the Import-RetentionTags.ps1 script to import retention tags from a previously exported XML file.

Important:
If you're importing or exporting retention tags and retention policies to Exchange Online, you must connect your Windows PowerShell session to Exchange Online. For details, see **Connect to Exchange Online using remote PowerShell**.

Note:
When running this script against Exchange Online, you may be prompted to confirm that you want to run software from an untrusted publisher. Verify that the name of the publisher appears as `CN=Microsoft Corporation, OU=MOPR, O=Microsoft Corporation, L=Redmond, S=Washington, C=US`, and then click **R** to allow the script to be run once or **A** to always run.

```
Import-RetentionTags.ps1 "c:\docs
\ExportedRetentionTags.xml"
```

How do you know this worked?

To verify that you have successfully imported retention tags and retention policies, do the following:

1. In the EAC, navigate to **Compliance Management** > **Retention tags**, and verify that the

retention tags have been imported successfully. Navigate to **Compliance Management** > **Retention policies**, and verify that the retention policies have been imported successfully.

2. Use the **Get-RetentionPolicy** and **Get-RetentionPolicyTag** cmdlets to verify that the tags and policies have been created. For an example about how to retrieve retention tags and retention policies, see Examples in Get-RetentionPolicyTag and Get-RetentionPolicy.

Configure Outlook client blocking

Messaging records management > Retention tags and retention policies > Messaging Records Management Procedures >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-02-14*

In Exchange Server 2013, you can use retention policies or managed folders for messaging records management (MRM). Only users running Microsoft Outlook 2010 and later have access to all client features for retention policies. However, retention policies are applied on the Mailbox server by the Managed Folder Assistant, regardless of the Outlook client version used by the user. Older Outlook clients do not expose the MRM functionality of these features. For example, because Outlook 2007 does not support retention policies, users can't apply personal tags to items or folders.

You can block users who are running older versions of Outlook from accessing their Exchange mailboxes. You can also block access on a per-mailbox or on a per-Client Access server basis.

For other management tasks related to MRM, see Messaging Records Management Procedures.

MRM Feature Availability by Client Application and Version

The following table lists the MRM features available in various client applications and versions.

MRM features

Client application	Available MRM client features
Outlook 2013 and Outlook 2010	All
Outlook 2007	Managed folders
Outlook 2003 and older	Not supported
Other e-mail client software	None

The following table shows version numbers for Outlook.

Outlook versions

Outlook version	Version number
Outlook 2013	15
Outlook 2010	14
Outlook 2007	12
Outlook 2003	11
Outlook 2002	10
Outlook 2000	9
Outlook 98	8.5
Outlook 97	8

Note:

Before making any changes, note that hotfixes and service pack releases may affect the client version string. Be careful when you restrict client access because server-side Exchange components must also use MAPI to log on. Some components report their client version as the component name (such as SMTP or OLE DB), while others report the Exchange build number (such as 6.0.4712.0). For this reason, avoid restricting clients that have version numbers that start with 6.<*x.x.*>. For example, to prevent MAPI access completely, instead of specifying **0.0.0-6.5535.65535.65535**, specify the two ranges so that the server components can log on. For example, specify the following: **0.0.0-5.9.9; 7.0.0-**.

After you perform these procedures, be aware that when users are blocked from accessing their mailboxes, they will receive the following warning message.

> Your Exchange Server administrator has blocked the version of Outlook that you are using.
>
> Contact your administrator for assistance.

To bypass the warning that MRM features aren't supported for e-mail clients running versions of Outlook earlier than Outlook 2010, you can use the *ManagedFolderMailboxPolicyAllowed* parameter of the **New-Mailbox**, **Enable-Mailbox**, and **Set-Mailbox** cmdlets in the Shell. When a managed folder mailbox policy is assigned to a mailbox by using the *ManagedFolderMailboxPolicy* parameter, the warning appears by default unless you use the *ManagedFolderMailboxPolicyAllowed* parameter.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You can't use the Exchange admin center (EAC) to perform these procedures. You must use the Shell.

- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Block versions of Outlook on a per-mailbox basis

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "User mailboxes" entry in the Recipients Permissions topic.

This example blocks all Outlook versions earlier than 11.8010.8036.

```
Set-CASMailbox -Identity adam@contoso.com -
MAPIBlockOutlookVersions "-11.8010.8036"
```

This example restores access to a mailbox that's blocked by a version of Outlook.

```
Set-CASMailbox -Identity adam@contoso.com -
MAPIBlockOutlookVersions $null
```

For detailed syntax and parameter information, see Set-CASMailbox.

Block Outlook versions on a Client Access server

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "RPC Client Access settings" entry in the Clients and mobile devices permissions topic.

This example blocks Outlook clients prior to version 12.0.0 from accessing the mailbox on an Exchange 2010 or later Client Access server.

Important:
The value used for the *BlockedClientVersions* parameter in this command is an example. You must determine the correct client version numbers. You can use the **Get-LogonStatistics** cmdlet to retrieve the versions of MAPI clients that are connected to the mailbox database.

```
Set-RpcClientAccess -Server CAS01 -BlockedClientVersions
"0.0.0-5.65535.65535;7.0.0;8.02.4-11.65535.65535"
```

For detailed syntax and parameter definition, see Set-RpcClientAccess.

Migrate from managed folders

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-14*

In Microsoft Exchange Server 2013, messaging records management (MRM) is performed by using retention tags and retention policies. A retention policy is a group of retention tags that can be applied to a mailbox. For more details, see Retention tags and retention policies. Managed folders, the MRM technology introduced in Exchange Server 2007, aren't supported.

A mailbox that has a managed folder mailbox policy applied can be migrated to use a retention policy. To do so, you must create retention tags that are equivalent to the managed folders linked to the user's managed folder mailbox policy.

Important:

Before you migrate from managed folders to retention policies in your production environment, we recommend that you test the process in a test environment.

Tip:

You can place mailboxes on retention hold to halt processing of retention policies or managed folder mailbox policies. Placing mailboxes on retention hold can be helpful in migration scenarios to avoid deleting messages or moving them to archive until new policy settings have been tested on test mailboxes or a small number of production mailboxes. For details, see Place a mailbox on retention hold.

For other management tasks related to MRM, see Messaging Records Management Procedures.

Comparing retention tags to managed folders

Unlike managed folders, which require users to move items to a managed folder based on retention settings, retention tags can be applied to a folder or an individual item in the mailbox. This process has minimal impact on the user's workflow and email organization methods. When a folder has retention tags applied, all items in that folder inherit the retention settings. Users can further specify retention settings by applying different retention tags to individual items in that folder.

Managed folders support different managed content settings for a folder, each with a different message class (such as email items or calendar items). Retention tags don't require a separate managed content settings object because the retention settings are specified in the tag's properties. It isn't supported to create retention tags for particular message classes, with the exception of a default policy tag (DPT) for voicemail messages. Retention tags also don't allow you to use journaling performed by the Managed Folder Assistant.

Note:

Journal rules, which are used to send copies of messages with a journal report to a journaling mailbox, are enforced in the transport pipeline by the Journaling agent and are independent of MRM. For more details, see Journaling.

The following table compares the MRM functionality available when using retention tags or managed folders.

Retention tags vs. managed folders

Functionality	Retention tags	Managed folders
Specify retention settings for default folders (such as Inbox)	Use retention policy tags (RPTs)	Use managed default folders
Specify retention settings for entire mailbox	Use a default policy tag (DPT)	Use managed default folders
Use retention settings for custom folders	Use personal tags	Using managed custom folders
Require managed content settings	No (retention settings included in a retention tag)	Yes
Use retention settings for different message classes (such as e-mail messages, voice mail, or calendar items)	No	Yes
Support the Move To Archive action, which moves items to the user's archive mailbox	Yes	No
Support the Move To Managed Folder action	No	Yes
Allow journaling using the Managed Folder Assistant	No	Yes
Policy applied to user	Retention policy	Managed folder mailbox policy
Maximum number of policies	1	1

that can be applied to a mailbox user		
Processed by the Managed Folder Assistant	Yes	Yes
Client support	Microsoft Outlook 2010 and Office Outlook Web App	Outlook 2010, Office Outlook 2007, and Outlook Web App

What do you need to know before you begin?

- Estimated time to complete: 20 minutes.
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- You can't use the Exchange admin center (EAC) to create retention tags based on retention policies.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Migrate mailbox users from managed folders

The following are general steps for migrating users from this managed folder mailbox policy to a retention policy. Each step is detailed later in this topic:

1. Gather information about managed folder mailbox policies applied to all Exchange 2010 and Exchange 2007 mailboxes, managed folders in each policy and managed content settings for each managed folder. You can use the EMC or the Shell on an Exchange 2010 or Exchange 2007 server to obtain this information.
2. Create retention tags for the migration.
3. Create a retention policy and link the newly created retention tags to the policy.
4. Remove the managed folder mailbox policy and then apply the retention policy to user mailboxes.

Important:
After you apply the retention policy to a user and the Managed Folder Assistant runs, the managed folders in the user's mailbox become unmanaged.

For the following procedures, Contoso mailboxes have a managed folder mailbox policy applied containing the following managed folders.

Managed folders for Contoso

Managed folder	Managed content settings	Retention enabled	Retention age	Retention action
Corp-DeletedItems	CS-Corp-DeletedItems	Yes	30 days	Delete and Allow Recovery
Corp-SentItems	CS-Corp-SentItems	Yes	1,825 days	Move to Deleted Items
Corp-JunkMail	CS-Corp-JunkMail	Yes	30 days	Permanently Delete
Corp-EntireMailbox	CS-Corp-EntireMailbox	Yes	365 days	Move to Deleted Items
30 Days	CS-30Days	Yes	30 days	Move to Deleted Items
5 Years	CS-5Years	Yes	1,825 days	Move to Deleted Items
Never Expire	CS-NeverExpire	No	365 days	Not applicable

How do you do this?

Step 1: Create retention tags for the migration

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

There are two methods you can use for this step:

- **Create retention tags based on the managed folders and their corresponding managed content settings** With this method, you use the **New-RetentionPolicyTag** cmdlet with the *ManagedFolderToUpgrade* parameter. When you specify this parameter, the corresponding retention tag is automatically applied to the managed folder.

Important:
If the managed folder you want to port has multiple managed content settings for different message classes, only one retention tag is created, and the highest retention age of all the managed content settings is used as the retention age for the ported tag, irrespective of the message class of the managed content settings. For example, review the following managed content settings for the managed folder Corp-

DeletedItems.

- **Create retention tags by manually specifying the retention settings** With this method, you use the **New-RetentionPolicyTag** cmdlet without the *ManagedFolderToUpgrade* parameter. When you don't specify this parameter, any retention policy tags you add to the policy are applied to the default folders, and the default policy tag is applied to the entire mailbox. However, any personal tags you add to the policy aren't automatically applied to the managed folders.

Note:

If you are in a mixed environment with Exchange 2013 and Exchange 2010 servers, you can use the **Port Managed Folder** wizard in the Exchange Management Console (EMC) on an Exchange 2010 server to easily port managed folder and corresponding managed content setting to retention tags.

Create retention tags based on managed folders

This example creates retention tags based on the corresponding managed content settings shown in the Contoso managed folder mailbox policy.

```
New-RetentionPolicyTag Corp-DeletedItems -
ManagedFolderToUpgrade Corp-DeletedItems
New-RetentionPolicyTag Corp-SentItems -
ManagedFolderToUpgrade Corp-SentItems
New-RetentionPolicyTag Corp-JunkMail -
ManagedFolderToUpgrade Corp-JunkMail
New-RetentionPolicyTag Corp-EntireMailbox -
ManagedFolderToUpgrade Corp-EntireMailbox
New-RetentionPolicyTag 30Days -ManagedFolderToUpgrade
30Days
New-RetentionPolicyTag 5Years -ManagedFolderToUpgrade
5Years
New-RetentionPolicyTag NeverExpire -ManagedFolderToUpgrade
NeverExpire
```

For detailed syntax and parameter information, see New-RetentionPolicyTag.

Create retention tags manually

Note:

You can also use the EAC to create retention tags manually (not based on settings in managed folders). For details, see Create a Retention Policy.

This example creates retention tags based on the managed folders and corresponding managed content settings shown in the Contoso managed folder mailbox policy. The retention settings are specified manually without using the *ManagedFolderToUpgrade* parameter.

```
New-RetentionPolicyTag Corp-DeletedItems -Type DeletedItems
-RetentionEnabled $true -AgeLimitForRetention 30 -
RetentionAction DeleteAndAllowRecovery
New-RetentionPolicyTag Corp-SentItems -Type SentItems -
RetentionEnabled $true -AgeLimitforRetention 1825 -
RetentionAction MoveToDeletedItems
New-RetentionPolicyTag Corp-JunkMail -Type JunkMail -
RetentionEnabled $true -AgeLimitforRetention 30 -
RetentionAction PermanentlyDelete
New-RetentionPolicyTag Corp-EntireMailbox -Type All -
RetentionEnabled $true -AgeLimitForRetention 365 -
RetentionAction MoveToDeletedItems
New-RetentionPolicyTag 30Days -Type Personal -
RetentionEnabled $true -AgeLimitForRetention 30 -
RetentionAction MoveToDeletedItems
New-RetentionPolicyTag 5Years -Type Personal -
RetentionEnabled $true -AgeLimitForRetention 1825 -
RetentionAction MoveToDeletedItems
New-RetentionPolicyTag NeverExpire -Type Personal -
RetentionEnabled $false
```

For detailed syntax and parameter information, see New-RetentionPolicyTag.

Step 2: Create a retention policy

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Note:
You can also use the EAC to create a retention policy and add retention tags to the policy. For details, see Create a Retention Policy.

This example creates the retention policy RP-Corp and links the newly created retention tags to the policy.

```
New-RetentionPolicy RP-Corp -RetentionPolicyTagLinks Corp-
DeletedItems,Corp-SentItems,Corp-JunkMail,Corp-
EntireMailbox,30Days,NeverExpire
```

For detailed syntax and parameter information, see New-RetentionPolicy.

Step 3: Remove the managed folder mailbox policy from user mailboxes

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Applying retention policies" entry in the Messaging policy and compliance permissions topic.

This example removes the managed folder mailbox policy and any managed folders from Ken Kwok's mailbox. Managed folders that have any messages are not removed.

```
Set-Mailbox -Identity Kwok -RemoveManagedFolderAndPolicy
RP-Corp
```

Step 4: Apply the retention policy to user mailboxes

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Applying retention policies" entry in the Messaging policy and compliance permissions topic.

Note:

You can also use the EAC to apply a retention policy to users. For details, see Apply a retention policy to mailboxes.

This example applies the newly created retention policy RP-Corp to the mailbox user Ken Kwok.

```
Set-Mailbox -Identity Kwok -RetentionPolicy RP-Corp
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this task worked?

To verify that you have migrated from managed folders to retention policies, do the following:

- Generate a report of all user mailboxes and the retention policy applied to them.

This command retrieves the retention policy applied to all mailboxes in an organization, and their retention hold status.

```
Get-Mailbox -ResultSize unlimited -Filter {Name -NotLike
"DiscoverySearch*"} | Format-Table
Name,RetentionPolicy,RetentionHoldEnabled -Auto
```

- After the Managed Folder Assistant has processed a mailbox with a retention policy, use the Get-RetentionPolicyTag cmdlet to retrieve the retention tags provisioned in the user mailbox.

This command retrieves the retention tags actually applied to April Stewart's mailbox.

```
Get-RetentionPolicyTag -Mailbox astewart
```

Turn off or suspend messaging records management

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-02-14*

To meet individual, IT, or business requirements, you may need to turn off or temporarily suspend messaging records management (MRM) for an individual user or for a Mailbox server. Reasons you may need to turn off or suspend MRM include:

- If a mailbox user is away from the office or is otherwise unable to access e-mail, you can temporarily disable MRM for the mailbox by placing it on retention hold. When a mailbox is on retention hold, it's no longer processed by the Managed Folder Assistant. When the mailbox user returns or is able to access the mailbox again, you can remove the retention hold from the mailbox.
- If you need to test or troubleshoot performance issues, you can temporarily turn off MRM on that server by clearing the schedule for the Managed Folder Assistant.
- If you need to remove a retention tag from mailboxes (which have a retention policy with that tag applied), you can remove the tag from the policy.
- If you want a retention policy or a managed folder mailbox policy to no longer apply to a mailbox, you can remove the policy from the mailbox.
- If your organization decides not to use MRM features, you can turn off MRM permanently for the entire organization. If you later decide to deploy MRM, you have the ability to do so.

What do you need to know before you begin?

- Estimated time to complete: 1 minute
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Place mailboxes on retention hold

You can place mailboxes on retention hold to turn off MRM temporarily (for example when users are on vacation). This suspends the processing of retention policies for the mailbox until retention hold is disabled. This is different from placing mailboxes on In-Place Hold or litigation hold.

For details about how to place a mailbox on retention hold, see Place a mailbox on retention hold.

To learn more about In-Place Hold and litigation hold, see In-Place Hold.

Remove retention tags from mailboxes

To remove a retention tag from a mailbox, you unlink the tag from the retention policy. When you unlink a retention policy tag (RPT) for a default folder, the default mailbox tag applies to all items in that folder. When you unlink a personal tag, it's no longer available to the user. Tags applied to existing messages will continue to be processed unless you remove the tag from the Exchange organization.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

This Shell example unlinks the retention tag Delete - 3 Days from the retention policy Corp-Users.

```
$tags = (Get-RetentionPolicy "Corp-
Users").RetentionPolicyTagLinks
$tags -= "Deleted Items - 3 Days"
Set-RetentionPolicy "Corp-Users" -RetentionPolicyTagLinks
$tags
```

For detailed syntax and parameter information, see Get-RetentionPolicy and Set-RetentionPolicy.

Remove retention policies from mailboxes

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Apply retention policies" entry in the Messaging policy and compliance permissions topic.

You can stop a retention policy from applying to a mailbox by removing the policy from the mailbox user's properties.

This Shell example removes the retention policy from the mailbox jpeoples.

```
Set-Mailbox jpeoples -RetentionPolicy $null.
```

This Shell example removes the retention policy from all mailboxes in the Exchange organization.

```
Get-Mailbox -ResultSize unlimited -Filter {RetentionPolicy
-ne $null} | Set-Mailbox -RetentionPolicy $null
```

This Shell example removes the retention policy Corp-Finance from all mailbox users who have the policy applied.

```
Get-Mailbox -ResultSize unlimited -Filter {RetentionPolicy
-eq "Corp-Finance"} | Set-Mailbox -RetentionPolicy $null
```

For detailed syntax and parameter information, see Set-Mailbox and Get-Mailbox.

Turn off MRM permanently for an entire organization

To turn off MRM for an organization, delete all retention tags and retention policies except for the ArbitrationMailbox policy, which is created by Exchange Setup. After this is complete, retention policies aren't enforced.

Caution:

Retention policies also include Move to Archive tags, which move messages to the user's archive mailbox. If you remove a retention policy that has a Move to Archive tag, users who had the policy applied will no longer have messages moved to the archive by the Managed Folder Assistant.
To avoid this, remove only the Delete and Allow Recovery and Permanently Delete tags from your organization and keep the policies that have the Move to Archive tags applied.
Alternatively, users who have and archive enabled could manually move items to their archive mailbox using Outlook or Outlook Web App.
Before removing retention tags or retention policies, we recommend that you check the settings of the tags being removed. Don't delete tags with the Move to Archive retention action.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Note:

Include the *Whatlf* switch in the following commands to simulate the action taken by a command.

This example removes all delete tags from an Exchange organization except the Never Delete tag, which is used in the ArbitrationMailbox policy created by Exchange Setup.

```
Get-RetentionPolicyTag | ? {$_.RetentionAction -ne
```

```
"MoveToArchive" -and $_.Name -ne "Never Delete"} | Remove-RetentionPolicyTag
```

This example removes all retention tags except the Never Delete tag.

```
Get-RetentionPolicyTag | ? {$_.Name -ne "Never Delete"} | Remove-RetentionPolicyTag
```

This command removes the Corp-Users retention policy from an Exchange organization.

```
Remove-RetentionPolicy Corp-Users
```

For detailed syntax and parameter information, see the following topics:

- Get-RetentionPolicyTag
- Remove-RetentionPolicyTag
- Remove-RetentionPolicy

Monitoring messaging records management

Exchange Server 2013 > Messaging policy and compliance > Messaging records management >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2009-11-03*

View performance counters for messaging records management

Performance counters for messaging records management

Messaging records management errors and events

View performance counters for messaging records management

Messaging policy and compliance > Messaging records management > Monitoring messaging records management >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2009-12-09*

You can use Windows Reliability and Performance Monitor (Perfmon.exe) to select and view performance counters for messaging records management (MRM). By using performance counters, you can monitor the Managed Folder Assistant while it runs resource-intensive MRM processes.

For a list of performance counters for MRM, see Performance counters for messaging records management.

Looking for other tasks related to monitoring MRM? Check out Monitoring messaging records management.

Use Windows Reliability and Performance Monitor to view performance counters for MRM

To perform this procedure, the account you use must be delegated membership in the local Administrators group.

1. To start Windows Reliability and Performance Monitor, click **Start**, click **Run**, and then type **perfmon**.
2. In the console tree, navigate to **Monitoring Tools** > **Performance Monitor**.
3. Click the plus sign (+) button on the toolbar. The **Add Counters** dialog box appears.
4. From the **Select counter from computer** list, select one of the following options:
   - If you are performing this procedure on a local computer, select **<Local computer>**. This is the default selection.
   - If you are performing this procedure remotely, select the server you want to monitor.
5. In the list of performance counters, expand **MSExchange Assistants - Per Database** or the **MSExchange Managed Folder Assistant**.
6. Select the performance counters you want to monitor.
7. For performance counters under **MSExchange Assistants - Per Database**, to view the counters for all mailbox databases, in **Instances of selected object**, click **All instances**. Or, to specify one or more mailbox databases, select instances from the list.
8. To add the selected counters so that the counters appear in Windows Reliability and Performance Monitor, and to begin collecting performance data, click **Add**.

Performance counters for messaging records management

Messaging policy and compliance > Messaging records management > Monitoring messaging records management >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2010-11-23*

The performance counters in this topic monitor the Managed Folder Assistant as it implements messaging records management (MRM) for Microsoft Exchange Server 2010. Because running the Managed Folder Assistant is a resource-intensive process, you should run it only when your server can tolerate the additional load. You should also monitor server performance when the Managed Folder Assistant is running. In addition to the performance counters listed in this topic, you may also want to monitor additional performance counters that monitor items such as disk performance and CPU usage.

For more information about monitoring computers running MRM, see Monitoring messaging records management.

Performance Counters for MRM

The following table describes performance counters for MRM.

Performance counters, performance objects, and description

Performance counter	Performance object	Description
Average Mailbox Processing Time In Seconds	MSExchange Assistants	Counts the average processing time of mailboxes for time-based assistants.
Mailboxes Processed	MSExchange Assistants	Counts the number of mailboxes processed by time-based assistants since the service started.
Mailboxes processed/sec	MSExchange Assistants	Determines the rate of mailboxes processed by time-based assistants per second.
Items Deleted but Recoverable	MSExchange Managed Folder Assistant	Counts the number of items deleted by the Managed Folder Assistant since the start of the most recent schedule interval. (The items are still recoverable through the Recoverable Items

		folder.) The number includes items in the mailboxes scheduled for processing during the schedule interval and items in any mailboxes that you specified for processing. This counter is reset to zero at the start of each schedule interval.
Items Journaled	MSExchange Managed Folder Assistant	Counts the number of items journaled by the Managed Folder Assistant since the start of the most recent schedule interval. The number includes items in the mailboxes scheduled for processing during the current work cycle and items in any mailboxes you specified for processing. This counter is reset to zero at the start of each work cycle.
Items Marked as Past Retention Date	MSExchange Managed Folder Assistant	Counts the number of items marked as past their retention date by the Managed Folder Assistant since the start of the most recent schedule interval. The number includes items in mailboxes scheduled for processing during the schedule interval and items in any mailboxes you specified for processing. This counter is reset to zero at the start of each

		schedule interval.
Items Moved	MSExchange Managed Folder Assistant	Counts the number of items moved by the Managed Folder Assistant since the start of the most recent schedule interval. The number includes items in the mailboxes scheduled for processing during the schedule interval and items in any mailboxes you specified for processing. This counter is reset to zero at the start of each schedule interval.
Items Permanently Deleted	MSExchange Managed Folder Assistant	Counts the number of items permanently deleted by the Managed Folder Assistant since the beginning of the most recent schedule interval. The number includes items in the mailboxes scheduled for processing during the schedule interval and items in any mailboxes you specified for processing. This counter is reset to zero at the beginning of each schedule interval.
Items Subject to Retention Policy	MSExchange Managed Folder Assistant	Counts the number of items subject to retention policy by the Managed Folder Assistant since the start of the most recent schedule interval. The number includes items in the mailboxes scheduled for processing during

		the schedule interval and items in any mailboxes you specified for processing. This counter is reset to zero at the start of each schedule interval. This counter is the sum of the following four expiration-related counters: • Items Journaled • Items Marked as Past Retention Date • Items Moved • Items Permanently Deleted
TotalSizeItemsExpired - Size of Items subject to Retention Policy (In Bytes)	MSExchange Managed Folder Assistant	Indicates the total size of items expired by the Managed Folder Assistant (SoftDelete, HardDelete, MoveToArchive). The following items are included: • Messages subject to deletion or move to a managed custom folder by a managed folder mailbox policy • Messages subject to deletion or move to archive by the user's retention policy • Messages expired by dumpster policy • Messages cleaned up by system cleanup tags This counter is reset to zero at every work cycle checkpoint of the Managed Folder Assistant work cycle.

TotalSizeItemsSoftDeleted - Size of Items Deleted but Recoverable (In Bytes)	MSExchange Managed Folder Assistant	Indicates the total size of items soft deleted by the Managed Folder Assistant. The following items are included: • Messages soft deleted by a managed folder mailbox policy • Messages soft deleted by a retention policy This counter is reset to zero at every work cycle checkpoint of the Managed Folder Assistant work cycle.
TotalSizeItemsPermanentlyDeleted - Size of Items Permanently Deleted (In Bytes)	MSExchange Managed Folder Assistant	Indicates the total size of items soft deleted by the Managed Folder Assistant. The following items are included: • Messages hard deleted by a managed folder mailbox policy • Messages hard deleted by a retention policy • Messages hard deleted by the Recoverable Items policy This counter is reset to zero at every work cycle checkpoint of the Managed Folder Assistant work cycle.
TotalSizeItemsMoved - Size of Items Moved due to an Archive policy tag (In Bytes)	MSExchange Managed Folder Assistant	Indicates the total size of items moved to a folder or moved to archive by the Managed Folder Assistant. The following items are included:

		• Messages moved to a managed custom folder by a managed folder mailbox policy • Messages moved to the personal archive by a retention policy This counter is reset to zero at every work cycle checkpoint of the Managed Folder Assistant work cycle.
TotalItemsWithPersonalTag - Items stamped with Personal Tag (Expiry or Archive)	MSExchange Managed Folder Assistant	Indicates the number of times a user tags items with a personal tag. This includes both Deletion and Archive tags. For example: • An item is tagged with a personal tag. • An item with a personal tag is retagged with another personal tag. If a folder is tagged with a personal tag, the counter is incremented by the total number of items in the folder.
TotalItemsWithDefaultTag - Items stamped with Default Tag (Expiry or Archive)	MSExchange Managed Folder Assistant	Indicates the number of items assigned a default policy tag (DPT) based on a user action, for example, when a user selects a message with a personal tag and selects **Use folder policy**.

		If a new user is assigned a retention policy with a DPT, the counter is incremented by the number of items that will be assigned the DPT due to the retention policy. **Note:** If a user has a retention policy with a DPT, new messages that arrive through transport get a default tag, and this isn't tracked by this counter.
TotalItemsWithSystemCleanupTag - Items stamped with System Cleanup Tag	MSExchange Managed Folder Assistant	Indicates the number of items tagged with the system cleanup tag. This includes mailbox metadata items that aren't visible to users.
TotalItemsExpiredByDefaultExpiryTag - Items expired due to a default Expiry Tag	MSExchange Managed Folder Assistant	Indicates the number of items expired (soft or hard deleted) by the Managed Folder Assistant due to any non-personal (default or system) tag in a retention policy. This doesn't include the items expired by Recoverable Items clean up or system clean up.
TotalItemsExpiredByPersonalExpiryTag - Items expired due to a personal Expiry Tag	MSExchange Managed Folder Assistant	Indicates the number of items expired (soft or hard deleted) by the Managed Folder Assistant due to a personal tag in the retention policy.
TotalItemsMovedByDefaultArchiveTag - Items moved due	MSExchange Managed Folder Assistant	Indicates the number of items moved to the archive by the

to a default Archive Tag		Managed Folder Assistant due to any non-personal (default or system) archive tag in a retention policy. This doesn't include the items moved to the Recoverable Items folder in archive by Recoverable Items cleanup.
TotalItemsMovedByPersonalArchiveTag - Items Moved due to an Archive Tag	MSExchange Managed Folder Assistant	Indicates the number of items moved to the archive by the Managed Folder Assistant due to a personal archive tag in a retention policy.
TotalMovedDumpsterItems - Mailbox Dumpsters Moved Items	MSExchange Managed Folder Assistant	Indicates the number of items moved to the Recoverable Items folder in the archive by Recoverable Items cleanup.

Messaging records management errors and events

Messaging policy and compliance > Messaging records management > Monitoring messaging records management >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-02-15*

Messaging records management (MRM) generates events that you can view in Event Viewer. This allows you to troubleshoot and verify the performance of the Managed Folder Assistant. Event Viewer tracks the following kinds of events in the following order, based on importance:

1. Error events
2. Warning events
3. Informational events

MRM Errors and Events

The following tables provide lists of events that you can use to troubleshoot MRM. The logging types include the following:

- Events labeled as **LogAlways** are always logged individually.
- Events labeled as **LogPeriodic** are logged only once in any five-minute period, not every time they occur. This helps to prevent excessive log entries.

MRM events in the Managed Folder Assistant category

Event ID	Category	Event type	Logging	Value or description
10003	Managed Folder Assistant	Error	LogPeriodic	Could not get the server configuration object from Active Directory. *<Exception details>*. Check for domain controller network connectivity issues or incorrect DNS configuration.
10004	Managed Folder Assistant	Error	LogAlways	The retention policy for folder *<folder>* in mailbox *<mailbox>* will not be applied. The managed folder assistant is unable to process managed content setting *<content*

				setting> for the managed folder <*managed folder*>. The RetentionAction is MoveToFolder but a destination folder was not specified. Please specify a destination folder.
10005	Managed Folder Assistant	Error	LogAlways	Retention policy will not be applied to folder <*folder*> in mailbox <*mailbox*>. Unable to process Managed Content Setting <*content setting*> for the Managed Folder <*managed folder*>. The RetentionAction is MoveToFolder but the destination folder <*folder*> is the same as the source folder <*folder*>. Please specify a destination folder

				that is different from the source folder.
10009	Managed Folder Assistant	Error	LogAlways	The managed folder assistant skipped processing all databases on the local server because it could not read the audit log parameters from Active Directory. It will try again later in the schedule window. Current database: *<database>*
10010	Managed Folder Assistant	Error	LogAlways	The managed folder assistant skipped processing all databases on the local server because the audit log is enabled but the path to the audit log is missing in Active Directory. It will try again later in the schedule window.

				Current database: *<database>*
10011	Managed Folder Assistant	Error	LogAlways	The managed folder assistant could not configure the audit log. It will stop processing the current database: '%1'. It will try again later in the schedule window. Exception details: *<details>*
10012	Managed Folder Assistant	Error	LogAlways	The managed folder assistant did not write to the audit log. It will stop processing the current database: *<database>*. It will try to write to the audit log again later in the schedule window. Exception details: *<details>*
10017	Managed Folder Assistant	Error	LogAlways	An exception occurred in the Managed Folder

				Assistant while it was processing Mailbox: <*mailbox*> Folder: Name: <*folder name*> Id: <*folder ID*> Item: Ids: <*IDs*>. Exception: <*exception*>.

MRM events in the Assistants category

Event ID	Category	Event type	Logging	Value or description
9004	Assistants	Warning	LogAlways	Service <*service*>. <*service*> failed to process mailbox <*mailbox*>. The following exception caused the failure: <*exception*>
9014	Assistants	Warning	LogAlways	Service <*service*>. Unable to process schedule changes. The following exception caused the failure: <*exception*>
9017	Assistants	Information	LogAlways	Service <*service*>. <*service*> for database <*database*> is entering a

				scheduled time window. There are *<number>* mailboxes to process.
9018	Assistants	Information	LogAlways	Service *<service>*. *<service>* for database *<database>* is exiting a scheduled time window. *<number>* out of *<number>* mailboxes were successfully processed. *<number>* mailboxes were skipped due to errors. *<number>* mailboxes were processed separately. *<number>* mailboxes were not processed due to insufficient time. **Note:** The Managed Folder Assistant will resume where it left off the next time it runs.

9019	Assistants	Warning	LogPeriodic	Service <*service*>. Unable to save progress for <*service*> on database <*database*>. (The assistant was unable to save where it stopped so that it could resume there when it restarts.) The following exception caused the failure: <*exception*>
9020	Assistants	Warning	LogAlways	Service <*service*>. <*assistant name*> failed to start for database <*database*>. The following exception caused the failure: <*exception*>
9021	Assistants	Information	LogAlways	Service <*service*>. <*service*> for database <*database*> is processing an on-demand request. There are <*number*> mailboxes to

				process.
9022	Assistants	Information	LogAlways	Service <*service*>. <*service*> for database <*database*> has finished an on-demand request. <*number*> out of <*number*> mailboxes were successfully processed. <*number*> mailboxes were skipped due to errors.
9023	Assistants	Warning	LogAlways	Service <*service*>. <*service*> failed to start time window processing on database <*database*>. The following exception caused the failure: <*exception*>
9025	Assistants	Information	LogAlways	Service <*service*>. <*service*> skipped <*number*> mailboxes on database <*database*>.

				Mailboxes: <*mailboxes*>
9026	Assistants	Warning	LogAlways	Service <*service*>. <*service*> failed to start on-demand processing on database <*database*>. The following exception caused the failure: <*exception*>
9027	Assistants	Error	LogAlways	Service <*service*>. <*service*> caused the process to terminate <*number*> times while processing mailbox <*mailbox*> on database <*database*>. This mailbox will no longer be processed in the requested time window or on-demand request. The following exception caused the failure: <*exception*>

9028	Assistants	Warning	LogAlways	Service <*service*>. <*service*> caused the process to terminate <*number*> times while processing mailbox <*mailbox*> on database <*database*>. The following exception caused the failure: <*exception*>
9033	Assistants	Warning	LogAlways	Service <*service*>. <*service*> for database <*database*> received an on-demand request. However, there are no mailboxes to process.
9034	Assistants	Information	LogAlways	Service <*service*> halted time based operations for managed folder assistant on database <*database*>.
9035	Assistants	Warning	LogAlways	Service <*service*>. <*assistant name*>

				was unable to process <*number*> mailboxes because insufficient time.
9037	Assistants	Error	LogAlways	Service <*service*>. An exception was encountered while processing a RPC. Method: <*method*>, Exception: <*exception*>

Journaling

Exchange Server 2013 > Messaging policy and compliance >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-20*

Journaling can help your organization respond to legal, regulatory, and organizational compliance requirements by recording inbound and outbound email communications. When planning for messaging retention and compliance, it's important to understand journaling, how it fits in your organization's compliance policies, and how Microsoft Exchange Server 2013 helps you secure journaled messages.

Contents

Why journaling is important

First, it's important to understand the difference between journaling and archiving:

- *Journaling* is the ability to record all communications, including email communications, in an organization for use in the organization's email retention or archival strategy. To meet an increasing number of regulatory and compliance requirements, many organizations must maintain records of communications that occur when employees perform daily business tasks.
- *Archiving* refers to backing up the data, removing it from its native environment, and storing it elsewhere, therefore reducing the strain of data storage. You can use Exchange journaling as a tool in your email retention or archival strategy.

Although journaling may not be required by a specific regulation, compliance may be achieved through journaling under certain regulations. For example, corporate officers in some financial sectors may be held liable for the claims made by their employees to their customers. To verify that the claims are accurate, a corporate officer may set up a system where managers review some part of employee-to-client communications regularly. Every quarter, the managers verify compliance and approve their employees' conduct. After all managers report approval to the corporate officer, the corporate officer reports compliance, on behalf of the company, to the regulating body. In this example, email messages might be one type of the employee-to-client communications that managers must review; therefore, journaling can be used to collect all email messages sent by client-facing employees. Other client communication mechanisms may include faxes and telephone conversations, which may also be subject to regulation. The ability to journal all classes of data in an enterprise is a valuable functionality of the IT architecture.

The following list shows some of the more well-known U.S. and international regulations where journaling may help form part of your compliance strategies:

- Sarbanes-Oxley Act of 2002 (SOX)
- Security Exchange Commission Rule 17a-4 (SEC Rule 17 A-4)
- National Association of Securities Dealers 3010 & 3110 (NASD 3010 & 3110)
- Gramm-Leach-Bliley Act (Financial Modernization Act)
- Financial Institution Privacy Protection Act of 2001
- Financial Institution Privacy Protection Act of 2003
- Health Insurance Portability and Accountability Act of 1996 (HIPAA)
- Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism Act of 2001 (Patriot Act)
- European Union Data Protection Directive (EUDPD)
- Japan's Personal Information Protection Act

Return to top

Journaling agent

In an Exchange 2013 organization, all email traffic is routed by Mailbox servers. All messages

traverse at least one server running the Transport service in their lifetime. The *Journaling agent* is a compliance-focused transport agent that processes messages on Mailbox servers. It fires on the **OnSubmittedMessage** and **OnRoutedMessage** transport events.

Note:

In Exchange 2013, the Journaling agent is a built-in agent. Built-in agents aren't included in the list of agents returned by the **Get-TransportAgent** cmdlet. For more details, see Transport agents.

Exchange 2013 provides the following journaling options:

- **Standard journaling** Standard journaling is configured on a mailbox database. It enables the Journaling agent to journal all messages sent to and from mailboxes located on a specific mailbox database. To journal all messages to and from all recipients and senders, you must configure journaling on all mailbox databases on all Mailbox servers in the organization.
- **Premium journaling** Premium journaling enables the Journaling agent to perform more granular journaling by using journal rules. Instead of journaling all mailboxes residing on a mailbox database, you can configure journal rules to match your organization's needs by journaling individual recipients or members of distribution groups. You must have an Exchange Enterprise client access license (CAL) to use premium journaling.

When you enable standard journaling on a mailbox database, this information is saved in Active Directory and is read by the Journaling agent. Similarly, journal rules configured with premium journaling are also saved in Active Directory and applied by the Journaling agent. For more information about how to configure standard and premium journaling, see Manage journaling.

Return to top

Journal rules

The following are key aspects of journal rules:

- **Journal rule scope** Defines which messages are journaled by the Journaling agent.
- **Journal recipient** Specifies the SMTP address of the recipient you want to journal.
- **Journaling mailbox** Specifies one or more mailboxes used for collecting journal reports.

Journal rule scope

You can use a journal rule to journal only internal messages, only external messages, or both. The following list describes these scopes:

- **Internal messages only** Journal rules with the scope set to journal internal messages sent between the recipients inside your Exchange organization.
- **External messages only** Journal rules with the scope set to journal external messages sent to recipients or received from senders outside your Exchange organization.
- **All messages** Journal rules with the scope set to journal all messages that pass through your organization regardless of origin or destination. These include messages that may have already

been processed by journal rules in the Internal and External scopes.

Journal recipient

You can implement targeted journaling rules by specifying the SMTP address of the recipient you want to journal. The recipient can be an Exchange mailbox, distribution group, mail user, or contact. These recipients may be subject to regulatory requirements, or they may be involved in legal proceedings where email messages or other communications are collected as evidence. By targeting specific recipients or groups of recipients, you can easily configure a journaling environment that matches your organization's processes and meets regulatory and legal requirements. Targeting only the specific recipients that need to be journaled also minimizes storage and other costs associated with retention of large amounts of data.

All messages sent to or from the journaling recipients you specify in a journaling rule are journaled. If you specify a distribution group as the journaling recipient, all messages sent to or from members of the distribution group are journaled. If you don't specify a journaling recipient, all messages sent to or from recipients that match the journal rule scope are journaled.

Unified Messaging-enabled journal recipients

Many organizations that implement journaling may also use Unified Messaging (UM) to consolidate their email, voice mail, and fax infrastructure. However, you may not want the journaling process to generate journal reports for messages generated by Unified Messaging. In these cases, you can decide whether to journal voice mail messages and missed call notification messages handled by an Exchange 2013 server running the Unified Messaging service or to skip such messages. If your organization doesn't require journaling of such messages, you can reduce the amount of hard disk storage space required to store journal reports by skipping these messages.

Note:

Messages that contain faxes generated by a the Unified Messaging service are always journaled, even if you disable the journaling of Unified Messaging voice mail and missed call notification messages.

For more information about how to enable or disable voice mail and missed call notification messages, see Disable or enable journaling of voice mail and missed call notifications.

Journaling mailbox

The journaling mailbox is used to collect journal reports. How you configure the journaling mailbox depends on your organization's policies, regulatory requirements, and legal requirements. You can specify one journaling mailbox to collect messages for all the journal rules configured in the organization, or you can use different journaling mailboxes for different journal rules or sets of journal rules.

Important:

You can't designate an Office 365 mailbox as a journaling mailbox. You can deliver journal reports to an on-premises archiving system or a third-party archiving service. If you're running a hybrid deployment with your mailboxes split between on-premises servers and Office 365, you can designate an on-premises mailbox as the journaling mailbox for your Office 365 and on-premises mailboxes.

Important:

Journaling mailboxes contain very sensitive information. You must secure journaling mailboxes because they collect messages that are sent to and from recipients in your organization. These messages may be part of legal proceedings or may be subject to regulatory requirements. Various laws require that messages remain tamper-free before they're submitted to an investigatory authority. We recommend that you create policies that govern who can access the journaling mailboxes in your organization, limiting access to only those individuals who have a direct need to access them. Speak with your legal representatives to make sure that your journaling solution complies with all the laws and regulations that apply to your organization.

Alternate journaling mailbox

When the journaling mailbox is unavailable, you may not want the undeliverable journal reports to collect in mail queues on Mailbox servers. Instead, you can configure an alternate journaling mailbox to store those journal reports. The alternate journaling mailbox receives the journal reports as attachments in the non-delivery reports (NDRs) generated when the journaling mailbox or the server on which it's located refuses delivery of the journal report or becomes unavailable.

When the journaling mailbox becomes available again, you can use the **Send Again** feature of Office Outlook to submit journal reports for delivery to the journaling mailbox.

When you configure an alternate journaling mailbox, all the journal reports that are rejected or can't be delivered across your entire Exchange 2013 organization are delivered to the alternate journaling mailbox. Therefore, it's important to make sure that the alternate journaling mailbox and the Mailbox server where it's located can support many journal reports.

Caution:

If you configure an alternate journaling mailbox, you must monitor the mailbox to make sure that it doesn't become unavailable at the same time as the journal mailboxes. If the alternate journaling mailbox also becomes unavailable or rejects journal reports at the same time, the rejected journal reports are lost and can't be retrieved.

Because the alternate journaling mailbox collects all the rejected journal reports for the entire Exchange 2013 organization, you must make sure that this doesn't violate any laws or regulations that apply to your organization. If laws or regulations prohibit your organization from allowing journal reports sent to different journaling mailboxes from being stored in the same alternate journaling mailbox, you may be unable to configure an alternate journaling mailbox. Discuss this with your legal representatives to determine whether you can use an alternate journaling mailbox.

When you configure an alternate journaling mailbox, you should use the same criteria that you used when you configured the journaling mailbox.

Important:

The alternate journaling mailbox should be treated as a special dedicated mailbox. Any messages addressed directly to the alternate journaling mailbox aren't journaled.

Return to top

Journal rule replication

Journal rules are stored in Active Directory and applied by all Mailbox servers in the Exchange 2013 organization. When you create, modify, or remove a journal rule, the change is replicated to all Active Directory servers in the organization. All Mailbox servers in the organization then retrieve the updated journal rule configuration from the Active Directory servers and apply the new or modified journal rules.

By replicating all the journal rules across the organization, Exchange 2013 enables you to provide a consistent set of journal rules across the organization. All messages that pass through your Exchange 2013 organization are subject to the same journal rules.

Important:

Replication of journal rules across an organization is dependant on Active Directory replication. Replication time between Active Directory domain controllers varies depending on the number of sites in the organization and the speed of links and other factors outside the control of Microsoft Exchange. Consider replication delays when you implement journal rules in your organization. For more information about Active Directory replication, see Introduction to Active Directory Replication and Topology Management Using Windows PowerShell.

Important:

Each Mailbox server caches distribution group membership to avoid repeated round trips to Active Directory. The expanded groups cache reduces the number of requests that each Mailbox server must make to an Active Directory domain controller. By default, entries in the expanded groups cache expire in four hours. Therefore, if you specify a distribution group as the journal recipient, changes to distribution group membership may not be applied to journal rules until the expanded groups cache is updated. To force an immediate update of the recipient cache, you must stop and start the Microsoft Exchange Transport service. You must do this for each Mailbox server where you want to forcibly update the recipient cache.

Return to top

Journal reports

A *journal report* is the message that the Journaling agent generates when a message matches a journal rule and is to be submitted to the journaling mailbox. The original message that matches the journal rule is included unaltered as an attachment to the journal report. The body of a journal report contains information from the original message such as the sender email address, message subject, message-ID, and recipient email addresses. This is also referred to as envelope journaling, and is the only journaling method supported by Exchange 2013.

Journal reports and IRM-protected messages

When implementing journaling in an Exchange 2013 environment, you must consider journaling reports and IRM-protected messages. IRM-protected messages will affect the search and discovery capabilities of third-party archiving systems that don't have RMS support built-in. In Exchange 2013, you can configure Journal Report Decryption to save a clear-text copy of the message in a journal report.

Return to top

Interoperability with Exchange 2007

There isn't much difference between journaling functionality in Exchange 2013, Exchange 2010, and Exchange 2007. However, in Exchange 2010 and later, Setup creates a separate container in Active Directory to store journal rules. When you set up the first Exchange 2010 or later server in an Exchange 2007 organization, Setup creates a copy of the existing journal rules in Exchange 2007 and stores them in the new container. When Setup completes, the journal rules in both versions of Exchange are consistent.

After Setup, if you change the journal rule configuration on Exchange 2010 (or later), you must make the same change on Exchange 2007 servers to make sure they're consistent. Similarly, changes made to Journal rules on Exchange 2007 must be also be made in Exchange 2010 or later. You can also export journal rules from Exchange 2007 and import them to Exchange 2013.

Return to top

Troubleshooting

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.. If you're having trouble with the **JournalingReportDNRTo** mailbox, see Transport and Mailbox Rules in Exchange Online don't work as expected.

Manage journaling

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-11-20*

Journaling can help your organization respond to legal, regulatory, and organizational compliance

requirements by recording inbound and outbound e-mail communications. This topic shows you how to perform basic tasks related to managing journaling in Exchange 2013.

Standard journaling is configured on a mailbox database. It enables the Journaling agent to journal all messages sent to and from mailboxes located on a specific mailbox database. You can also use premium journaling enables the Journaling agent to perform more granular journaling by using journal rules. Instead of journaling all mailboxes residing on a mailbox database, you can configure journal rules to match your organization's needs by journaling individual recipients or members of distribution groups. You must have an Exchange Enterprise client access license (CAL) to use premium journaling.

To learn more about journaling, see Journaling.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Journaling" entry in the Messaging policy and compliance permissions topic.
- A journaling mailbox has been created, or an existing mailbox is available for use as the journaling mailbox.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection. If you're having trouble with the **JournalingReportDNRTo** mailbox, see Transport and Mailbox Rules in Exchange Online don't work as expected.

What do you want to do?

Create a journal rule

Use the EAC to create a journal rule

1. Navigate to **Compliance management** > **Journal rules**, and then click **Add +**.
2. In **Journal rule**, provide a name for the journal rule and then compete the following fields:
   - **If the message is sent to or received from** Specify the recipient that the rule will target. You can either select a specific recipient or apply the rule to all messages.
   - **Journal the following messages** Specify the scope of the journal rule. You can journal only the internal messages, only the external messages, or all messages regardless of origin or destination.
   - **Send journal reports to** Type the address of the journaling mailbox that will receive all the journal reports.

3. Click **Save** to create the journal rule.

Use the Shell to create a journal rule

This example creates the journal rule Discovery Journal Recipients to journal all messages sent from and received by the recipient user1@contoso.com.

```
New-JournalRule -Name "Discovery Journal Recipients" -
Recipient user1@contoso.com -JournalEmailAddress "Journal
Mailbox" -Scope Global -Enabled $True
```

How do you know this worked?

To verify that you have successfully created the journal rule, do one of the following:

- From the EAC, verify that the new journal rule you created is listed on the **Journal rules** tab.
- From the Shell, verify that the new journal rule exists by running the following command (the example below verifies the rule created in the Shell example above):

```
Get-JournalRule "Discovery Journal Recipients"
```

View or modify a journal rule

Use the EAC to view or modify a journal rule

1. Navigate to **Compliance management** > **Journal rules**.
2. In the list view, you'll see all the journal rules in your organization.
3. Double-click the rule you want to view or modify.
4. In **Journal Rule**, modify the settings you want. For more information about the settings in this dialog box, see the procedure Use the EAC to create a journal rule earlier in this topic.

Use the Shell to view or modify a journal rule

This example displays a summary list of all journal rules in the Exchange organization:

```
Get-JournalRule
```

This example retrieves the journal rule Brokerage Journal Rule, and pipes the output to the **Format-List** command to display rule properties in a list format:

```
Get-JournalRule "Brokerage Journal Rule" | Format-List
```

If you want to modify the properties of a specific rule, you need to use the Set-JournalRule cmdlet. This example changes the name of the journal rule `JR-Sales` to `TraderVault`. The following rule settings are also changed:

- Recipient
- JournalEmailAddress
- Scope

```
Set-JournalRule JR-Sales -Name TraderVault -Recipient
traders@woodgrovebank.com -JournalEmailAddress
tradervault@woodgrovebank.com -Scope Internal
```

How do you know this worked?

To verify that you have successfully modified a journal rule, do one of the following:

- From the EAC, navigate to **Compliance management**, > **Journal rules**. Double-click the rule you modified and verify your changes were saved.
- From the Shell, verify that you modified the journal rule successfully by running the following command. This command will list the properties you modified along with the name of the rule (the example below verifies the rule modified in the Shell example above):

```
Get-TransportRule "TraderVault" | Format-List
Name,Recipient,JournalEmailAddress,Scope
```

Enable or disable a journal rule

Important:
When you disable a journal rule, the journaling agent will stop journaling messages targeted by that rule. While a journal rule is disabled, any messages that would have normally been journaled by the rule aren't journaled. Make sure that you don't compromise the regulatory or compliance requirements of your organization by disabling a journaling rule.

Use the EAC to enable or disable a journal rule

1. Navigate to **Compliance management** > **Journal rules**.
2. In the list view, in the **On** column next to the rule's name, select the check box to enable the rule or clear it to disable the rule.

Use the Shell to enable or disable a journal rule

This example enables the rule Contoso.

```
Enable-JournalRule "Contoso Journal Rule"
```

This example disables the rule Contoso.

```
Disable-JournalRule "Contoso Journal Rule"
```

How do you know this worked?

To verify that you have successfully enabled or disabled a journal rule, do one of the following:

- From the EAC, view the list of journal rules check the status of the check box in the **On** column.
- From the Shell, run the following command to return a list of all journal rules in your organization along, including their status:

```
Get-JournalRule | Format-Table Name,Enabled
```

Remove a journal rule

Use the EAC to remove a journal rule

1. Navigate to **Compliance management** > **Journal rules**.
2. In the list view, select the rule you want to remove, and then click **Delete** 🗑.

Use the Shell to remove a journal rule

This example removes the rule Brokerage Journal Rule.

```
Remove-JournalRule "Brokerage Journal Rule"
```

How do you know this worked?

To verify that you have successfully removed the journal rule, do one of the following:

- From the EAC, verify that the rule you removed is no longer listed on the **Journal rules** tab.
- From the Shell, run the following command to verify that the rule you remove is no longer listed:

```
Get-JournalRule
```

Enable or disable per-mailbox database journaling

Caution:
Disabling message journaling on a mailbox database may result in your organization being out of compliance with any applicable messaging retention policies. When you disable message journaling on a mailbox database, journal receipts are no longer sent for messages sent or received by mailboxes on that mailbox database.

Use the EAC enable or disable per-mailbox database journaling

1. Navigate to **Servers** > **Databases**.
2. In the list view, double-click the mailbox database for which you want to enable journaling.
3. Click **Maintenance**, and then click **Browse** next to the **Journal recipient** box to select the journaling mailbox. Specifying a journal recipient enables journaling for the database.

To disable journaling, remove the journal recipient by clicking **Remove X**.

Use the Shell to enable or disable per-mailbox database journaling

This example enables journaling for the mailbox database Sales Database and sets Sales Database journal mailbox as the journal recipient.

```
Set-MailboxDatabase "Sales Database" -JournalRecipient
"Sales Database Journal Mailbox"
```

This example disables per-mailbox database journaling on the Sales Database mailbox database.

```
Set-MailboxDatabase "Sales Database" -JournalRecipient
$Null
```

This example disables per-mailbox database journaling on all mailbox databases in the Exchange organization. The Get-MailboxDatabase cmdlet is used to retrieve all mailbox databases in the Exchange organization, and results from the cmdlet are piped to the Set-MailboxDatabase cmdlet.

```
Get-MailboxDatabase | Set-MailboxDatabase -JournalRecipient
$Null
```

How do you know this worked?

To verify that you have successfully enabled or disabled per-mailbox database journaling, do one of the following:

1. From the EAC, navigate to **Servers** > **Databases**.
2. Double click the database you want to verify, and then select the **Maintenance** tab.
3. If the correct journaling recipient is listed in the **Journal recipient** box, you have successfully enabled journaling for the mailbox database. If there is no journaling recipient listed, journaling is disabled for the database.

- From the Shell, run the following command to return a list of all mailbox databases in your organization, including the journal recipients associated with them. Journaling is enabled for databases that have a journal recipient listed, otherwise it's disabled.

```
Get-MailboxDatabase | Format-Table Name,JournalRecipient
```

For more information

Journaling

Disable or enable journaling of voice mail and missed call notifications

New-JournalRule

Get-JournalRule

Set-JournalRule

Enable-JournalRule

Disable-JournalRule

Remove-JournalRule

Set-MailboxDatabase

Disable or enable journaling of voice mail and missed call notifications

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-10*

In Microsoft Exchange Server 2013, when you create a journal rule to journal email messages sent to or from recipients or senders in an Exchange organization, voice mail and missed call notifications generated by the Unified Messaging (UM) service are included. Use the procedures in this topic to turn this feature on or off for your entire organization.

Looking for other management tasks related to journaling? Check out Manage journaling.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Journaling" entry in the Messaging policy and compliance permissions topic.
- You can only use the Shell to perform this procedure.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to disable or enable journaling of voice mail and missed call notifications

This example disables journaling of voice mail and missed call notifications by setting the *VoicemailJournalingEnabled* parameter to `$false`.

```
Set-TransportConfig -VoicemailJournalingEnabled $false
```

This example enables the journaling of voice mail and missed call notifications by setting the same parameter to `$true`.

```
Set-TransportConfig -VoicemailJournalingEnabled $true
```

For detailed syntax and parameter information, see Set-TransportConfig.

For More Information

Journaling

Manage journaling

Transport rules

Exchange Server 2013 > Messaging policy and compliance >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-01-23*

Using transport rules, you can look for specific conditions in messages that pass through your organization and take action on them. Transport rules let you apply messaging policies to email messages, secure messages, protect messaging systems, and prevent information leakage.

Many organizations today are required by law, regulatory requirements, or company policies to apply messaging policies that limit the interaction between recipients and senders, both inside and outside the organization. In addition to limiting interactions among individuals, departmental groups inside the organization, and entities outside the organization, some organizations are also subject to the following messaging policy requirements:

- Preventing inappropriate content from entering or leaving the organization
- Filtering confidential organization information
- Tracking or archiving copying messages that are sent to or received from specific individuals
- Redirecting inbound and outbound messages for inspection before delivery
- Applying disclaimers to messages as they pass through the organization

Looking for management tasks related to managing transport rules? See Manage Transport Rules.

For each rule, you have the option of enforcing it, testing it and notifying the sender, or just testing the rule. You can notify the sender that they might be violating one of the rules—even before they send an offending message. You can accomplish this by configuring Policy Tips and setting the mode of the rule. Policy Tips are similar to MailTips, and can be configured to present a brief note in the Microsoft Outlook 2013 client that provides information about possible policy violations to a person creating a message. For more information, see Policy Tips.

Contents

Overview of transport rules

Transport rules are similar to the Inbox rules that are available in many email clients. The main difference between transport rules and rules you would set up in a client application such as Outlook is that transport rules take action on messages while they're in transit as opposed to after the message is delivered. Transport rules also contain a richer set of conditions, exceptions, and actions, which provides you with the flexibility to create a customized rule.

The following list summarizes the basic workflow for transport rules:

1. You use the Exchange admin center (EAC),the Shell, or a DLP policy to create a transport rule. After you create your rule, it's stored in Active Directory.
2. As messages go through the transport pipeline, the Transport rules agent is invoked. The Transport rules agent is a special Transport agent that processes the Transport rules you create.
3. The Transport rules agent scans the message, and if the message fits the conditions you specify in a transport rule, it takes the specified action on that message based on the mode of the rule.

The following sections provide detailed information about transport rule components, transport rule modes, the Transport rules agent, and how the transport rules are applied.

Return to top

Transport rule components

Transport rules consist of the following components:

- **Conditions** Conditions specify the characteristics of messages to which you want to apply a transport rule action. Some conditions examine message fields or headers, such as the To, From, or Cc fields. Other conditions examine message characteristics such as message subject, body, attachments, message size, and message classification. Most conditions require that you specify a comparison operator, such as equals, doesn't equal, or contains, and a value to match. If there are no conditions or exceptions, the rule is applied to all messages.
- **Exceptions** Exceptions are based on the same characteristics used to build conditions. However, unlike conditions, exceptions identify messages to which transport rule actions shouldn't be applied. Exceptions override conditions and prevent actions from being applied to an email message, even if the message matches all configured conditions.
- **Actions** Actions are applied to messages that match all the conditions and don't match any of the exceptions. There are many actions available, such as rejecting, deleting, or redirecting messages, adding additional recipients, adding prefixes in the message subject, or inserting disclaimers in the message body.

For a complete list of transport rule conditions, see Transport rule conditions (predicates). The list of conditions is also available in the **Transport rule** dialog in the EAC. If you use the Shell, you can

retrieve the list of conditions by using the Get-TransportRulePredicate cmdlet.

For a complete list of transport rule actions available, see Transport rule actions. The list of actions is also available in the **Transport rule** dialog box in the EAC. If you use the Shell, you can retrieve the list of actions by using the Get-TransportRuleAction cmdlet.

Return to top

Transport rule modes

There are three modes for each transport rule that define how the rule will be implemented:

- **Enforce**: All actions on the rule will be enforced.
- **Test with Policy Tips**: Any Policy Tip actions will be sent, but other enforcement actions will not be acted on
- **Test without Policy Tips**: Actions will be listed in a log file, but senders will not be notified in any way, and enforcement actions will not be acted on

In order to make sure rules work as you intend, we recommend testing rules before setting them to **Enforce**.

How transport rules are applied

All transport rules are processed by the Transport rules agent on Exchange servers. The Transport rules agent is a built-in agent that fires on the **OnResolvedMessage** transport event. All messages in an Exchange 2013 organization are processed by the Transport service.

Differences in processing based on message type

There are several types of messages that pass through an organization. The following table shows which messages types can be processed by transport rules.

Type of message	Can a rule be applied?
Regular messages Messages that contain a single rich text format (RTF), HTML, or plain text message body or a multipart or alternative set of message bodies.	Yes
Encrypted messages (Office 365 Message Encryption) Messages that are encrypted using Office	Rules can always access envelope headers contained in encrypted messages and process messages based on conditions that inspect headers.

365 Message Encryption.	For a rule to inspect or modify Office 365 message encrypted content, your organization must: • Use Exchange Server or Exchange Online. • Have transport decryption set to **Mandatory** or **Optional**. Transport decryption is set to **Optional** by default. • Have the encryption key. You can also create a rule that automatically decrypts encrypted messages.
Encrypted messages (S/MIME) Messages that are encrypted using S/MIME.	Rules can access only envelope headers contained in S/MIME encrypted messages and process messages based on conditions that inspect headers. Rules with conditions that require inspection of message content, or actions that modify content, can't be processed.
Protected messages Messages that are protected by applying an Active Directory Rights Management Services (AD RMS) rights policy template.	Rules can always access envelope headers contained in protected messages and process messages based on conditions that inspect headers. For a rule to inspect or modify protected message content, your organization must: • Use Exchange Server or Exchange Online. • Have transport decryption set to **Mandatory** or **Optional**. Transport decryption is set to **Optional** by default. • Have the encryption key.
Clear-signed messages Messages that have been signed but not encrypted.	Yes

Unified messaging email messages Messages that are created or processed by the Unified Messaging service, such as voice mail, fax, missed call notifications, and messages created or forwarded by using Microsoft Outlook Voice Access.	Yes
Anonymous	Yes
Messages sent by anonymous senders.	Yes
Read reports Reports that are generated in response to read receipt requests by senders. Read reports have a message class of `IPM.Note*.MdnRead` or `IPM.Note*.MdnNotRead`.	Yes

Return to top

How the Transport rules agent evaluates messages

The Transport rules agent evaluates the following elements when processing rules for a message:

- **Message scope** The first check performed by rules agents is whether a message falls within the scope of the agent. Transport rules aren't applied to all types of messages.
- **Priority** For messages that fall within the scope of the rules agent, the agent starts processing rules based on rule priority in ascending order. The rule with a priority of 0 is processed first, followed by the rule with a priority of 1 and so on. Transport rule priority values range from `0` to `n-1`, where `n` is the total number of transport rules. Only enabled rules are processed. You can change the rule priority.
- **Rules with no conditions or exceptions** If a rule has no conditions and no exceptions, it's applied to all messages.
- **Conditions** The conditions describe the type of message for which the rule is intended, and the rules agent applies the rules to the messages that match the criteria specified in the rule conditions.
- **Rules with multiple conditions** It may be necessary to use more than one condition to specify a rule. For a rule's action to be applied to a message, it must match all the conditions selected in the rule. For example, if a rule uses the conditions **The sender is a member of this group** and **The subject includes any of these words**, the message must match both conditions. It must be

sent by a member of the specified distribution group, and the message subject must contain the specified word.

- **Conditions with multiple values** Some conditions allow you to specify more than one value. If one condition allows you to enter multiple values, the message must match any value specified for that condition. For example, if an email message has the subject **Stock price information**, and the **The subject includes any of these words** condition on a transport rule is configured to match the words **Contoso** and **stock**, the condition is satisfied because the subject contains at least one of the condition values.
- **Exceptions** A rule isn't applied to messages that match any of the exceptions defined in the rule. This is exactly opposite of how the rules agent treats conditions. For example, if the exceptions **Except if the sender is this person** and **Except if the subject or body includes any of these words** are selected, the message fails to match the rule condition if the message is sent from any of the specified senders, or if the message contains any of the specified words.
- **Actions** Messages that match the rules conditions get all actions specified in the rule applied to them. For example, if the actions **Prepend the subject of the message with** and **Add recipients to the Bcc box** are selected, both actions are applied to the message. The message will get the specified string prefixed to the message subject, and the recipients specified will be added as Bcc recipients.

Keep in mind that some actions, such as the **Delete the message without notifying anyone** action, prevent subsequent rules from being applied to a message. You can also configure a rule so that when that rule is applied, the rules agent stops processing any subsequent rules.

Return to top

Transport rules and group membership

When you define a transport rule using a condition that expands membership of a distribution group, the resulting list of recipients is cached by the Transport service on the Mailbox server that applies the rule. This is known as the *Expanded Groups Cache* and is also used by the Journaling agent for evaluating group membership for journal rules. By default, the Expanded Groups Cache stores group membership for four hours. Recipients returned by the recipient filter of a dynamic distribution group are also stored. The Expanded Groups Cache makes repeated round-trips to Active Directory and the resulting network traffic from resolving group memberships unnecessary.

In Exchange 2013, this interval and other parameters related to the Expanded Groups Cache are configurable. You can lower the cache expiration interval, or disable caching altogether, to ensure group memberships are refreshed more frequently. You must plan for the corresponding increase in load on your Active Directory domain controllers for distribution group expansion queries. You can also clear the cache on a Mailbox server by restarting the Microsoft Exchange Transport service on that server. You must do this on each Mailbox server where you want to clear the cache. When creating, testing, and troubleshooting transport rules that use conditions based on distribution group membership, you must also consider the impact of Expanded Groups Cache.

Return to top

Rule storage and replication

The Transport rules you create are stored in Active Directory and are available after Active Directory replication on all Exchange servers in your Exchange 2013 organization. This allows you to apply a consistent set of rules across the entire Exchange organization.

When a transport rule is created or an existing transport rule is modified or deleted, the change is replicated to all Active Directory domain controllers in the organization. All the Exchange servers in the organization then read the new configuration from the Active Directory servers and apply the new or modified transport rules.

Important:

Replication of transport rules across an organization depends on Active Directory replication. Replication time between Active Directory domain controllers varies depending on the number of sites in the organization, slow links, and other factors outside the control of Exchange. When you configure transport rules in your organization, make sure that you consider replication delays. For more information about Active Directory replication, see Active Directory Replication Technologies.

Important:

The Transport service on each Mailbox server maintains a recipient cache that's used to look up recipient and distribution list information. The recipient cache reduces the number of requests that each Mailbox server must make to an Active Directory domain controller. The recipient cache updates every four hours. You can't modify the recipient cache update interval. Therefore, changes to transport rule recipients, such as the addition or removal of distribution list members, may not be applied to transport rules until the recipient cache is updated. To force an immediate update of the recipient cache, you must stop and start the Microsoft Exchange Transport service. You must do this for each Mailbox server where you want to forcibly update the recipient cache.

Note:

Each time the Transport service on the Mailbox server retrieves a new transport rule configuration, an event is logged in the Security log in Event Viewer.

Rule replication and storage in mixed environments

There are two mixed environment scenarios that are common: hybrid deployments where part of your organization resides on Office 365, and Exchange 2013 coexisting with Exchange 2010 or Exchange 2007.

In the hybrid scenario, there is no replication of rules between your on-premises deployment and Office 365. Therefore, when you create a rule in your on-premises Exchange organization, you need to create a matching rule in Office 365. The rules you create in Office 365 are stored in the cloud, along with the rest of your Office 365 organization configuration, whereas the rules you create in your on-premises Exchange organization are stored locally in Active Directory. When managing rules in a hybrid scenario, you need to make sure that you keep the two sets of rules synchronized

by making the change in both places, or making the change in one environment and then exporting the rules and importing them in the other environment.

Important:
Even though there is a substantial overlap between the conditions and actions available in Office 365 and on-premises Exchange, there are differences. If you plan on creating the same rule in both locations, make sure that all conditions and actions you plan to use are available. To see the list of available conditions and actions for each deployment, see the following topics: **Transport Rule Conditions (Predicates)** in Office 365 Transport rule conditions (predicates) in on-premises Exchange **Transport rule actions** in Office 365 Transport rule actions in on-premises Exchange

When you coexist with Exchange 2010 or Exchange 2007, all transport rules are stored in Active Directory and replicated across your organization regardless of the Exchange Server version you used to create the rules. However, all transport rules are associated with the Exchange server version that was used to create them and are stored in a version-specific container in Active Directory. When you first deploy Exchange 2013 in your organization, any existing rules are imported to Exchange 2013 as part of the setup process. However, any changes afterwards would need to be made with both versions. For example, if you change an existing rule using the Exchange 2013 EAC, you need to make the same change using the Exchange 2010 or Exchange 2007 EMC.

Return to top

For more information

Manage Transport Rules

Transport rule conditions (predicates)

Transport rule actions

Transport agents

Transport rule conditions (predicates)

Exchange Server 2013 > Messaging policy and compliance > Transport rules >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-31*

Transport rule conditions and exceptions are used to define when a transport rule is applied. For example, when adding a disclaimer, you could define the conditions to only add the disclaimer to messages containing specific words, to messages from a specific person or group, or to all messages except those from a specific group.

Looking for:

- Management tasks related to transport rules? See Transport rule procedures
- Exchange Online version of this topic? See **Transport Rule Conditions (Predicates)**
- Exchange Online Protection version of this topic? See **Transport rule conditions (predicates)**

Contents

Conditions and Condition Properties for a Mailbox Server

To determine whether a transport rule should be applied to a message, most conditions have one or more properties for which you must specify a value. For example, the **The sender is** condition requires that you specify the sender for the message. Some conditions have two properties. For example, the **A message header includes any of these words** condition requires one property to specify the header to examine, such as To, From, Received, or Content-Type, and a second property for the text to look for in the specified message header. Others don't have properties. For example, the **Any attachment has executable content** condition simply inspects whether any attachment in a message has executable content, and therefore doesn't require any values.

To assign a value to a condition in the Exchange admin center (EAC), you can use the drop-down lists and secondary dialog boxes that are displayed in the **Transport rule** page. These help you select the correct property types and valid values for those property types. If you're using the Shell to define conditions, see the descriptions in Condition properties.

The following table lists the conditions that can be used with transport rules in Exchange 2013 Mailbox Server.

Note:

Each condition listed in the following table also has an equivalent exception that can be selected in the EAC. In the Shell, conditions that can be used as exceptions start with `ExceptIf`. For example, for the `FromMemberOf` condition, the parameter that can be used as an exception in transport rule cmdlets is called *ExceptIfFromMemberOf*.
The same condition object contains the logic for use in both transport rule conditions and exceptions. Therefore, when you use the Get-`TransportRuleCondition` cmdlet to list conditions, exceptions aren't listed as separate conditions.

Condition name in EAC	Condition name in Shell	Condition property type	Description
The sender is	`From`	Addresses	This condition matches

			messages sent by the specified mailboxes, mail-enabled users, or contacts.
The sender is located	`FromScope`	FromUserScope	This condition matches messages that are sent by senders within the specified scope.
The sender is a member of	`FromMemberOf`	Addresses	This condition matches messages where the sender is a member of the specified distribution group.
The sender address includes	`FromAddressContainsWords`	Words	This condition matches messages that contain the specified words in the sender's address.
The sender address matches	`FromAddressMatchesPatterns`	Patterns	This condition matches messages that contain text patterns in the sender's address that match the specified regular expression.
The sender's specified properties include any of these words	`SenderADAttributeContainsWords`	Words	This condition matches messages where the specified attribute of the sender contains specified words. **Note:** When matching a country using

			CountryOrRegion, you must use the ISO country code, rather than the country name. For example, to match all senders from Germany: `Set-TransportRule`
The sender's specified properties match these text patterns	`SenderADAttributeMatchesPatterns`	Patterns	This condition matches messages where the specified attribute of the sender contains text patterns that match the specified regular expression.
The sender has overridden the Policy Tip	`HasSenderOverride`	Not applicable	This condition matches messages where the sender has chosen to override a DLP policy.
Sender's IP address is in the range	`SenderIPRanges`	IPRanges	This condition matches messages where the sender's IP address falls within the specified ranges.
The sender's domain is	`SenderDomainIs`	Domain name	This condition matches messages where the sender's domain matches the specified domain name.
The recipient is	`SentTo`	Addresses	This condition matches messages where one of the recipients is the

			specified mailbox, mail-enabled user, or contact. The specified recipients can be listed in the **To**, **Cc**, or **Bcc** fields. **Note:** You can't specify a distribution group with this condition. If you need to create a rule that takes action on messages sent to a distribution group, use the **To box contains** (**AnyOfToHeader**) condition instead.
The recipient is located	`SentToScope`	ToUserScope	This condition matches messages that are sent to recipients within the specified scope.
The recipient is a member of	`SentToMemberOf`	Addresses	This condition matches messages that contain recipients who are members of the specified distribution group. The distribution group can be listed in the **To**, **Cc**, or **Bcc** fields.
The recipient address includes	`RecipientAddressContainsWords`	Words	This condition matches messages where a recipient's address contains any of the

			specified words.
The recipient address matches	`RecipientAddressMatchesPatterns`	Patterns	This condition matches messages where a recipient's address matches a specified regular expression.
The recipient's specified properties include any of these words	`RecipientADAttributeContainsWords`	Words	This condition matches messages where the specified attribute of a recipient contains any of the specified words..
The recipient's specified properties match these text patterns	`RecipientADAttributeMatchesPatterns`	Patterns	This condition matches messages where the specified attribute of a recipient matches the specified regular expression.
A recipient's domain is	`RecipientDomainIs`	Domain Name	This condition matches messages where the domain of any recipient of the message matches the specified domain name.
The subject or body includes	`SubjectOrBodyContainsWords`	Words	This condition matches messages that have the specified words in the **Subject** field or message body.
The subject or body matches	`SubjectOrBodyMatchesPatterns`	Patterns	This condition matches messages where text

			patterns in the **Subject** field or message body match a specified regular expression.
The subject includes	`SubjectContainsWords`	Words	This condition matches messages that have the specified words in the **Subject** field.
The subject matches	`SubjectMatchesPatterns`	Patterns	This condition matches messages where text patterns in the **Subject** field match a specified regular expression.
Any attachment's content includes	`AttachmentContainsWords`	Words	This condition matches messages with attachments that contain a specified string.
Any attachment's content matches	`AttachmentMatchesPatterns`	Patterns	This condition matches messages with attachments that contain a text pattern that matches a specified regular expression. **Note:** Only the first 150 KB of the attachment is scanned when trying to match a pattern.
Any attachment's	`AttachmentIsUnsupported`	Not applicable	This condition matches

content can't be inspected			messages with attachments that aren't supported.
Any attachment's file name matches	`AttachmentNameMatchesPatterns`	Patterns	This condition matches messages that contain text patterns in an attachment file name that matches a specified regular expression.
Any attachment's file extension matches	`AttachmentExtensionMatchesWords`	Words	This condition matches messages that contain an attachment whose extension matches any of the specified words.
Any attachment size is greater than or equal to	`AttachmentSizeOver`	Size	This condition matches messages that contain attachments greater than or equal to the specified value.
The message didn't complete scanning	`AttachmentProcessingLimitExceeded`	Not applicable	This condition matches messages for which the rules engine couldn't complete scanning of the attachments. This condition can be used to create rules that work together with other attachment processing rules and gives you the ability to

			handle messages whose content couldn't be fully scanned.
Any attachment has executable content	`AttachmentHasExecutableContent`	Not applicable	This condition matches messages that contain executable files as attachments. The system uses auto-detection of file types by inspecting file properties rather than the actual file extension, thus preventing spammers from being able to bypass transport rule filtering by renaming the file extension.
Any attachment is password protected	`AttachmentIsPasswordProtected`	Not applicable	This condition matches messages that contain compressed archive attachments that are password protected, and therefore cannot be scanned.
The message contains sensitive information	`MessageContainsDataClassifications`	SensitiveInformation	This condition matches messages that contain sensitive information. This condition is required for rules using the `NotifySender`

			(Notify the sender with a Policy Tip) action.
The To box contains	`AnyOfToHeader`	Addresses	This condition matches messages where the **To** field includes any of the specified recipients.
The To box contains a member of	`AnyOfToHeaderMemberOf`	Addresses	This condition matches messages where the **To** field contains a recipient who is a member of the specified distribution group.
The Cc box contains	`AnyOfCcHeader`	Addresses	This condition matches messages where the **Cc** field includes any of the specified recipients.
The Cc box contains a member of	`AnyOfCcHeaderMemberOf`	Addresses	This condition matches messages where the **Cc** field contains a recipient who is a member of the specified distribution group.
The To or Cc box contains	`AnyOfToCcHeader`	Addresses	This condition matches messages where the **To** or **Cc** fields include any of the specified recipients.

The To or Cc box contains a member of	`AnyOfToCcHeaderMemberOf`	Addresses	This condition matches messages where the **To** or **Cc** fields contains a recipient who is a member of the specified distribution group.
The message size is greater than or equal to	`MessageSizeOver`	Size	This condition matches messages whose overall size is greater than or equal to the specified value.
The message character set name includes any of these words	`ContentCharacterSetContainsWords`	Character Sets	This condition matches messages that have any of the character set names specified.
The sender is one of the recipients'	`SenderManagementRelationship`	ManagementRelationship	This condition matches messages where the sender has the specified management relationship with a recipient.
The message is between members of these groups	`BetweenMemberOf1` and `BetweenMemberOf2`	First property: Addresses (`BetweenMemberOf1`) Second property: Addresses (`BetweenMemberOf2`)	This condition matches messages that are sent between members of two distribution groups.
The manager of the sender or recipient is	`ManagerForEvaluatedUser` and `ManagerAddress`	First property: EvaluatedUser	This condition matches messages where the

		(`ManagerForEvaluatedUser`) Second property: Addresses (`ManagerAddresses`)	specified user's (sender or recipient) manager exists in the list of specified addresses.
The sender's and any recipient's property compares as	`ADAttributeComparisonAttribute` and `ADComparisonOperator`	First property: ADAttribute (`ADComparisonAttribute`) Second property: Evaluation (`ADComparisonOperator`)	This condition matches messages where the sender's specified Active Directory attribute matches or doesn't match (as specified in the **Evaluation** property) the same attribute of any recipient.
The message type is	`MessageTypeMatches`	MessageType	This condition matches messages of the specified type.
The message is classified as	`HasClassification`	Classification	This condition matches messages that have the specified classification.
The message isn't marked with any classifications	`HasNoClassification`	Not applicable	This condition matches messages that don't have a message classification.
The message has an SCL greater than or equal to	`SCLOver`	SclValue	This condition matches messages that are assigned a spam confidence level (SCL) matching or exceeding

			the specified value.
The message importance is set to	`WithImportance`	Importance	This condition matches messages marked with the specified priority.
A message header includes	`HeaderContainsMessageHeader` and `HeaderContainsWords`	First property: MessageHeader (`HeaderContainsMessageHeader`) Second property: Words (`HeaderContainsWords`)	This condition matches messages where the specified message header contains one of the specified words.
A message header matches	`HeaderMatchesMessageHeader` and `HeaderMatchesPatterns`	First property: MessageHeader (`HeaderMatchesMessageHeader`) Second property: Patterns (`HeaderMatchesPatterns`)	This condition matches messages where the specified message header contains a text pattern that matches a specified regular expression.

Condition property values for a Mailbox server

Each property that you use to define a transport rule condition requires a value. Here's a list of values for each condition property available in Exchange 2013 Mailbox servers.

Property type	Valid values	Description
ADAttribute	One of the Active Directory attributes available for use	`ADAttribute` accepts the name of one of the following Active Directory attributes: • `DisplayName` • `FirstName` • `Initials` • `LastName` • `Office` • `PhoneNumber` • `OtherPhoneNumber` • `Email`

		• Street • POBox • City • State • ZipCode • Country • UserLogonName • HomePhoneNumber • OtherHomePhoneNumber • PagerNumber • MobileNumber • FaxNumber • OtherFaxNumber • Notes • Title • Department • Company • Manager • CustomAttribute1 - CutomAttribute15 When you use the Shell to create a Transport rule that contains a condition which uses an Active Directory attribute, you must specify an attribute name from the preceding list followed by a colon (:) and the word or text pattern you want to match in the specified attribute. The entire notation should be enclosed in quotation marks ("). For example, to specify the attribute City and the values San Francisco or Palo Alto, you must use City:San Francisco, Palo Alto. You can also specify multiple Active Directory attributes and value pairs. For example, "City:San Francisco, Palo Alto","Department:Sales, Finance". In this case, the

		recipient's `City` attribute should contain either `San Francisco` or `Palo Alto`, and the `Department` attribute should contain either `Sales` or `Finance`.
Addresses	Array of Active Directory mailbox, contact, or distribution group objects	`Addresses` accepts one or more mailbox, contact, mail-enabled user, or distribution group object.
Character Sets	Array of valid character set names	`Character Sets` is a list of names of specific content character sets that can be found in a message. For example, if you wanted to create a rule that checks messages for character sets used in Microsoft ForeFront Online Protection for Exchange, you would use the following list: • Arabic/iso-8859-6 • Chinese/big5 • Chinese/euc-cn • Chinese/euc-tw • Chinese/gb2312 • Chinese/iso-2022-cn • Cyrillic/iso-8859-5 • Cyrillic/koi8-r • Cyrillic/windows-1251 • Greek/iso-8859-7 • Hebrew/iso-8859-8 • Japanese/euc-jp • Japanese/iso-022-jp • Japanese/shift-jis • Korean/euc-kr • Korean/johab • Korean/ks_c_5601-1987

		• Turkish/windows-1254 • Turkish/iso-8859-9 • Vietnamese/tcvn
Classification	Message classification object	`Classification` accepts a message classification object. To see what message classifications are available, you can use the Get-MessageClassification cmdlet.
Domain Name	Any valid SMTP domain name	`Domain Name` is the FQDN for any valid SMTP domain.
EvaluatedUser	Single value of `Sender` or `Recipient`	`EvaluatedUser` is used to determine whether the value specified in the *ManagerAddresses* parameter is the manager of the sender or one of the recipients.
Evaluation	Single value of `Equal` or `NotEqual`	`Evaluation` is used when comparing the Active Directory attributes of the sender and recipients.
FromUserScope	Single value of `InOrganization` or `NotInOrganization`	`FromUserScope` specifies whether the message is sent by a sender who is considered to be inside the organization or external to the organization. The following values can be used: • **InOrganization** A sender is considered to be inside the organization if either of the following conditions is true: ○ The sender is a mailbox, mail-enabled user, distribution group, or public folder that exists in the organization's Active Directory. ○ The domain of the sender is an accepted domain in the Exchange organization, but

		isn't an `ExternalRelay` domain. Also, the message must be sent or received by using an authenticated connection. • **NotInOrganization** A sender is considered to be outside the organization if the sender's domain isn't an accepted domain in the Exchange organization, or is in an accepted domain that is configured as an `ExternalRelay` domain.
Importance	Single value of `High`, `Low`, or `Normal`	`Importance` specifies the message priority.
IPRanges	Array of IP ranges	`IPRanges` is used to specify one or more IP address ranges.
ManagementRelationship	Single value of `Manager` or `DirectReport`	`ManagementRelationship` specifies the relationship between two evaluated users, for example the sender and the recipient. The evaluated user's Active Directory information is located to determine the manager and direct reports.
MessageHeader	Single string	`MessageHeader` accepts a string that can be used to specify the SMTP message header to examine. This property is used together with the `Words` or `Patterns` properties, which specify the value of the header field to match.
MessageType	Single message type name	`MessageType` accepts one of the following message types: • OOF

		• AutoForward • Encrypted • Calendaring • PermissionControlled • Voicemail • Signed • ApprovalRequest • ReadReceipt – Does not match for a read receipt to a meeting request.
Patterns	Array or regular expressions	`Patterns` accepts one or more regular expressions that can be used to match text that follows an identifiable pattern. Enclose the expression in quotation marks (").
SclValue	Single integer	`SclValue` accepts an integer that can be used to match the spam confidence level (SCL) assigned to a message. SCL values range from -1 through 9.
SensitiveInformation	Sensitive Information Types	`SensitiveInformation` accepts the sensitive information types defined in your organization. For a list of built-in sensitive information types in Exchange 2013, see Sensitive information types inventory.
Size	Single integer with quantifier such as KB or MB	`Size` accepts an integer that specifies the size of an email attachment or the overall message. When using the EAC, the value specified is in kilobytes. When using the Shell, you can enter an integer value qualified by one of the following units: • B (bytes) • KB (kilobytes)

		• MB (megabytes) • GB (gigabytes) For example, `20MB`
ToUserScope	One of the following values: • `InOrganization` • `NotInOrganization` • `ExternalPartner` • `ExternalNonPartner`	`ToUserScope` specifies the scope of the recipients. The `InOrganization` and `NotInOrganization` values are evaluated similar to the `FromUserScope` property, but in the context of the recipient. The following is a description of the other possible values: • **ExternalPartner** These domains are configured to send mail to an external domain by using Domain Secure security • **ExternalNonPartner** These represent all other domains that aren't considered `ExternalPartner` domains.
Words	Array of strings	`Words` accepts one string or an array of strings. It's used in all conditions that inspect different parts of a message for specific words or strings. In Exchange 2013, only instances of the word without a prefix or suffix are matched. For example, if you specify the word "contoso", the rule will fire only if an exact match is found. The following variations where the word appears as a suffix, a prefix, or between other characters (other than the space character) aren't

		considered an exact match: • Acontoso • Contosoa • Acontosob The property isn't case-sensitive. The asterisk (*) is treated as a literal character, and not used as a wildcard character.

Conditions and Condition Properties for an Edge Transport Server

The following table lists the conditions that can be used with transport rules in Exchange 2013 Edge Transport Server. These can only be managed in the Exchange Management Shell and not in the EAC. To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell.

In the Exchange Management Shell, the properties are available as parameters of the New-TransportRule and Set-TransportRule cmdlets. Property values are specified after the property name. You don't instantiate conditions and actions by using the Get-TransportRulePredicate or Get-TransportRuleAction cmdlets because these cmdlets only allow you to list the conditions and actions available for use on Edge Transport servers on which the cmdlets are used. The New-TransportRule and Set-TransportRule cmdlets have all the conditions and actions available as parameters, allowing you to create or modify a transport rule using a single command.

Note:

Each condition listed below also has an equivalent exception that can be selected. In the Shell, the conditions that can be used as exceptions start with `ExceptIf`. For example, for the `FromAddressMatches` condition, the parameter that can be used as an exception in transport rule cmdlets is called `ExceptIfFromAddressMatches`. The same condition object contains the logic for use in a transport rule condition and exception. Therefore, when you use the `Get-TransportRulePredicate` cmdlet to list conditions, exceptions aren't listed as separate conditions.

Conditions available only on Edge Transport servers

Condition name in Shell	Condition property type	Description
`SubjectContains`	Words	This condition matches messages that contain the specified words in the **Subject**

		field.
SubjectOrBodyContains	Words	This condition matches messages that contain the specified words in the **Subject** field or message body.
HeaderContains	First property: MessageHeader Second property: Words	This condition matches messages where the value of the specified message header contains the specified words.
FromAddressContains	Words	This condition matches messages that contain the specified words in the **From** field.
AnyOfRecipientAddressContains	Words	This condition matches messages that contain the specified words in the **To**, **Cc**, or **Bcc** fields of the message.
SubjectMatches	Patterns	This condition matches messages where text patterns in the **Subject** field match a specified regular expression.
SubjectOrBodyMatches	Patterns	This condition matches messages where text patterns in the **Subject** field or message body match a specified regular expression.
HeaderMatches	First property: MessageHeader Second property: Patterns	This condition matches messages where the specified message header field contains

		text patterns that match a specified regular expression.
`FromAddressMatches`	Patterns	This condition matches messages that contain text patterns in the **From** field of the messages that match a specified regular expression.
`AnyOfRecipientAddressMatches`	Patterns	This condition matches messages where text patterns in the **To**, **Cc**, or **Bcc** fields of the message match a specified regular expression.
`SCLOver`	SclValue	This condition matches messages with an SCL that's equal to or greater than the value specified.
`AttachmentSizeOver`	Size	This condition matches messages that contain attachments larger than the specified value.
`FromScope`	Scope	This condition matches messages that are sent from the specified scope.
`MessageSizeOver`	Size	This condition matches messages when the message size is larger than or equal to the specified value.

Condition property values for an Edge Transport Server

The following table lists the property types used in transport rule conditions for the Edge Server-

only conditions.

Property types used in Edge server transport rule conditions

Condition type	Valid values	Description
ADAttribute	One of the Active Directory attributes available for use	`ADAttribute` accepts the name of one of the following Active Directory attributes available for use with this property type in transport rules: • `DisplayName` • `FirstName` • `Initials` • `LastName` • `Office` • `PhoneNumber` • `OtherPhoneNumber` • `Email` • `Street` • `POBox` • `City` • `State` • `ZipCode` • `Country` • `UserLogonName` • `HomePhoneNumber` • `OtherHomePhoneNumber` • `PagerNumber` • `MobileNumber` • `FaxNumber` • `OtherFaxNumber` • `Notes` • `Title` • `Department` • `Company` • `Manager` • `CustomAttribute1` - `CutomAttribute15` When you use the Shell to create a transport rule consisting of the `RecipientAddressContains` or `RecipientAddressMatches` predicates, you must specify an attribute name from the preceding list followed by a colon (:) and the word or text pattern you want to match in the specified attribute. The entire

		notation should be enclosed in quotation marks ("). For example, to specify the attribute `City` and the values `San Francisco` or `Palo Alto`, you must use `City:San Francisco, Palo Alto`. You can also specify multiple Active Directory attributes and value pairs. For example, `"City:San Francisco, Palo Alto","Department:Sales, Finance"`. In this case, the recipient's `City` attribute should contain either `San Francisco` or `Palo Alto`, and the `Department` attribute should contain either `Sales` or `Finance`.
Addresses and Addresses2	Array of Active Directory mailbox, contact, or distribution group objects	`Addresses` and `Addresses2` accept a single mailbox, contact, mail-enabled user, or distribution group object.
Classification	Message classification object	`Classification` accepts a message classification object. To specify a message classification object, you must use the Get-MessageClassification cmdlet. For example, use the following command to search for messages with the `ExCompanyInternal` classification and prepend the message subject with CompanyInternal. `New-TransportRule "Rule Name" -HasClassification @(Get-MessageClassification`

		`ExCompanyInternal).Identity -PrependSubject "CompanyInternal"`
EvaluatedUser	Single value of `Sender` or `Recipient`	`ManagementRelationship` accepts an `EvaluatedUser` value for the `ManagerForEvaluatedUser` property. It instructs the Transport Rules agent whether the predicate should inspect a message's sender or the recipient.
Evaluation	Single value of `Equal` or `NotEqual`	`ADAttributeComparison` accepts a value of type `Evaluation` for the `ADComparisonOperator` property. This allows you to compare the specified Active Directory attribute values for the sender and recipient.
FromUserScope	Single value of `InOrganization` or `NotInOrganization`	`FromScope` accepts a scope value of type `FromUserScope`. This specifies whether the message is sent by a sender who is considered to be inside the organization. The following values can be used: • **InOrganization** A sender is considered to be inside the organization if either of the following conditions is true: ○ The sender is a mailbox, mail-enabled user, distribution group, or public folder that exists in the organization's Active Directory. ○ The domain of the sender is an accepted domain in the Exchange organization, but isn't an `ExternalRelay` domain. Also, the message must be sent or received by using an authenticated

		connection. **Note:** To determine whether mail contacts are considered to be inside or outside the organization, the domain part of the sender's address is compared with the configured accepted domains. • **NotInOrganization** A sender is considered to be outside the organization if the sender's domain isn't an accepted domain in the Exchange organization and is an `ExternalRelay` domain.
Importance	Single value of `High`, `Low`, or `Normal`	`Importance` accepts the message priority.
ManagementRelationship	Single value of `Manager` or `DirectReport`	`ManagementRelationship` specifies the relationship between two evaluated users, for example the sender and the recipient. The evaluated user's Active Directory information is located to determine the manager and direct reports.
MessageHeader	Single string	`MessageHeader` accepts a string that can be used to specify the SMTP message header to examine. This property is used together with the `Words` or `Patterns` properties, which specify the value of the header field to match. You don't need to add a colon (`:`) in the header name.
MessageType	Single message type name	`MessageType` accepts one of the following message types: • OOF

		• AutoAccept • AutoForward • Encrypted • Calendaring • PermissionControlled • Voicemail • RSS • Signed • ApprovalRequest • ReadReceipt
Patterns	Array or regular expressions	`Patterns` accepts a regular expression that can be used to match text that follows an identifiable pattern. Enclose the expression in quotation marks (").
SclValue	Single integer	`SclValue` accepts an integer that can be used to match the spam confidence level (SCL) assigned to a message. SCL values range from -1 through 9.
Size	Single integer with quantifier such as KB or MB	`Size` accepts an integer that specifies the size of an email attachment. You can enter an integer value qualified by one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) For example, `20MB`
ToUserScope	One of the following values: • `InOrganization` • `NotInOrganization` • `ExternalPartner` • `ExternalNonPartner`	`SentToScope` accepts a scope value of type `ToUserScope`. The `InOrganization` and `NotInOrganization` values are evaluated similar to the `FromUserScope` property, but in the context of the recipient. The

		following is a description of the other possible values: • **ExternalPartner** These domains are configured to send mail to an external domain by using Domain Secure security • **ExternalNonPartner** These represent all other domains that aren't considered `ExternalPartner` domains.
Words	Array of strings	`Words` accepts one string or an array of strings. It's used in all predicates that inspect different parts of a message for specific words or strings. Only instances of the word without a prefix or suffix are matched. For example, if you specify the word "contoso", the rule will fire only if an exact match is found. The following variations where the word appears as a suffix, a prefix, or between other characters (other than the space character) aren't considered an exact match: • Acontoso • Contosoa • Acontosob The property isn't case-sensitive. The asterisk (*) is treated as a literal character, and not used as a wildcard character.

For more information

Transport rules

Transport rule actions

Transport rule procedures

Transport Rule Conditions (Predicates) for Exchange Online

Transport rule conditions (predicates) for Exchange Online Protection

New-TransportRule

Transport rule actions

Exchange Server 2013 > Messaging policy and compliance > Transport rules >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-12*

Transport rule actions let you apply messaging policies to email messages flowing through your organization that match conditions and exceptions defined in Transport rule conditions (predicates). For example, you could define a transport rule action to forward an email message to an approving manager, or add a disclaimer or personalized signature.

Looking for:

- Management tasks related to transport rules? See Manage Transport Rules
- Exchange Online version of this topic? See **Transport rule actions**
- Exchange Online Protection version of this topic? See **Transport rule actions**

Contents

Transport rule actions and properties

Each transport rule action contains the action, and any information needed for the action. For example, when you use the **Redirect the Message to** action, you must specify the email address of the person you're redirecting the message to. Sometimes two pieces of information are needed. For example, when you add a disclaimer, you need the disclaimer text plus information about what to do if the disclaimer can't be sent.

Here's a list of the information required for each action in Exchange 2013. When you add these

actions using the Exchange admin center, you're prompted for the required information. When you add them using the Shell, you must specify them on the command line. Valid values for each property are described in Property values.

Action name in Exchange admin center	Action name in Shell	Properties	Description
Forward the message for approval to	`ModerateMessageByUser`	`Addresses`	Forwards the message to the specified moderators as an attachment wrapped in an approval request. See Forward a message to a manager for approval for more information.
Forward to the sender's manager for approval	`ModerateMessageByManager`	Not applicable	Forwards the message to the sender's manager for approval, if the manager attribute is populated in Active Directory. **Important:** If the sender's manager attribute isn't populated in Active Directory, the message is delivered to recipients without moderation.
Redirect the message to	`RedirectMessageTo`	`Addresses`	Redirects the email message to one or more specified recipients. The message isn't delivered to the original recipients, and no notification is sent to the sender or the original recipients.
Reject the message	`RejectReason`	`RejectReason`	Deletes the email message

with the explanation			and sends a non-delivery report to the sender with the specified text as the rejection reason. The recipient doesn't receive the message or notification.
Reject the message with the enhanced status code	`RejectMessageEnhancedStatusCode`	`EnhancedStatusCode`	Deletes the email message and sends a non-delivery receipt to the sender with the specified status code. The recipient doesn't receive the message or notification.
Delete the message without notifying anyone	`DeleteMessage`	Not applicable	Deletes the email message without sending a notification to either the recipient or the sender.
Bcc the message to	`BlindCopyTo`	`Addresses`	Adds one or more recipients as blind carbon copy (Bcc) recipients. The original recipients aren't notified and can't see the Bcc addresses.
Add these recipients to the To box	`AddToRecipient`	`Addresses`	Adds one or more recipients to the To field of the message. The original recipients can see the additional address.
Cc the message to	`CopyTo`	`Addresses`	Adds one or more recipients to the carbon

			copy (Cc) field of the message. The original recipients can see the Cc address.
Add the sender's manager as a recipient type	`AddManagerAsRecipientType`	`AddedRecipientType`	Adds the sender's manager, if defined in the manager attribute in Active Directory, as the specified recipient type.

Append the disclaimer	`ApplyHtmlDisclaimerText`
	`ApplyHtmlDisclaimerTextLocation`
	`ApplyHtmlDisclaimerFallbackAction`
		First property: `DisclaimerText`
		Second property: `FallbackAction`
			Applies an HTML disclaimer to the message. There are three parameters in the Shell that correspond to this action. When you append the disclaimer, the property `ApplyHtmlDisclaimerTextLocation` is set to **Append**.

Prepend the disclaimer	`ApplyHtmlDisclaimerText`
	`ApplyHtmlDisclaimerTextLocation`
	`ApplyHtmlDisclaimerFallbackAction`
		First property: `DisclaimerText`
		Second property: `FallbackAction`
			Applies an HTML disclaimer to the message. There are three parameters in the Shell that correspond to this action. When you prepend the disclaimer, the property `ApplyHtmlDisclaimerTextLocation` is set to **Prepend**.

Remove this header	`RemoveHeader`	`MessageHeader`	Removes the specified message header from a message.
Set the message	`SetHeaderName` `SetHeaderValue`	First property: `MessageHeader`	Creates a new message

header to this value		Second property: `HeaderValue`	header or modifies an existing message header and sets the value of that message header to the specified value. There are two parameters in the Shell that correspond to this action.
Apply a message classification	`ApplyClassification`	`Classification`	Applies a message classification to the message.
Set the spam confidence level (SCL) to	`SetScl`	`SclValue`	This action sets the spam confidence level (SCL) on an email message.
Prepend the subject of the message with	`PrependSubject`	`Prefix`	Prepends the Subject of the message with the specified text.
Apply rights protection to the message with	`ApplyRightsProtectionTemplate`	`RMSTemplateIdentity`	Applies the specified Rights Management Services (RMS) template to the message.

Require TLS encryption	`RouteMessageOutboundRequireTls`	Either of the following:
		`Not applicable`
		`Applicable`
			Forces the outbound messages to be routed over TLS encrypted connections.

Notify the sender with a Policy Tip	`NotifySender`	`NotifySenderType`	Notifies the sender when the message goes against a DLP policy configured in the organization. • When you use this action, you must use the

			`MessageContainsDataClassification` (**The message contains sensitive information**) condition. • You can also use the following conditions: ○ `SentTo` (**The recipient is**) ○ `SentToScope` (**The recipient is located**) ○ `From` (**The sender is**) ○ `FromMemberOf` (**The sender is a member of**) ○ `FromScope` (**The sender is located**) • Some actions can't be combined with `NotifySender`: ○ `RejectMessageReasonText` (**Reject the message and include an explanation**) ○ `RejectMessageEnhancedStatusCode` (**Reject the message with the enhanced status code of**) ○ `DeletedMessage` (**Delete the message without notifying anyone**)
Generate incident report and send it to	`GenerateIncidentReport`	First property: `Addresses` Second property: `IncidentReportContent`	Generates an incident report and sends it to the specified recipients.
Stop processing more rules	`StopRuleProcessing`	Not applicable	Specifies whether the processing of subsequent rules should be stopped for

			this message.

Property values

Each property that you use to define a transport rule action requires a value. Here's a list of values for each action property in Exchange 2013.

Property	Valid values	Description
`AddedRecipientType`	One of the following values: • **To** • **Cc** • **Bcc** • **Redirect**	`AddedRecipientType` accepts a single value: • `To`, `Cc`, and `Bcc` values are self-explanatory and correspond to the addressing fields of email messages. • `Redirect` delivers the message only to the specified recipient. The message isn't delivered to any of the original recipients.
`Addresses`	Either a single or an array of valid recipients.	`Addresses` accepts an array of mailbox, mail-enabled contact, mail-enabled user, or distribution group objects.
`Classification`	Single message classification object	`Classification` accepts a single message classification object. To see what message classification objects are available, you can use the Get-MessageClassification cmdlet.
`DisclaimerText`	HTML string	`DisclaimerText` can be any text you want to insert to the message as a disclaimer. HTML tags and cascade style sheet (CSS) tags.
`EnhancedStatusCode`	Single DSN code of `5.7.1`, or any value from `5.7.10` through `5.7.999`	`EnhancedStatusCode` specifies the DSN code and related DSN message to display to the senders of messages rejected by a Transport rule action. The DSN message associated with the specified DSN status code is displayed in the user

		information portion of the NDR displayed to the sender. The specified DSN code must be an existing default DSN code or a customized DSN status code that you can create by using the **New-SystemMessage** cmdlet.
`FallbackAction`	One of the following values: • `Wrap` • `Ignore` • `Reject`	`FallbackAction` specifies what the Transport rule should do if a disclaimer can't be applied to a message. There can be cases where the contents of a message can't be altered, for example if a message is encrypted. The default fallback action is `Wrap`. The following list describes each fallback action: • **Wrap** If the disclaimer can't be inserted into the original message, Exchange encloses, or *wraps*, the original message in a new message envelope. Then the disclaimer is inserted into the new message. **Important:** If an original message is wrapped in a new message envelope, subsequent Transport rules are applied to the new message envelope, and not to the original message. Therefore, you must configure Transport rules with disclaimer actions that wrap original messages in a new message body with a lower priority than other Transport rules. **Note:** If the original message can't be wrapped in a new message envelope, the original message

		isn't delivered. The sender of the message receives an NDR that explains why the message wasn't delivered. • **Ignore** If the disclaimer can't be inserted into the original message, Exchange lets the original message continue unmodified. No disclaimer is added. • **Reject** If the disclaimer can't be inserted into the original message, Exchange doesn't deliver the message. The sender of the message receives an NDR that explains why the message wasn't delivered.
`HeaderValue`	Single string	`HeaderValue` accepts a single string that's applied to the header specified header.
`IncidentReportContent`	One of the following values: • Sender • Recipients • Subject • CC • BCC • Severity • Override • RuleDetections • FalsePositive • DataClassifications • IdMatch • AttachOriginalMail	`IncidentReportContent` specifies the list of properties of the original message to be included in the incident report. You can choose to include any combination of these properties. In addition to the properties you specify, the message ID is always included in the incident report. The following list describes these properties: • **Sender**, **Subject**, and **AttachOriginalMail** Self-explanatory values. The first two correspond to the message properties with the

		same name. The third property specifies that the entire message that triggered the rule is attached to the incident report. • **Recipients**, **CC**, and **BCC** Designate the recipients the message was sent to, and correspond to the To, Cc, and Bcc boxes respectively. For each of these properties, only the first 10 recipients are included in the incident report. If the message was addressed to more than 10 recipients, the report will also include the count of additional recipients. For example, if the message included 22 people in the To: box, the incident report will list the first 10 of these recipients and specify that there were 12 additional recipients. • **Severity** Specifies the audit severity of the rule that was triggered. If the message was processed by more than one rule, the highest severity is included in the incident report. • **Override** Specifies the override if the sender has chosen to override a Policy Tip. If the sender has provided a justification, the first 100

		characters of the justification are also included. • **RuleDetections** Specifies the list of rules that the message triggered. • **FalsePositive** Specifies the false positive if the sender marked the message as a false positive for a Policy Tip. • **DataClassifications** Specifies the list of sensitive information types detected in the message. • **IdMatch** Specifies the sensitive information type detected, the exact matched content from the message, and the 150 characters before and after the matched sensitive information.
`MessageHeader`	Single string	`MessageHeader` accepts a string that can be used to specify the SMTP message header to modify.
`NotifySenderType`	One of the following values: • NotifyOnly • RejectMessage • RejectUnlessFalsePositiveOverride • RejectUnlessSilentOverride • RejectUnlessExplicitOverride	`NotifySenderType` specifies the type of Policy Tip the sender will receive if a message they sent goes against a DLP policy configured in the organization. The following list describes each notification type: • **NotifyOnly** Notifies sender, but delivers the message normally. • **RejectMessage** Notifies

		sender, and rejects the message. • **RejectUnlessFalsePositiveOverride** RejectsThe message unless the sender has marked it with a false positive. • **RejectUnlessSilentOverride** Rejects the message unless the sender has chosen to override the policy restriction. • **RejectUnlessExplicitOverride** This is similar to **RejectUnlessSilentOverride** type, but the sender also provides a justification for overriding the policy restriction.
`Prefix`	Single string	`Prefix` accepts a string that's prepended to the subject of the email message. **Tip:** To prevent the string that's specified with the `Prefix` Transport rule action from being added to the subject every time that a reply to the message encounters the Transport rule, add the `SubjectContains` exception to the Transport rule. The `SubjectContains` exception should contain the string that you specified with the `Prefix` Transport rule action. If you add the `SubjectContains` exception to the Transport rule, the Transport rule doesn't add another instance of the `Prefix` string to the subject if the `Prefix` string already appears in

		the subject.
`RejectReason`	Single string	`RejectReason` accepts a string that's used to populate the administrator information portion of the NDR returned to the email sender if an email message is rejected.
`RMSTemplateIdentity`	RMS Template identity	`RMSTemplateIdentity` accepts an RMS Template identity. You can get a list of RMS templates available on an Active Directory RMS server in the Active Directory forest using the **Get-RMSTemplate** cmdlet.
`SclValue`	Single integer	`SclValue` accepts a single integer from `-1` through `9`, which is used to configure the SCL of the email message.

For more information

Manage Transport Rules

Transport rules

Transport rule conditions (predicates)

Transport rule actions for Exchange Online

Transport rule actions for Exchange Online Protection

Add an email disclaimer, legal disclaimer, common signature, or email footer or header

Using transport rules to inspect message attachments

Exchange Server 2013 > Messaging policy and compliance > Transport rules >

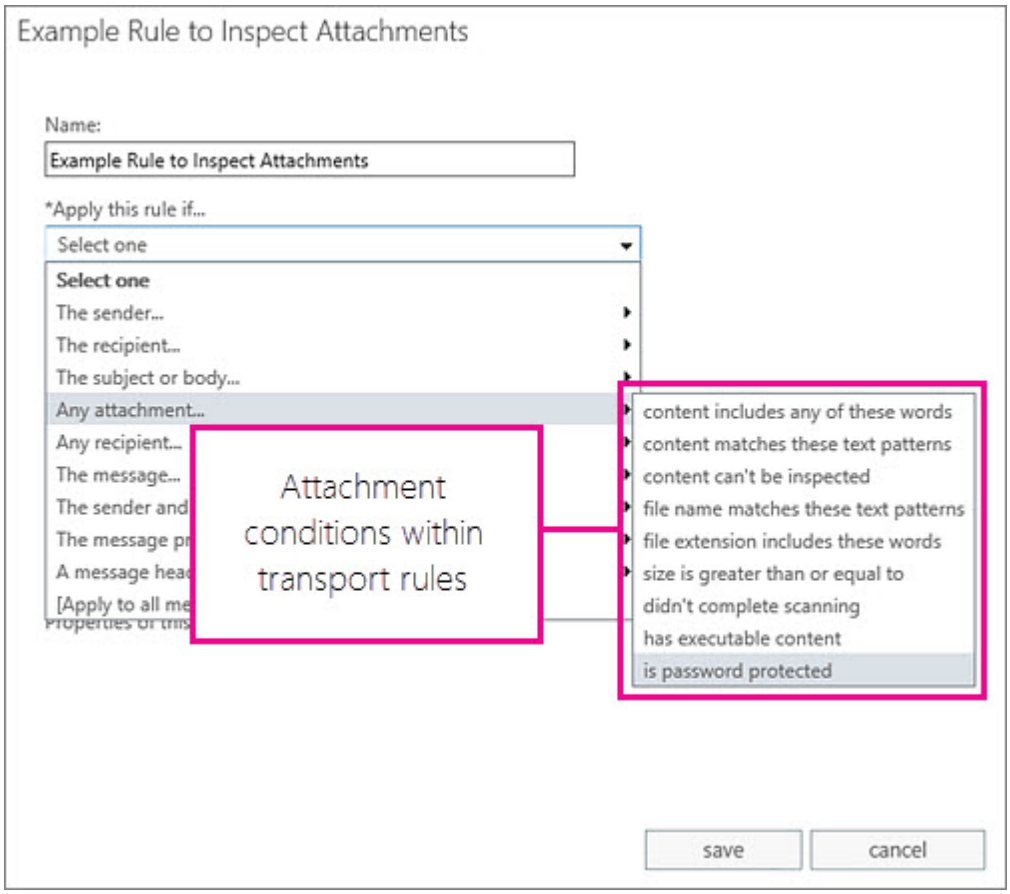
Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-06-10*

You can inspect email attachments in your organization by setting up transport rules. Exchange offers transport rules that provide the ability to examine email attachments as a part of your messaging security and compliance needs. When you inspect attachments, you can then take action on the messages that were inspected based on the content or characteristics of those attachments. Here are some attachment-related tasks you can do by using transport rules:

- Search files in compressed attachments such as .zip and .rar files and, if there's any text that matches a pattern you specify, add a disclaimer to the end of the message.
- Inspect content within attachments and, if there are any keywords you specify, redirect the message to a moderator for approval before it's delivered.
- Check for messages with attachments that can't be inspected and then block the entire message from being sent.
- Check for attachments that exceed a certain size and then notify the sender of the issue if you choose to prevent the message from being delivered.
- Create notifications that alert users if they send a message that has matched a transport rule.

Exchange administrators can create transport rules by going to **Exchange Admin Center** > **Mail flow** > **Rules**. You need to be assigned permissions before you can perform this procedure. After you start to create a new rule, you can see the full list of attachment-related conditions by clicking **More options** > **Any attachment** under **Apply this rule if**. The attachment-related options are shown in the following diagram.



For more information about transport rules, including the full range of conditions and actions that you can choose, see Transport rules. Exchange Online Protection (EOP) and hybrid customers can benefit from the transport rules best practices provided in **Best practices for configuring EOP**. If

you're ready to start creating rules, see Manage Transport Rules.

Inspect the content within attachments

You can use the transport rule conditions in the following table to examine the content of attachments to messages. For these conditions, only the first 150 KB of an attachment is inspected. In order to start using these conditions when inspecting messages, you need to add them to a transport rule. Learn about creating or changing rules at Manage Transport Rules.

Condition name in EAC	Condition name in the Shell	Description
Any attachment content includes any of these words	`AttachmentContainsWords`	This condition matches messages with supported file type attachments that contain a specified string or group of characters.
Any attachment content matches these text patterns	`AttachmentMatchesPatterns`	This condition matches messages with supported file type attachments that contain a text pattern that matches a specified regular expression.

The Exchange Management Shell names for the conditions listed here are parameters that require the `TransportRule` cmdlet.

Learn more about the cmdlet at New-TransportRule.

Learn more about property types for these conditions at Conditions and Condition Properties for a Mailbox Server.

Transport rules can inspect only the content of supported file types. If the transport rules agent encounters an attachment that isn't in the list of supported file types, the `AttachmentIsUnsupported` condition is triggered. The supported file types are listed in the following section. Any file not listed will trigger the `AttachmentIsUnsupported` condition.

Compressed archive files

If the message contains a compressed archive file such as a .zip or .cab file, the transport rules agent will inspect the files contained within that attachment. Such messages are processed in a manner similar to messages that have multiple attachments. The properties of compressed archive files aren't inspected. For example, if the container file type supports comments, that field isn't

inspected.

Supported file types for transport rule content inspection

The following table lists the file types supported by transport rules. The system automatically detects file types by inspecting file properties rather than the actual file name extension, thus helping to prevent malicious hackers from being able to bypass transport rule filtering by renaming a file extension. A list of file types with executable code that can be checked within the context of transport rules is listed later in this topic.

Category	File extension	Notes
Office 2013, Office 2010, and Office 2007	.docm, .docx, .pptm, .pptx, .pub, .one, .xlsb, .xlsm, .xlsx	Microsoft OneNote and Microsoft Publisher files aren't supported by default. You can enable support for these file types by using IFilter integration. For more information, see Register Filter Pack IFilters with Exchange 2013. The contents of any embedded parts contained within these file types are also inspected. However, any objects that aren't embedded—for example, linked documents—aren't inspected.
Office 2003	.doc, .ppt, .xls	None
Additional Office files	.rtf, .vdw, .vsd, .vss, .vst	None
Adobe PDF	.pdf	None
HTML	.html	None
XML	.xml, .odp, .ods, .odt	None

Text	.txt, .asm, .bat, .c, .cmd, .cpp, .cxx, .def, .dic, .h, .hpp, .hxx, .ibq, .idl, .inc, inf, .ini, inx, .js, .log, .m3u, .pl, .rc, .reg, .txt, .vbs, .wtx	None
OpenDocument	.odp, .ods, .odt	No parts of .odf files are processed. For example, if the .odf file contains an embedded document, the contents of that embedded document aren't inspected.
AutoCAD Drawing	.dxf	AutoCAD 2013 files aren't supported.
Image	.jpg, .tiff	Only the metadata text associated with these image files is inspected. There is no optical character recognition.

Inspect the file properties of attachments

The following transport rule conditions inspect the properties of a file that is attached to a message. In order to start using these conditions when inspecting messages, you need to add them to a transport rule. A list of supported file types with executable code that can be checked within the context of transport rules is listed here. For more information about creating or changing rules, see Manage Transport Rules.

Condition name in EAC	Condition name in the Shell	Description
Any attachment file name matches these text patterns	`AttachmentNameMatchesPatterns`	This condition matches messages with supported file type attachments when those attachments have a name that contains the characters you specify.
Any attachment file	`AttachmentExtensionMatchesWords`	This condition matches

extension includes these words		messages with supported file type attachments when the file name extension matches what you specify.
Any attachment size is greater than or equal to	`AttachmentSizeOver`	This condition matches messages with supported file type attachments when those attachments are larger than the size you specify.
Any attachment didn't complete scanning	`AttachmentProcessingLimitExceeded`	This condition matches messages when an attachment is not inspected by the transport rules agent.
Any attachment has executable content	`AttachmentHasExecutableContent`	This condition matches messages that contain executable files as attachments. The supported file types are listed here.
Any attachment is password protected	`AttachmentIsPasswordProtected`	This condition matches messages with supported file type attachments when those attachments are protected by a password.

The Exchange Management Shell names for the conditions listed here are parameters that require the `TransportRule` cmdlet.

Learn more about the cmdlet at New-TransportRule.

Learn more about property types for these conditions at Conditions and Condition Properties for a Mailbox Server.

Supported executable file types for transport rule

inspection

The transport agent uses true type detection by inspecting file properties rather than merely the file extensions. This helps to prevent malicious hackers from being able to bypass your rule by renaming a file extension. The following table lists the executable file types supported by these conditions. If a file is found that is not listed here, the `AttachmentIsUnsupported` condition is triggered.

Type of file	Native extension
Self-extracting archive file created with the WinRAR archiver.	.rar
32-bit Windows executable file with a dynamic link library extension.	.dll
Self-extracting executable program file.	.exe
Java archive file.	.jar
Uninstallation executable file.	.exe
Program shortcut file.	.exe
Compiled source code file or 3-D object file or sequence file.	.obj
32-bit Windows executable file.	.exe
Microsoft Visio XML drawing file.	.vxd
OS/2 operating system file.	.os2
16-bit Windows executable file.	.w16
Disk-operating system file.	.dos
European Institute for Computer Antivirus Research standard antivirus test file.	.com
Windows program information file.	.pif
Windows executable program file.	.exe

Extending the number of supported file types

The supported file types listed in this topic can be revised at any time using IFilter integration. For more information, see Register Filter Pack IFilters with Exchange 2013.

The file types you add using this process become supported file types and no longer trigger the `AttachmentIsUnsupported` condition.

Data loss prevention policies and attachment transport rules

To help you manage important business information in email, you can include any of the attachment-related conditions along with the rules of a data loss prevention (DLP) policy. For example, you might want to allow messages with passport numbers to be sent but only if the passport numbers are in a password-protected attachment. To accomplish this, do the following:

- Create a DLP policy that inspects mail for passport-related sensitive information. Learn more at DLP procedures.
- Add the **Any attachment is password protected** exception in the **Except if...** transport rule area.
- Define an action to take on mail that contains passport numbers that are not in the protected file.

DLP policies and attachment-related conditions can help you enforce your business needs by defining those needs as transport rule conditions, exceptions, and actions. When you include the sensitive information inspection in a DLP policy, any attachments to messages are scanned for that information only. However, attachment-related conditions such as size or file type are not included until you add the conditions listed in this topic. DLP is not available with all versions of Exchange; learn more at Data loss prevention.

For more information

Data loss prevention

Transport rules

Transport rule conditions (predicates)

Using transport rules to inspect message attachments in Exchange Online

Organization-wide disclaimers, signatures, footers, or headers

Applies to: *Exchange Online Protection*

Topic Last Modified: *2014-01-31*

You can add an email disclaimer, legal disclaimer, disclosure statement, signature, or other information to the top or bottom of email messages that enter or leave your organization. This might be needed for legal, business, or regulatory requirements, to identify potentially unsafe e-mail messages, or for other reasons unique to your organization.

To set up a disclaimer, you create a transport rule that includes the conditions, such when the sender is in a specific group or when the message includes specific text patterns, and the text to add. To apply multiple disclaimers to a single email message, you use multiple transport rules.

Important:
• If you want the information to be added only to outgoing messages, you must add a condition such as recipients located outside the organization. By default, transport rules are applied to both incoming and outgoing messages.

Contents

Looking for procedures? See Add an email disclaimer, legal disclaimer, common signature, or email footer or header.

Examples

Here are a few ideas for how to use disclaimers.

Type	Sample text added
Legal – outgoing messages	This email and any files transmitted with it are confidential and intended solely for the use of the individual or entity to whom they are addressed. If you have received this email in error, please notify the system manager.
Legal – incoming messages	Employees are expressly required not to make

	defamatory statements and not to infringe or authorize any infringement of copyright or any other legal right by email communications. Employees who receive such an email must notify their supervisor immediately.
Notice that message was sent to an alias	This message was sent to the Sales discussion group.
Signature – pulls in data for each employee	Kathleen Mayer Sales Department Contoso www.contoso.com kathleen@contoso.com cell: 111-222-1234
Advertisement	Click here for March specials

The examples in this article are not intended for use as-is. Modify them for your needs.

Scoping your disclaimer

As you work on your disclaimers, consider which messages they should apply to. For example, you might want different disclaimers for internal and external messages or for messages sent by users in specific departments. To make sure only the first message in a conversation gets a disclaimer, add an exception that looks for unique text in your disclaimer.

Here are some examples of the conditions and exceptions you can use.

Description	Conditions and exceptions in EAC	Conditions and exceptions in the Shell
Outside your organization, if the original message doesn't include text from your disclaimer, such as "CONTOSO LEGAL NOTICE"	Condition: **The recipient is located** > **Outside the organization** Exception: **The subject or body** > **Subject or body matches these text patterns** > **CONTOSO LEGAL NOTICE**	`-FromScope NotInOrgani`

Incoming messages with executable attachments	Condition 1: **The sender is located** > **Outside the organization** Condition 2: **Any attachment** > **has executable content**	`-FromScope NotInOrgani`
Sender is in the marketing department	Condition: **The sender** > **is a member of this group** > **group name**	`-FromMemberOf "Marketi`
Every message that comes from an external sender to the sales discussion group	Condition 1: **The sender is located** > **Outside the organization** Condition 2: **The message** > **To or Cc box contains this person** > **group name**	`-FromScope NotInOrgani`
Prepend an advertisement to outgoing messages for one month	Condition 1: **The recipient is located** > **Outside the organization** Specify the dates at the bottom of the **New rule** dialog.	-ApplyHtmlDisclaimerLocation 'Prepend' -SentToScope 'NotInOrganization' –ActivationDate '03/1/2014' –ExpiryDate '03/31/2014'

For a complete list of transport rule conditions you can use to target the disclaimer, see one of the following:

- **Transport Rule Conditions (Predicates)** (Exchange Online)
- Transport rule conditions (predicates) (Exchange 2013)
- **Transport Rule Conditions (Predicates)** (Exchange Online Protection)

Formatting your disclaimer

You can format your disclaimer as needed. Here's what can be included in your disclaimer text.

Type of information	Description
Text	The maximum length is 5,000 characters, including any HTML tags and inline Cascading Style Sheets (CSS).

HTML and inline CSS	You can use HTML and inline CSS styles to format the text. For example, use the `<HR>` tag to add a line before the disclaimer. HTML in a disclaimer is ignored if the disclaimer is added to a plain text message.
Add Images	Use the `<IMG>` tag to point to an image available on the internet. For example, `<IMG src="http://contoso.com/images/companylogo.gif" alt="Contoso logo">` Keep in mind that Outlook Web App and Outlook block external web content, including images, by default. Users may need to perform a specific action if they want to view the blocked external content. This means that images added by using the `IMG` tag may not be visible by default. We recommend that you test a disclaimer with `IMG` tags in the email clients your recipients are likely to use to make sure it displays well.
Add information for personalized signatures	If you want everyone to have signatures formatted the same way with the same information, you can add unique information for each employee, such as `DisplayName`, `FirstName`, `LastName`, `PhoneNumber`, `Email`, `FaxNumber`, and `Department`. This information must be enclosed in two percent signs (%%) on each side of the information. For example, to use `DisplayName`, you must use **%% DisplayName%%** in your disclaimer. When a disclaimer rule is triggered, the corresponding values for that user are inserted. The data comes from the sender's Active

	Directory user account (for on-premises Exchange Server), or from the sender's Office 365 account for Exchange Online. For a complete list of attributes that can be used in disclaimers and personalized signatures, see the description for the `ADAttribute` property in Transport rule conditions (predicates) (Exchange Server), **Transport Rule Conditions (Predicates)** (Exchange Online), or **Transport rule conditions (predicates)** (Exchange Online Protection).

For example, here's an example of an HTML disclaimer that includes a signature, an `IMG` tag, and embedded CSS.

```
<div style="font-size:9pt;  font-family: 'Calibri',sans-
serif;">
%%displayname%%</br>
%%title%%</br>
%%company%%</br>
%%street%%</br>
%%city%%, %%state%% %%zipcode%%</div>
 </br>
<div style="background-color:#D5EAFF; border:1px dotted
#003333; padding:.8em; ">
<div><img alt="Fabrikam"  src="http://fabrikam.com/images/
fabrikamlogo.png"></div>
<span style="font-size:12pt;  font-family: 'Cambria','times
new roman','garamond',serif; color:#ff0000;">HTML
Disclaimer Title</span></br>
<p style="font-size:8pt; line-height:10pt; font-family:
'Cambria','times roman',serif;">This message contains
confidential information and is intended only for the
individual(s) addressed in the message. If you are not the
named addressee, you should not disseminate, distribute, or
copy this e-mail. If you are not the intended recipient,
```

```
you are notified that disclosing, distributing, or copying
this e-mail is strictly prohibited.  </p>
<span style="padding-top:10px; font-weight:bold;
color:#CC0000; font-size:10pt; font-family:
'Calibri',Arial,sans-serif; "><a href="http://
www.fabrikam.com">Fabrikam, Inc. </a></span></br></br>
</div>
```

Fallback options if the disclaimer can't be added

Some messages, such as encrypted messages, prevent Exchange from modifying the content of the original message. You can control how your organization handles these messages. You specify whether to wrap a message that can't be modified in a message envelope that contains the disclaimer, reject the message if a disclaimer can't be added, or ignore the disclaimer action and deliver the message without a disclaimer.

The following list describes each fallback action:

- **Wrap** If the disclaimer can't be inserted into the original message, Exchange encloses, or "wraps," the original message in a new message envelope. Then the disclaimer is inserted into the new message. If the original message can't be wrapped in a new message envelope, the original message is not delivered. The sender of the message receives a non-delivery report (NDR) that explains why the message was not delivered.

Important:
If an original message is wrapped in a new message envelope, subsequent transport rules are applied to the new message envelope, not to the original message. Therefore, you must configure transport rules with disclaimer actions that wrap original messages in a new message body after you configure other transport rules.

- **Reject** If the disclaimer can't be inserted into the original message, Exchange doesn't deliver the message. The sender of the message receives an NDR that explains why the message wasn't delivered.
- **Ignore** If the disclaimer can't be inserted into the original message, Exchange delivers the original message unmodified. No disclaimer is added.

For more information

Add an email disclaimer, legal disclaimer, common signature, or email footer or header

Transport rules (Exchange Server 2013)

Transport rules (Exchange Online)

Transport rules (Exchange Online Protection)

Transport rule procedures

Exchange Server 2013 > Messaging policy and compliance > Transport rules >

Applies to: *Exchange Server*

Topic Last Modified: *2014-01-31*

You can begin using transport rules by using the following procedures. To learn about concepts and objectives for transport rules, see Transport rules.

Add an email disclaimer, legal disclaimer, common signature, or email footer or header Information to help you set up a legal disclaimer, email disclaimer, consistent signature, email header, or email footer by using transport rules.

Create a domain or user-based safe sender or blocked sender list using transport rules Information to help you create domain or user-based safe sender and blocked sender lists by using transport rules.

Register Filter Pack IFilters with Exchange 2013 Information to help you register additional file types for attachments so that transport rules that apply to attachments can scan these file types.

Manage Transport Rules Information to help you create, view, modify, enable, disable, or remove a transport rule, and information about importing and exporting transport rule collections.

Add an email disclaimer, legal disclaimer, common signature, or email footer or header

Messaging policy and compliance > Transport rules > Transport rule procedures >

Applies to: *Exchange Online Protection*

Topic Last Modified: *2014-01-31*

You can add an HTML or text legal disclaimer, disclosure statement, signature, or other information to the top or bottom of email messages that enter or leave your organization. To do this, you create a transport rule that adds the required information when specific conditions are met.

For examples and information about how to scope and format disclaimers, signatures, and other additions to email messages, see Organization-wide disclaimers, signatures, footers, or headers.

Important:

- If you want the information to be added only to outgoing messages, you must add a condition such as recipients located outside the organization. By default, transport rules are applied to incoming and outgoing messages.
- To avoid multiple disclaimers being added in an email conversation, add an exception that looks for unique text in your disclaimer. That way the disclaimer is only added to the original message.
- Test each disclaimer. When you create the transport rule, you have the option to start using it immediately (**Enforce**), or to test it first and view the results in the messaging log. We recommend testing all transport rules prior to setting them to **Enforce**.

What do you need to know before you begin?

- Estimated time to complete each procedure: 15 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport Rules" entry in the Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to add a disclaimer or other email header or footer

1. Go to **Mail flow** > **Rules**.
2. Click **Add +**, and then click **Apply disclaimers**.
3. In **New rule**, name the rule.
4. In the **Apply this rule if** box, select the conditions for displaying the disclaimer. For example, select **The recipient is located** condition, and then select **Outside the organization**.
   - If you want this rule to apply to every message that enters or leaves your organization, select **[Apply to all messages]**.
   - If you have additional conditions or exceptions, select **More options** at the bottom of the page.

For example, to add an exception that looks for text from your disclaimer:

a. Select **Add exception**.

b. Select **The subject or body**, select **Subject or body matches these text patterns**, and then specify the words or phrases in your disclaimer.

Tip:

To put your disclaimer at the top of the email message:

- Select **More options**.
- In the **Do the following list**, select **Apply a disclaimer to the message** > **prepend a disclaimer**.

1. Next to the **Do the following** box, select **Enter text** to enter the text of your disclaimer. For information about what can be added, see Enter and format your disclaimer.
2. Click **Select one**, and select one of the Fallback options if the disclaimer can't be added.
3. Specify the audit severity level to assign the severity level that appears in the message log.
4. Select the mode for the rule. Select **Enforce** to turn on the disclaimer immediately, or select **Test without Policy Tips** to put a message in the message tracking log instead of adding the disclaimer.
5. Click **Save** to complete creating the rule.

For more examples of how to scope your disclaimer, see Scoping your disclaimer.

Use the Shell to add a disclaimer or other email header or footer

Use New-TransportRule to create a disclaimer, with parameters for each condition and exception. For detailed parameter information, see Transport rule conditions (predicates) for Exchange Server, **Transport Rule Conditions (Predicates)** for Exchange Online, or **Transport rule conditions (predicates)** for Exchange Online Protection.

This example creates a new transport rule that adds a disclaimer with an image to the end of all email messages sent outside the organization.

```
New-TransportRule -Name ExternalDisclaimer -SentToScope
'NotInOrganization'  -ApplyHtmlDisclaimerText
"<h3>Disclaimer Title</h3><p>This is the disclaimer text.</
p><img alt='Contoso logo' src='http://www.contoso.com/
images/logo.gif'>"
```

This example creates a new transport rule that adds an advertisement for one month to the beginning of all outgoing messages.

```
New-TransportRule -Name MarchSpecial -Enabled $true -
SentToScope 'NotInOrganization' -
ApplyHtmlDisclaimerLocation 'Prepend' -ActivationDate
```

```
'03/1/2014' -ExpiryDate '03/31/2014'-
ApplyHtmlDisclaimerText "<table align=center width=200
border=1 bordercolor=blue bgcolor=green cellpadding=10
cellspacing=0><tr><td nowrap><a href=http://
www.contoso.com/marchspecials.htm>Click to see March
specials</a></td></tr></table>"
```

For more examples of how to scope your disclaimer, see Scoping your disclaimer.

How do you know this worked?

To verify that your disclaimer works as expected, do the following:

- Send yourself both a plain text email and an HTML email that match the conditions and exceptions you defined, and verify that the text appears as you intended.
- If you added an exception to avoid adding the disclaimer to successive messages in a conversation, forward your test messages to yourself to make sure that they don't get an extra copy of the disclaimer.
- Send yourself some messages that should not get the disclaimer and verify that the disclaimer is not included.

For more information

After you configure a disclaimer or email header or footer, see Manage Transport Rules for information about how to view, modify, enable, disable, or remove a rule.

Create a domain or user-based safe sender or blocked sender list using transport rules

Messaging policy and compliance > Transport rules > Transport rule procedures >

Applies to: *Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-01-27*

If you want to be sure that you receive mail from a particular sender, because you trust them and their messages, you can create an Exchange transport rule that serves as a domain or user-based safe sender list. Any messages sent from the domain or user you specify bypasses spam filtering.

Conversely, if you want to block messages sent from a particular domain or user, you can create an

Exchange transport rule that serves as a domain or user-based blocked sender list. Any messages sent from the domain or user you specify will be blocked.

Tip:
A domain-based list isn't as secure as an IP address-based list, because domains can be spoofed. Also, if the sending IP address is on a Block list, it will still be blocked even if filtering for the domain or user is being bypassed. This is because a transport rule on a domain or user does not override the global IP Block list. We recommend using an IP address-based list in most cases. To create an IP address-based list, you can use the IP Allow list or IP Block list in the connection filter. Any messages sent from these IP addresses aren't checked by the content filter. For instructions on how to configure the connection filter policy by adding IP addresses to the IP Allow list or IP Block list, see **Configure the Connection Filter Policy**.

For additional management tasks related to transport rules, see **Transport rules**.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to bypass spam filtering for a domain or user

1. In the EAC, navigate to **Mail flow** > **Rules**. Choose **Add +** and then choose **Bypass spam filtering**.
2. Give the rule a name. Under **Apply this rule if**, choose **The sender** and then select one of the following conditions:
   - If you want to specify a domain, choose **domain is**. In the **Specify domain** dialog box, enter the domain of the sender you want to designate as safe, such as contoso.com. **Add +** to move it to the list of phrases. Repeat this step if you want to add additional domains, and click **OK** when you are finished.
   - If you want to specify a user, choose **is this person**. In the **Select members** dialog box, add the user from the list or type the user and click **Check names**. Repeat this step if you want to add additional users, and click **OK** when you are finished.
3. If you'd like, you can make selections to audit the rule, test the rule, activate the rule during a specific time period, and other selections. We recommend selecting the Stop processing more rules check box to ensure that no other rule can reverse the bypass action. We also recommend testing the rule for a period of time before you enforcing it in your organization. For more

information about these selections, see Manage Transport Rules.

4. Choose **Save** to save the rule.

After you create and enforce the rule, spam filtering is bypassed for the domain or user you specified.

Use the EAC to block messages sent from a domain or user

1. In the EAC, navigate to **Mail flow** > **Rules**. Choose **Add +** and then choose **Create a new rule**.
2. Give the rule a name and then click **More options**.
3. Under **Apply this rule if**, choose **The sender** and then select one of the following conditions:
   - If you want to specify a domain, choose **domain is**. In the Specify domain dialog box, enter the sender domain from which you want to block messages, such as contoso.com. Click **Add +** to move it to the list of phrases. Repeat this step if you want to add additional domains, and click **OK** when you are finished.
   - If you want to specify a user, choose **is this person**. In the **Select members** dialog box, add the user from the list or type the user and click **Check names**. Repeat this step if you want to add additional users, and click **OK** when you are finished.
4. Under **Do the following**, choose **Block the message** and then click one of the other options such as **Delete the message without notifying anyone**.
5. If you'd like, you can make selections to audit the rule, test the rule, activate the rule during a specific time period, and other selections. We recommend testing the rule for a period of time before you enforcing it in your organization. For more information about these selections, see Manage Transport Rules.
6. Choose **Save** to save the rule.

After you create and enforce the rule, any messages sent from the domain or user you specify will be blocked.

Register Filter Pack IFilters with Exchange 2013

Messaging policy and compliance > Transport rules > Transport rule procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-24*

Transport rules with attachment scanning conditions perform text extraction when analyzing the content of attachments. Exchange 2013 can scan most commonly used attachment types natively.

Additional attachment types can be included by registering IFilters with Exchange 2013. This topic shows you how to register IFilters released by Microsoft and third-party providers.

After you register an IFilter for a specific file type, transport rules with attachment processing conditions will be able to scan these attachments. As a result, these file types will no longer trigger the *AttachmentIsUnsupported* condition.

Caution:

The procedures listed in this topic involve modifying the registry on your Exchange servers. Incorrectly editing the registry can cause serious problems that may require you to reinstall your operating system. Problems resulting from editing the registry incorrectly may not be able to be resolved. Before editing the registry, back up any valuable data.
These procedures also require you to stop and restart the Microsoft Exchange Transport service on your Mailbox servers.

For additional management tasks related to Transport rules, see Manage Transport Rules.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes per server.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange server configuration settings" entry in the Exchange and Shell infrastructure permissions topic.
- You must perform the procedures below on servers that already have Exchange 2013 Mailbox server role installed. If you add additional Mailbox servers after you perform these procedures, you must perform them again on the newly provisioned servers.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Register the Microsoft Office 2010 Filter Pack

By default, the following Office file types aren't supported by Exchange transport rules:

- Office OneNote
- Office Publisher

If you want to support these files, you must deploy the Microsoft Office 2010 Filter Pack. This Filter Pack isn't deployed during Exchange 2013 Setup and isn't a prerequisite for deployment.

Deploy the Microsoft Office 2010 Filter Pack

Deploying the Office 2010 Filter Pack consists of two main steps:

- Downloading and installing the Filter Pack, which registers the IFilters with Windows (Search).
- Modifying the registry so the IFilters are also registered with Exchange 2013. This allows Exchange to support attachment scanning for the file formats.

Important:
You must perform this procedure on all Mailbox servers in your organization.

1. Download and save the Microsoft Office 2010 Filter Pack (`FilterPack64bit.exe`) from the Microsoft Download Center.
2. Run the `FilterPack64bit.exe` file on your Mailbox server and follow the instructions to complete the installation.
3. Start Registry Editor and locate the following registry subkey:

`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ExchangeServer\v15\HubTransportRole\CLSID`

4. Under **CLSID**, add a subkey for OneNote files as follows:
   a. Right-click **CLSID**, point to **New**, and then click **Key**.
   b. Change the name of the new key to `{B8D12492-CE0F-40AD-83EA-099A03D493F1}`.
   c. Click the key you just created and set the **(Default)** value to where you installed the Office 2010 Filter Pack. By default, the filter pack gets installed at `C:\Program Files\Common Files\Microsoft Shared\Filters\ONIFilter.dll`.
   d. Right click **{B8D12492-CE0F-40AD-83EA-099A03D493F1}**, point to **New**, and then click **String Value**.
   e. Name the new string value `ThreadingModel` and set it to `Both`.
5. Under **CLSID**, add a subkey for Publisher files as follows:
   a. Right-click **CLSID**, point to **New**, and then click **Key**.
   b. Change the name of the new key to `{A7FD8AC9-7ABF-46FC-B70B-6A5E5EC9859A}`.
   c. Click the key you just created and set the **(Default)** value to where you installed the Office 2010 Filter Pack. By default, the filter pack gets installed at `C:\Program Files\Common Files\Microsoft Shared\Filters\PUBFILT.dll`.
   d. Right-click **{A7FD8AC9-7ABF-46FC-B70B-6A5E5EC9859A}**, point to **New**, and then click **String Value**.
   e. Name the new string value `ThreadingModel` and set it to `Both`.
6. Locate the following registry key:

`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ExchangeServer\v15\HubTransportRole\filters`

7. Under **filters**, add a subkey for .one extensions as follows.
   a. Right-click **filters**, point to **New**, and then click **Key**.
   b. Change the name of the new key to `.one`.
   c. Click the key you just created and set the **(Default)** value to `{B8D12492-CE0F-40AD-83EA-099A03D493F1}`.

8. Under **filters**, add a subkey for .pub extensions as follows:
   a. Right-click **filters**, point to **New** and then click **Key**.
   b. Change the name of the new key to `.pub`.
   c. Click the key you just created and set the **(Default)** value to `{A7FD8AC9-7ABF-46FC-B70B-6A5E5EC9859A}`.
9. Close Registry Editor.
10. On your Mailbox server, stop and then restart the following services in the specified order:
   a. Stop the Microsoft Exchange Transport service.
   b. Stop the Microsoft Filtering Management Service.
   c. Start the Microsoft Filtering Management Service.
   d. Start the Microsoft Exchange Transport service.

How do you know this worked?

To verify that you have successfully registered the Microsoft Office 2010 Filter Pack IFilters, do the following:

1. Create a Transport rule with the following properties. For detailed instructions about how to create Transport rules, see Manage Transport Rules.
   - The sender is your mailbox.
   - Any attachment's content includes "Testing IFilters".
   - Generate an incident report and send it to your mailbox.
2. Create a OneNote file that contains the phrase "Testing IFilters", attach it to a new email message, and send it to yourself.
3. Verify that you receive a Transport rule incident report for the rule you just created. This confirms that the rules engine was able to analyze the contents of the OneNote file.
4. Repeat Steps 2 and 3 with a Publisher file.

Register third-party IFilters to support additional file formats

You can extend the attachment scanning capability for additional file types by registering additional third-party IFilters. Support for additional files can be added by installing and registering the file type's IFilter on each of your Mailbox servers.

Important:
Microsoft hasn't tested third-party IFilters with transport rules, therefore we recommend that you deploy and test any third-party IFilters in a test environment before deploying into your production environment.

Deploy the Adobe PDF IFilter

This procedure shows how to deploy the Adobe PDF IFilter to support processing of PDF attachments in transport rules.

Note:

By default, Exchange 2013 supports the scanning of PDF files in transport rules. The PDF example here is used simply to illustrate how you can extend support for additional file types using third-party IFilters.

1. Download the Adobe PDF IFilterand then follow the installation instructions.
2. Start Registry Editor and locate the following subkey:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ExchangeServer\v15
\HubTransportRole\CLSID
```

3. Under **CLSID**, add a subkey for PDF files as follows:
   a. Right-click **CLSID**, point to **New**, and then click **Key**.
   b. Change the name of the new key to `{E8978DA6-047F-4E3D-9C78-CDBE46041603}`.

Note:

Each IFilter has a unique class ID (CLSID). You can find the CLSID in the installation documentation for the IFilter you're registering or by searching for the file extension under the `HKEY_CLASSES_ROOT\CLSID` key in the registry.

   c. Click the key you just created and set the **(Default)** value to where you installed the PDF IFilter. By default, the PDF IFilter is installed at `C:\Program Files\Adobe\Adobe PDF IFilter 9 for 64-bit platforms\bin\PDFFilter.dll`.
4. Locate the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ExchangeServer\v15
\HubTransportRole\filters
```

5. Under **filters**, add a subkey for .pdf extensions as follows:
   a. Right-click **filters**, point to **New**, and then click **Key**.
   b. Change the name of the new key to `.pdf`.
   c. Click the key you just created and set the **(Default)** value to `{E8978DA6-047F-4E3D-9C78-CDBE46041603}`.
6. Close Registry Editor.
7. On your Mailbox server, stop and restart the following services in the specified order:
   a. Stop the Microsoft Exchange Transport service.
   b. Stop the Microsoft Filtering Management Service.
   c. Start the Microsoft Filtering Management Service.
   d. Start the Microsoft Exchange Transport service.

How do you know this worked?

Use the same procedure listed in the How do you know this worked? section earlier in this topic, substituting Publisher files with Adobe PDF files.

For more information

Using transport rules to inspect message attachments

Transport rules

Transport rule conditions (predicates)

Transport rule actions

Manage Transport Rules

Messaging policy and compliance > Transport rules > Transport rule procedures >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-08-08*

You can use transport rules to look for specific conditions on messages that pass through your organization and take action on them. This topic shows you how to create, copy, adjust the order, enable or disable, delete, or import or export rules, and how to monitor rule usage.

Tip:

To make sure your rules work the way you expect, be sure to thoroughly test each rule and interactions between rules.

Interested in scenarios where these procedures are used? See the following topics:

- Organization-wide disclaimers, signatures, footers, or headers
- Using transport rules to inspect message attachments
- **Use transport rules to aggressively filter bulk email messages**
- Define rules to encrypt or decrypt messages
- Create a domain or user-based safe sender or blocked sender list using transport rules

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport rules" entry in Messaging policy and compliance permissions (Exchange Server), or in **Feature permissions in Exchange Online**.
- If you are using Exchange Online or Exchange Online Protection, you can have up to 100 rules.
- When a rule is listed as **version 14**, this means that the rule is based on an Exchange Server 2010 transport rule format. All options are available for these rules.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a transport rule

You can create a transport rule by setting up a Data Loss Prevention (DLP) policy, creating a new rule, or by copying a rule. You can use the Exchange admin center (EAC) or the Shell.

Use a DLP policy to create transport rules

Each DLP policy is a collection of transport rules. After you create the DLP policy, you can fine-tune the rules using the procedures below.

1. Create a DLP policy. For instructions, see:
   - Exchange Server 2013 DLP Procedures
   - Exchange Online DLP procedures
2. Modify the transport rules created by the DLP policy. See View or modify a transport rule.

Use the EAC to create a transport rule

The EAC allows you to create transport rules by using a template, copying an existing rule, or from scratch.

1. Go to **Mail flow** > **Rules**.
2. Create the rule by using one of the following options:
   - To create a rule from a template, click **Add +** and select a template.
   - To copy a rule, select the rule, and then select **Copy** 🗎.
   - To create a new rule from scratch, **Add +** and then select **Create a new rule**.
3. In the **New rule** dialog box, name the rule, and then select the conditions and actions for this rule:
   - In **Apply this rule if...**, select the condition you want from the list of available conditions.
     - Some conditions require you to specify values. For example, if you select **The sender is...** condition, you must specify a sender address.
     - If the condition you want isn't listed, or if you need to add exceptions, select **More options**. Additional conditions and exceptions will be listed.
     - If you don't want to specify a condition, and want this rule to apply to every message in your organization, select **[Apply to all messages]** condition.
   - In **Do the following...**, select the action you want the rule to take on messages matching the criteria from the list of available actions.
     - Some of the actions will require you to specify values. For example, if you select the **Forward the message for approval to...** condition, you will need to select a recipient in your organization.
     - If the condition you want isn't listed, select **More options**. Additional conditions will be listed.
   - Specify how rule match data for this rule is displayed in the Data Loss Prevention (DLP) reports and the transport rule reports.

- Under **Audit this rule with severity level**, select a level to specify the severity level for this rule. The severity level has no impact on the priority in which the rule is processed: it just impacts how the report is categorized in the reports.

Note:
If you clear the **Audit this rule with severity level** checkbox, rule matches will not show up in the rule reports

- Set the mode for the rule. You can use one of the two test modes to test the rule without impacting mail flow. In both test modes, when the conditions are met, an entry is added to the message trace.
  - **Enforce** This turns on the rule and it starts processing messages immediately. All actions on the rule will be performed.
  - **Test with Policy Tips** This turns on the rule, and any Policy Tip actions (**Notify the sender with a Policy Tip**) will be sent, but no actions related to message delivery will be performed. Data Loss Prevention (DLP) is required in order to use this mode. To learn more, see Policy Tips.
  - **Test without Policy Tips** Only the Generate incident report action will be enforced. No actions related to message delivery are performed.

4. If you are satisfied with the rule, go to step 5. If you want to add more conditions or actions, or if you want to specify exceptions or set additional properties, click **More options**. After you click **More options**, complete the following fields to create your rule:
   - To add more conditions, click **Add condition**. If you have more than one condition, you can remove any one of them by clicking **Remove X** next to it. Note that there are a larger variety of conditions available once you click **More options**.
   - To add more actions, click **Add action**. If you have more than one action, you can remove any one of them by clicking **Remove X** next to it. Note that there are a larger variety of actions available once you click **More options**.
   - To specify exceptions, click **Add exception**, then select exceptions using the **Except if...** dropdown. You can remove any exceptions from the rule by clicking the **Remove X** next to it.
   - If you want this rule to take effect after a certain date, click **Activate this rule on the following date:** and specify a date. Note that the rule will still be enabled prior to that date, but it won't be processed.

Similarly, you can have the rule stop processing at a certain date. To do so, click **Deactivate this rule on the following date:** and specify a date. Note that the rule will remain enabled, but it won't be processed.

- You can choose to avoid applying additional rules once this rule processes a message. To do so, click **Stop processing more rules**. If you select this, and a message is processed by this rule, no subsequent rules are processed for that message.
- You can specify how the message should be handled if the rule processing can't be completed. By default, the rule will be ignored and the message will be processed regularly, but you can choose to resubmit the message for processing. To do so, check the **Defer the message if rule processing doesn't complete** check box.
- If your rule analyzes the sender address, it only examines the message headers by default.

However, you can configure your rule to also examine the SMTP message envelope. To specify what's examined, click one of the following values for **Match sender address in message**:

- **Header** Only the message headers will be examined.
- **Envelope** Only the SMTP message envelope will be examined.
- **Header or envelope** Both the message headers and SMTP message envelope will be examined.

- You can add comments to this rule in the **Comments** box.

5. Click **Save** to complete creating the rule.

Use the Shell to create a transport rule

This example uses the New-TransportRule cmdlet to create a new transport rule that prepends "`External message to Sales DG:`" to messages sent from outside the organization to the Sales Department distribution group.

```
New-TransportRule -Name "Mark messages from the Internet to Sales DG" -FromScope NotInOrganization -SentTo "Sales Department" -PrependSubject "External message to Sales DG:"
```

The rule parameters and action used in the above procedure are for illustration only. Review all the available transport rule conditions and actions to determine which ones meet your requirements.

How do you know this worked?

To verify that you have successfully created a new transport rule, do the following:

- From the EAC, verify that the new transport rule you created is listed in the **Rules** list.
- From the Shell, verify that you created the new transport rule successfully by running the following command (the example below verifies the rule created in the Shell example above):

```
Get-TransportRule "Mark messages from the Internet to Sales DG"
```

View or modify a transport rule

Use the EAC to view or modify a transport rule

1. From the EAC, go to **Mail flow** > **Rules**.
2. When you select a rule in the list, the conditions, actions, exceptions and select properties of that rule are displayed in the details pane. To view all the properties of a specific rule, double click it. This opens the rule editor window, where you can make changes to the rule. For more information about rule properties, see Use the EAC to create a new Transport rule section, earlier in this topic.

Use the Shell to view or modify a transport rule

The following example gives you a list of all rules configured in your organization:

```
Get-TransportRule
```

To view the properties of a specific transport rule, you provide the name of that rule or its GUID. It is usually helpful to send the output to the **Format-List** cmdlet to format the properties. The following example returns all the properties of the transport rule named **Sender is a member of Marketing**:

```
Get-TransportRule "Sender is a member of marketing" |
Format-List
```

To modify the properties of an existing rule, use the Set-TransportRule cmdlet. This cmdlet allows you to change any property, condition, action or exception associated with a rule. The following example adds an exception to the rule "Sender is a member of marketing" so that it won't apply to messages sent by the user Kelly Rollin:

```
Set-TransportRule "Sender is a member of marketing" -
ExceptIfFrom "Kelly Rollin"
```

How do you know this worked?

To verify that you have successfully modified a transport rule, do the following:

- From the rules list in the EAC, click the rule you modified in the **Rules** list and view the details pane.
- From the Shell, verify that you modified the transport rule successfully by running the following command to list the properties you modified along with the name of the rule (the example below verifies the rule modified in the Shell example above):

```
Get-TransportRule "Sender is a member of marketing" |
Format-List Name,ExceptIfFrom
```

Set the priority of a transport rule

The rule at the top of the list is processed first. This rule has a **Priority** of 0.

Use the EAC to set the priority of a rule

1. From the EAC, go to **Mail flow** > **Rules**. This displays the rules in the order in which they are processed.
2. Select a rule, and use the arrows to move the rule up or down the list.

Use the Shell to set the priority of a rule

The following example sets the priority of "Sender is a member of marketing" to 2:

```
Set-TransportRule "Sender is a member of marketing"
```

```
priority "2"
```

How do you know this worked?

To verify that you have successfully modified a transport rule, do the following:

- From the rules list in the EAC, look at the order of the rules.
- From the Shell, verify the priority of the rules (the example below verifies the rule modified in the Shell example above):

```
Get-TransportRule * | Format-List Name,Priority
```

Enable or disable a transport rule

Rules are enabled when you create them. You can disable a transport rule.

Use the EAC to enable or disable a transport rule

1. From the EAC, go to **Mail flow** > **Rules**.
2. To disable a rule, clear the check box next to its name.
3. To enable a disabled rule, select the check box next to its name.

Use the Shell to enable or disable a transport rule

The following example disables the transport rule "Sender is a member of marketing":

```
Disable-TransportRule "Sender is a member of marketing"
```

The following example enables the transport rule "Sender is a member of marketing":

```
Enable-TransportRule "Sender is a member of marketing"
```

How do you know this worked?

To verify that you have successfully enabled or disabled a transport rule, do the following:

- From the EAC, view the list of rules in the **Rules** list and check the status of the check box in the **ON** column.
- From the Shell, run the following command which will return a list of all rules in your organization along with their status:

```
Get-TransportRule | Format-Table Name,State
```

Remove a transport rule

Use the EAC to remove a transport rule

1. From the EAC, go to **Mail flow** > **Rules**.
2. Select the rule you want to remove and then click **Delete** 🗑.

Use the Shell to remove a transport rule

The following example removes the transport rule "Sender is a member of marketing":

```
Remove-TransportRule "Sender is a member of marketing"
```

How do you know this worked?

To verify that you have successfully removed the transport rule, do the following:

- From the EAC, view the rules in the **Rules** list and verify that the rule you removed is no longer shown.
- From the Shell, run the following command and verify that the rule you remove is no longer listed:

```
Get-TransportRule
```

Monitor rule usage

If you're using Exchange Online or Exchange Online Protection, you can check the number of times each rule is matched by using a rules report. In order to be included in the reports, a rule must have the **Audit this rule with severity level** check box selected. You can look at a report online, or download an Excel version of all the mail protection reports.

Note:
While most data is in the report within 24 hours, some data may take as long as 5 days to appear.

Use the Office 365 admin center to generate a rules report

1. In the Office 365 admin center, select **Reports**.
2. In the **Rules** section, select **Top rule matches for mail** or **Rule matches for mail**.

To learn more, see View mail protection reports.

Download an Excel version of the reports

1. On the Reports page in the Office 365 admin center, select **Mail protection reports (Excel)**.
2. If it is your first time using the Excel mail protection reports, a tab opens to the download page.
   a. Select **Download** to download the Microsoft Office 365 Excel Plugin for Exchange Online Reporting.
   b. Open the download.
   c. In the **Mail Protection reports for Office 365 Setup** dialog box, select **Next**, accept the terms of the license agreement, and then select **Next**.
   d. Select the service you are using, and then select **Next**.
   e. Verify the prerequisites, and then select **Next**.
   f. Select **Install**. A shortcut to the reports is placed on your desktop.
3. On your desktop, select **Office 365 Mail Protection Reports**.

4. In the report, select the **Rules** tab.

Import or export a transport rule collection

You must use the Shell to import or export a transport rule collection. For information about how to import a transport rule collection from an XML file, see Import-TransportRuleCollection. For information about how to export a transport rule collection to an XML file, see Export-TransportRuleCollection.

Need more help?

Resources for Exchange Online:

Transport rules

Transport Rule Conditions (Predicates)

Transport rule actions

Resources for Exchange Online Protection:

Transport rules

Transport rule conditions (predicates)

Transport rule actions

Resources for Exchange Server 2013:

Transport rules

Transport rule conditions (predicates)

Transport rule actions

Managing message approval

Exchange Server 2013 > Messaging policy and compliance > Transport rules >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-25*

You can require all messages sent to specific recipients be approved by moderators by Using the moderated transport feature in Microsoft Exchange Server 2013. You can configure any type of recipient as a moderated recipient, and Exchange will ensure that all messages sent to those recipients go through an approval process.

In any type of organization, you may need to restrict access to specific recipients. The most

common scenario is the need to control messages sent to large distribution groups. Depending on your organization's requirements, you may also need to control the messages sent to executive mailboxes or partner contacts. You can use moderated recipients to accomplish these tasks.

Note:

Microsoft Exchange Server 2007 doesn't support moderated recipients. If a message sent to a moderated distribution group is expanded on an Exchange 2007 Hub Transport server, the message will bypass moderation and will be delivered to all members of the distribution group. If you have Exchange 2007 Hub Transport servers in your Exchange organization, you need to designate an Exchange 2013 Mailbox server as the expansion server for moderated distribution groups. This ensures that all messages sent to the distribution group are moderated.

Contents

Moderated transport components

The moderated transport application consists of the following components:

- **Categorizer** The categorizer in the Transport service on a Mailbox server initiates the approval process. When the categorizer detects a moderated recipient while processing a message, it reroutes the message to the arbitration mailbox.
- **Mailbox Transport service** The Mailbox Transport service on a Mailbox server processes the messages that the categorizer marks for moderation. When the Mailbox Transport service encounters such a message, it delivers the original message to the arbitration mailbox and sends approval requests to the moderators. When a moderator responds with a decision, the Mailbox Transport service marks that decision on the message that's stored in the arbitration mailbox. If an approved message is submitted again by the Information Assistant, the Mailbox Transport service removes the approval workflow wrappers so the message that's delivered is identical to the original message submitted by the sender.
- **Information Assistant** The Information Assistant process in the Mailbox Transport service monitors the arbitration mailbox. The Information Assistant resubmits any approved messages to the Transport service on a Mailbox server for delivery to the intended recipients, or it deletes rejected messages. The Information Assistant is also responsible for sending rejection notifications to the sender. In addition, it cleans up the arbitration mailbox by deleting any stale or orphaned messages from the arbitration mailbox. For example, if a moderator simply deletes an approval request instead of making a decision, the corresponding message waiting for approval in the arbitration mailbox needs to be removed by the Information Assistant.
- **Arbitration mailbox** The arbitration mailbox is used to store the original message that's

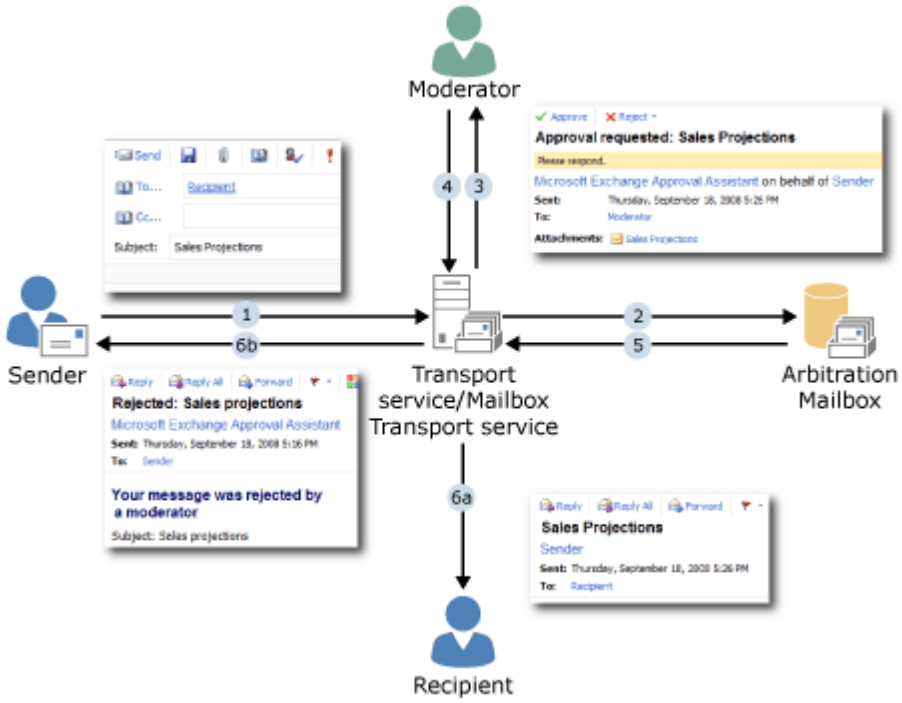
awaiting approval. By default, one arbitration mailbox is created for moderated transport during setup. It's used for all moderated recipients. You can add additional arbitration mailboxes for load balancing purposes. If you're using multiple arbitration mailboxes, you need to specify which mailbox to use for each moderated recipient.

Return to top

Message flow for moderated recipients

When a user sends a message to a moderated recipient, the message follows a path to its destination, as shown in the following figure and described in the following steps.

Moderated transport message flow



1. The sender creates a message and sends it to the moderated recipient.
2. The categorizer in the Transport service intercepts the message, marks it for moderation, and then reroutes it to the Mailbox Transport service on the Mailbox server where the arbitration mailbox resides.
3. The Mailbox Transport service delivers the message to the arbitration mailbox and sends an approval request to the moderator.
4. The moderator uses the buttons in the approval request to either accept or reject the message.
5. The Mailbox Transport service marks the moderator's decision on the original message stored in the arbitration mailbox.
6. The Information Assistant in the Mailbox Transport service reads the approval status on the message stored in the arbitration mailbox, and then processes the message depending on the moderator's decision:
   a. If the moderator has approved the message, the Information Assistant resubmits the message to the Transport service on a Mailbox server, and the message is delivered to the recipient.
   b. If the moderator has rejected the message, the Information Assistant deletes the message

from the arbitration mailbox and notifies the sender that the message was rejected.

Note:

If the moderator doesn't respond to the message within five days, the Information Assistant will delete the message from the arbitration mailbox and notify the sender that their message has expired.

Return to top

Handling multiple moderated recipients

It's possible to send a message to a group of recipients that includes both moderated recipients and recipients that aren't moderated. In this case, a separate approval process occurs for each moderated recipient.

Consider a message that's sent to 12 recipients, one of which is a moderated distribution group. The categorizer bifurcates or forks this message into two copies. One message is delivered immediately to the 11 recipients that aren't moderated, and the second message is submitted to the approval process for the moderated distribution group.

If a message is intended for more than one moderated recipient, a separate copy is created for each moderated recipient and is submitted to the approval process.

A moderated distribution group may contain other moderated recipients. In this case, after the message to the distribution group is approved, a separate approval process occurs for each moderated recipient that's a member of the distribution group. However, you can also enable the automatic approval of the distribution group members after the message to the moderated distribution group is approved. To do this, you use the *BypassNestedModerationEnabled* parameter on the **Set-DistributionGroup** cmdlet.

Return to top

Bypassing moderation

Messages from moderators are delivered to the moderated recipient immediately, bypassing the approval process. By definition, a moderator has the authority to determine what messages are appropriate for a moderated recipient.

Moderation is also bypassed for owners of distribution groups and dynamic distribution groups. The owner of a distribution group can be responsible for managing the distribution group membership, but may not be able to moderate messages sent to it. For example, the account provisioning staff may be the owners of a distribution group called All Employees, but only specific people in human resources may have moderator rights for the same distribution group.

Return to top

Forward a message to a manager for approval

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-15*

When you configure a recipient for moderation, all messages sent to that recipient are subject to approval by the designated moderators. For more information about how Microsoft Exchange Server 2013 handles recipient moderation, see Managing message approval.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the"Moderated Transport" entry in the Mail flow permissions topic.
- Microsoft Exchange Server 2007 doesn't support moderated recipients. If you have Exchange 2007 Hub Transport servers in your organization, you need to specify an Exchange 2013 Mailbox server as the distribution group expansion server for a moderated distribution group.
- These examples show how to configure a distribution group for moderation, but you can follow similar steps to configure any type of recipient for moderation. Note that some recipient types can only be configured for moderation using the Exchange Management Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to configure a recipient for moderation

This example configures the following moderation settings for the distribution group named All Employees:

- Enable moderation for the distribution group.
- Designate David Hamilton and Yossi Ran as moderators.
- Allow the members of the distribution group named HR to bypass moderation.

- Notify internal senders if their message to the distribution group is rejected, but do not send any notifications to external senders.

To accomplish the tasks in this example scenario, perform the following procedure:

1. In the EAC, navigate to **Recipients** > **Groups**.
2. In the result pane, select the **All employees** distribution group and click **Edit** .
3. On the properties page, click **Message approval**, and complete the following:
   a. Select the **Messages sent to this group have to be approved by a moderator** check box.
   b. In the **Group moderators** list, click **Add +**.
   c. In the **Select group moderators** dialog, find and select David Hamilton, click **Add**, find and select Yossi Ran, and click **Add**. When you are finished, click **OK**.
   d. In the **Senders who don't require message approval** list, click **Add +**.
   e. In the **Select senders** dialog, find and select HR from the list and click **Add**. When you are finished, click **OK**.
   f. In **Select moderation notifications**, select **Notify all senders when their messages aren't approved**.
4. Click **Save**.

Use the Shell to configure a recipient for moderation

Run the following command:

```
Set-<RecipientType> <Identity> -ModerationEnabled $true -
ModeratedBy <recipient1,recipient2...> -
ByPassModerationFromSendersOrMembers
<recipient1,recipient2...> -SendModerationNotifications
<Never | Always | Internal>
```

This example configures the following moderation settings for the distribution group named All Employees:

- Enable moderation for the distribution group.
- Designate David Hamilton and Yossi Ran as moderators.
- Allow the members of the distribution group named HR to bypass moderation.
- Notify internal senders if their message to the distribution group is rejected, but do not send any notifications to external senders.

To accomplish the tasks in this example scenario, run the following command:

```
Set-DistributionGroup "All Employees" -ModerationEnabled
$true -ModeratedBy "David Hamilton","Yossi Ran" -
ByPassModerationFromSendersOrMembers HR -
SendModerationNotifications Internal
```

To add or remove users from the list of moderators or recipients who bypass moderation without

affecting other entries, use the following syntax:

```
Set-<RecipientType> <Identity> -ModeratedBy
@{Add="<recipient1>","<recipient2>"...;
Remove="<recipient1>","<recipient2>"...} -
ByPassModerationFromSendersOrMembers
@{Add="<recipient1>","<recipient2>"...;
Remove="<recipient1>","<recipient2>"...}
```

This example configures the following moderation settings for the distribution group named All Employees:

- Add the user chris@contoso.com to the list of existing moderators.
- Remove the user michelle@contoso.com from the list of existing senders who bypass moderation.

```
Set-DistributionGroup "All Employees" -ModeratedBy
@{Add="chris@contoso.com"} -
ByPassModerationFromSendersOrMembers
@{Remove="michelle@contoso.com"}
```

How do you know this worked?

To verify that you have successfully configured a recipient for moderation, do the following:

1. Send a test message to the moderated recipient.
2. Verify the designated moderators receive notification.
3. Verify the recipients who bypass moderation receive the message directly.

Manage and troubleshoot message approval

Messaging policy and compliance > Transport rules > Managing message approval >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-20*

You may receive the following error when you attempt to remove an arbitration mailbox:

Can't remove the arbitration mailbox <*mailbox*> because it's being used for the approval workflow for existing recipients that have either membership restrictions or moderation

enabled. You should either disable the approval features on those recipients or specify a different arbitration mailbox for those recipients before removing this arbitration mailbox.

An arbitration mailbox can be used to handle the approval workflow for moderated recipients and distribution group membership approvals. You use PowerShell to find all the recipients that are configured to use the arbitration mailbox. After you identify the recipients, you can either configure them to use a different arbitration mailbox, or you can disable moderation for them.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Aribtration" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Step 1: Use the Shell to find all the recipients that use the arbitration mailbox you are trying to delete

Run the following commands:

```
$AM = Get-Mailbox "<arbitration mailbox>" -Arbitration
$AMDN = $AM.DistinguishedName
Get-Recipient -RecipientPreviewFilter {ArbitrationMailbox -
eq $AMDN}
```

For example, to find all the recipients that use the arbitration mailbox named Arbitration Mailbox01, run the following commands:

```
$AM = Get-Mailbox "Arbitration Mailbox01" -Arbitration
$AMDN = $AM.DistinguishedName
Get-Recipient -RecipientPreviewFilter {ArbitrationMailbox -
eq $AMDN}
```

Note:

The arbitration mailbox is specified using the distinguished name (DN). If you know the DN of the arbitration mailbox, you can run the single command: `Get-Recipient -RecipientPreviewFilter {ArbitrationMailbox -eq <DN>}`.

Step 2: Use the Shell to specify a different arbitration

mailbox or disable moderation for the recipients

To stop moderated recipients from using the arbitration mailbox you are trying to delete, you can either specify a different arbitration mailbox, or you can disable moderation for the recipients.

If you choose to specify a different arbitration mailbox for the recipients, run the following command:

```
Set-<RecipientType> <Identity> -ArbitrationMailbox
<different arbitration mailbox>
```

For example, to reconfigure the distribution group named All Employees to use the arbitration mailbox named Arbitration Mailbox02 for membership approval, run the following command:

```
Set-DistributionGroup "All Employees" -ArbitrationMailbox
"Arbitration Mailbox02"
```

If you choose to disable moderation for the recipients, run the following command:

```
Set-<RecipientType> <Identity> -ModerationEanbled $false
```

For example, to disable moderation for the mailbox named Human Resources, run the following command:

```
Set-Mailbox "Human Resources" -ModerationEanbled $false
```

How do you know this worked?

The procedure was successful if you can delete the arbitration mailbox without receiving the error that it's being used.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Data loss prevention

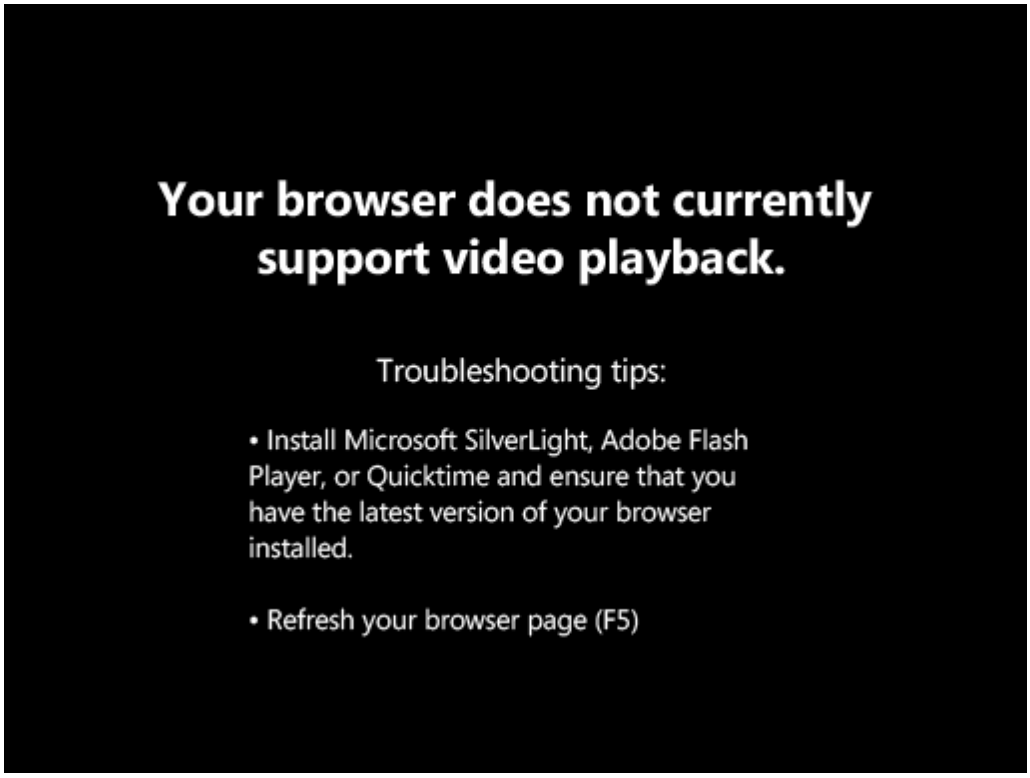
Exchange Server 2013 > Messaging policy and compliance >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-08*

Data loss prevention (DLP) is an important issue for enterprise message systems because of the

extensive use of email for business critical communication that includes sensitive data. In order to enforce compliance requirements for such data, and manage its use in email, without hindering the productivity of workers, DLP features make managing sensitive data easier than ever before. For a conceptual overview of DLP, watch the following video.



DLP policies are simple packages that contain sets of conditions, which are made up of transport rules, actions, and exceptions that you create in the Exchange Administration Center (EAC) and then activate to filter email messages and attachments. You can create a DLP policy, but choose to not activate it. This allows you to test your policies without affecting mail flow. DLP policies can use the full power of existing transport rules. In fact, a number of new types of transport rules have been created in Microsoft Exchange Server 2013 and Exchange Online in order to accomplish new DLP capability. One important new feature of transport rules is a new approach to classifying sensitive information that can be incorporated into mail flow processing. This new DLP feature performs deep content analysis through keyword matches, dictionary matches, regular expression evaluation, and other content examination to detect content that violates organizational DLP policies. The latest release of Exchange Online and Exchange 2013 SP1 add Document Fingerprinting, which helps you detect sensitive information in standard forms. For more information about transport rules, see Transport rules and Integrating sensitive information rules with transport rules. You can also manage your DLP policies by using Exchange Management Shell cmdlets. For more information about policy and compliance cmdlets, see Policy and compliance cmdlets.

In addition to the customizable DLP policies themselves, you can also inform email senders that they may be about to violate one of your policies—even before they send an offending message. You can accomplish this by configuring Policy Tips. Policy Tips are similar to MailTips, and can be configured to present a brief note in the Microsoft Outlook 2013 client that provides information about possible policy violations to a person creating a message. In the latest release of Exchange Online and in Exchange 2013 SP1, Policy Tips are also displayed in Outlook Web App and OWA for

Devices. For more information, see Policy Tips.

Note:

Exchange Online: DLP is a premium feature that requires an Exchange Online Plan 2 subscription. For more information, see Exchange Online Licensing.
Exchange 2013: DLP is a premium feature that requires an Exchange Enterprise Client Access License (CAL). For more information about CALs and server licensing, see Exchange Server Licensing.
Exchange Enterprise CAL with Services: There is a behavior distinction to take note of if you are an Exchange Enterprise CAL with Services customer with a hybrid deployment, where you have some mailboxes located on premises and some in Exchange Online. DLP policies are applied in Exchange Online. Therefore, messages sent from one on-premises user to another on-premises user do not have DLP policies applied, because the message doesn't leave the on-premises infrastructure.

Looking for management tasks related to Data Loss Prevention? See DLP procedures.

Contents

Establish policies to protect sensitive data

The data loss prevention features can help you identify and monitor many categories of sensitive information that you have defined within the conditions of your policies, such as private identification numbers or credit card numbers. You have the option of defining your own custom policies and transport rules or using the pre-defined DLP policy templates provided by Microsoft in order to get started quickly. For more information about the policy templates that are included, see DLP policy templates supplied in Exchange. A *policy template* includes a range of conditions, rules, and actions that you can choose from in order to create and save an actual DLP policy that will help you inspect messages. The policy templates are models from which you can select or build your own specific rules to create a policy that meets your needs for data loss prevention.

Three different methods exist for you to begin using DLP:

1. **Apply an out-of-the-box template supplied by Microsoft.** The quickest way to start using DLP policies is to create and implement a new policy using a template. This saves you the effort of building a new set of rules from nothing. You will need to know what type of data you want to

check for or which compliance regulation you are attempting to address. You will also need to know your organizations expectations for processing such data. More information at DLP policy templates supplied in Exchange and Create a DLP policy from a template.

2. **Import a pre-built policy file from outside your organization.** You can import policies that have already been created outside of your messaging environment by independent software vendors. In this way you can extend the DLP solutions to suit your business requirements. More information at Policy templates from Microsoft partners, Define your own DLP templates and information types, and Import a DLP policy from a file.
3. **Create a custom policy without any pre-existing conditions.** Your enterprise may have its own requirements for monitoring certain types of data known to exist within a messaging system. You can create a custom policy entirely on your own in order to start checking and acting upon your own unique message data. You will need to know the requirements and constraints of the environment in which the DLP policy will be enforced in order to create such a custom policy. More information at Create a custom DLP policy.

After you have added a policy, you can review and change its rules, make the policy inactive, or remove it completely. The procedures for these actions are provided in the Manage DLP policies topic.

Sensitive information types in DLP policies

When you create or change DLP policies, you can include rules that include checks for sensitive information. The sensitive information types listed in the Sensitive information types inventory topic are available to be used in your policies. The conditions that you establish within a policy, such as how many times something has to be found before an action is taken or exactly what that action is can be customized within your new custom policies in order to meet your specific policy requirements. For more information about creating DLP policies see, Create a custom DLP policy. For more information about the full suite transport rules, see Transport rules.

To make it easy for you to make use of the sensitive information-related rules, Microsoft has supplied policy templates that already include some of the sensitive information types. You cannot add conditions for all of the sensitive information types listed here to policy templates however, because the templates are designed to help you focus on the most-common types of compliance-related data within your organization. For more information about the pre-built templates, see DLP policy templates supplied in Exchange. You can create numerous DLP policies for your organization and have them all enabled so that many disparate types of information are examined. You can also create a DLP policy that is not based on an existing template. To begin creating such a policy, see Create a custom DLP policy. For more information about sensitive information types, see Sensitive information types inventory.

Detecting sensitive form data with Document

Fingerprinting

With Exchange 2013 SP1 and the latest version of Exchange Online, you can use Document Fingerprinting to easily create a sensitive information type based on a standard form. To learn how to protect form data, see Protect form data with document fingerprinting.

Policy Tips notify users about sensitive content expectations

You can use Policy Tip notification messages to inform email senders about possible compliance issues while they are composing an email message. When you configure a Policy Tip in a DLP policy, the notification message will only show up if something in the sender's email message meets the conditions described in your policy. Policy Tips are similar to MailTips that were introduced in Microsoft Exchange 2010. For more information, see Policy Tips.

Detecting sensitive information along with traditional message classification

Exchange 2013 and Exchange Online present a new method of helping you manage message and attachment data when compared with traditional message classification. A key factor in the strength of a DLP solution is the ability to correctly identify confidential or sensitive content that may be unique to the organization, regulatory needs, geography, or other business needs. Exchange 2013 can achieve this by using a new architecture for deep content analysis coupled with detection criteria that you establish through rules in your DLP policies. Helping prevent data loss in Exchange 2013 relies on configuring the correct set of sensitive information rules so that they provide a high degree of protection while minimizing inappropriate mail flow disruption with false positives and negatives. These types of rules, referred to throughout the DLP information as sensitive information detection, function within the framework offered by transport rules in order to enable DLP capabilities. To learn more about these new features, see Integrating sensitive information rules with transport rules. The traditional message classification fields can still be applied to messages in Exchange and these can be combined withthe new sensitive information detection either together within a single DLP policy or running concurrently so they are evaluated independently within Exchange. To learn more about the legacy Exchange 2010 message classifications, see Understanding Message Classifications in the TechNet Library.

Information about DLP-processed messages

To obtain information about messages and DLP policy detections in your environment, see DLP

policy detection reports. Data related to DLP detections, is highly integrated into the delivery reports message tracking tool of Exchange 2013.

Installation prerequisites

In order to make use of DLP features, you must have Exchange 2013 or Exchange Online configured with at least one sender mailbox. Data Loss Prevention is a premium feature that requires an Enterprise Client Access License (CAL). For more information about getting started with Exchange 2013, see Planning and deployment. For more information about getting started with Exchange Online, see **Exchange Online**.

For more information

Messaging policy and compliance

DLP procedures

Document Fingerprinting

Policy and compliance cmdlets

DLP policy templates

Exchange Server 2013 > Messaging policy and compliance > Data loss prevention >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-28*

You can use data loss prevention (DLP) policy templates to get started with your DLP solution in Microsoft Exchange Server 2013. A DLP policy template is a model for a policy. You can select a template to begin the process of building your own customized DLP policy. Within your DLP policy, you can customize the rules to ensure that it meets your business needs for data loss prevention. Several policy templates are supplied by Microsoft, but these are not the only way to implement a data loss prevention solution in Exchange.

Looking for management tasks related to DLP policy templates? See DLP procedures.

Contents

For more information

Extend the templates and information types to meet your needs

You can incorporate sensitive-content definitions and policy templates from Microsoft Partners or from files that you develop yourself as an addition to the DLP policy templates, information types, and rules already provided in Exchange 2013. Presented here are several ways in which you can add your own unique DLP content and extend DLP functionality. The templates already provided by Microsoft are a convenient method to get started with a DLP solution. In order to extend the DLP features with your own unique DLP policy template files, you must understand the XML schema requirements for policy templates that are created independent of Exchange. To learn more about the Exchange Management Shell cmdlets associated with DLP policy templates, see cmdlets related to `Get-DlpPolicyTemplate` in Policy and compliance cmdlets. Furthermore, you can define your own sensitive content types after you understand the format and procedure to incorporate them. To learn more about the Exchange Management Shell cmdlets associated with DLP policy templates, see cmdlets related to `Get-ClassificationRuleCollection` in Policy and compliance cmdlets.

Caution:
You should turn on your DLP policies in test mode before enforcing them in your production environment. During such tests, we recommended that you configure sample user mailboxes and send test messages that invoke your test policies in order to confirm the results.

Create your own new DLP policy template or your own sensitive information types in a classification rule package

You can create a DLP policy template file apart from Exchange that meets the specific XML schema definition provided by Microsoft and then import the file into your system so that you can create DLP policies from it. By creating your own template files, you can define your own model for DLP policies that Microsoft has not already provided. This is different than creating a DLP policy by using the Exchange Administration Center, which typically happens after policy templates are available. If you create a policy template independent of Exchange, you will need to import it before you can use it to scan messages. You can also create your own sensitive information definitions apart from those defined by Microsoft in Exchange. There is a separate XML schema definition for DLP policy template files and classification rule packages. To get started with this, see the following information:

Define your own DLP templates and information types

Import a DLP policy from a file

Include DLP functionality with existing transport rules

You can incorporate DLP detection capabilities with traditional transport rules without creating a new DLP policy. If you have created a complex set of rules in a previous version of Exchange, and you want to duplicate them or add sensitive information detection in Exchange 2013, then you can use the transport rules editor in the Exchange Administration Center or the Exchange management shell to incorporate these two features. To get started with this, see the following information:

Transport rules

Manage Transport Rules
Policy and compliance cmdlets

Use DLP policies created by Microsoft

Numerous DLP policies are supplied by Microsoft. This is the easiest way to get started with a DLP solution that is flexible and simple to implement. You can always use the provided policies as a starting point and customize them further to meet your requirements. To get started with this, see the following information:

- DLP policy templates supplied in Exchange
- Create a DLP policy from a template

For more information

Data loss prevention

How DLP rules are applied to evaluate messages

Messaging policy and compliance > Data loss prevention > DLP policy templates >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-11-25*

You can set up sensitive information rules within your Microsoft Exchange data loss prevention (DLP) policies to detect very specific data in email messages. This topic will help you understand how these rules are applied and how messages are evaluated. You can avoid workflow disruptions for your email users and achieve a high degree of accuracy with your DLP detections if you know how your rules are enforced. Let's use the Microsoft-supplied credit card information rule as an

example. When you activate a transport rule or DLP policy, the Exchange transport rules agent compares all messages that your users send with the rule sets that you create.

Get precise about your needs

Suppose you need to act on credit card information in messages. The actions you take once it is found are not the subject of this topic, but you can learn more about that in **Transport rule actions**. With as most certainty as possible, you need to ensure that what is detected in a message is truly credit card data and not something else that could be a legitimate use of groups of numbers that merely resemble credit card data; for example, a reservation code or a vehicle identification number. To meet this need, let's make it clear that the following information should be classified as a credit card.

Margie's Travel,
I have received updated credit card information for Spencer.
Spencer Badillo
Visa: 4111 1111 1111 1111
Expires: 2/2012
Please update his travel profile.

Let's also make it clear that the following information should not be classified as a credit card.

Hi Alex,
I expect to be in Hawaii too. My booking code is 1234 1234 1234 1234 and I'll be there on 3/2012.
Regards, Lisa

The following XML snippet shows how the needs expressed earlier are currently defined in a sensitive information rule that is provided with Exchange and it is embedded within one of the supplied DLP policy templates.

```
<Entity id="50842eb7-edc8-4019-85dd-5a5c1f2bb085"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_credit_card" />
        <Any minMatches="1">
          <Match idRef="Keyword_cc_verification" />
          <Match idRef="Keyword_cc_name" />
          <Match idRef="Func_expiration_date" />
        </Any>
      </Pattern>
    </Entity>
```

Pattern-matching in your solution

The XML rule definition shown earlier includes pattern-matching, which improves the likelihood that the rule will detect only the important information and not detect vague, related information. For more information about the XML schema for DLP rules and templates, see Define your own DLP templates and information types.

In the credit card rule, there is a section of XML code for patterns, which includes a primary identifier match and some additional corroborative evidence. All three of these requirements are explained here:

1. `<IdMatch idRef="Func_credit_card" />` — This requires a match of a function, titled credit card, that is internally defined. This function includes a couple of validations as follows:
   a. It matches a regular expression—in this instance for 16 digits—that could also include variations like a space delimiter so that it also matches **4111 1111 1111 1111** or a hyphen delimiter so that it also matches **4111-1111-1111-1111**.
   b. It evaluates the Lhun's checksum algorithm against the 16-digit number in order to ensure the likelihood of this being a credit card number is high.
   c. It requires a mandatory match, after which corroborative evidence is evaluated.
2. `<Any minMatches="1">` — This section indicates that the presence of at least one of the following items of evidence is required.

3. The corroborative evidence can be a match of one of these three:

```
<Match idRef="Keyword_cc_verification" />
<Match idRef="Keyword_cc_name" />
<Match idRef="Func_expiration_date" />
```

These three simply mean a list of keywords for credit cards, the names of the credit cards, or an expiration date is required. The expiration date is defined and evaluated internally as another function.

The process of evaluating content against rules

The five steps here represent actions that Exchange takes to compare your rule with email messages. For our credit card rule example, the following steps are taken.

Step	Action
1. Get Content	Spencer Badillo Visa: 4111 1111 1111 1111 Expires: 2/2012
2. Regular Expression Analysis	4111 1111 1111 1111 -> a 16-digit number is

	detected
3. Function Analysis	• 4111 1111 1111 1111 -> matches checksum • 1234 1234 1234 1234 -> doesn't match
4. Additional Evidence	1. Keyword Visa is near the number. 2. A regular expression for a date (2/2012) is near the number.
5. Verdict	1. There is a regular expression that matches a checksum. 2. Additional evidence increases confidence.

The way this rule is set up by Microsoft makes it mandatory that corroborating evidence such as keywords are a part of the email message content in order to match the rule. So the following email content would not be detected as containing a credit card:

Margie's Travel,
I have received updated information for Spencer.
Spencer Badillo
4111 1111 1111 1111
Please update his travel profile.

You can use a custom rule that defines a pattern without extra evidence, as shown in the next example. This would detect messages with only credit card number and no corroborating evidence.

```
      <Pattern confidenceLevel="85">
         <IdMatch idRef="Func_credit_card" />
      </Pattern>
   </Entity>
```

The illustration of credit cards in this article can be extended to other sensitive information rules as well. To see the complete list of the Microsoft-supplied rules in Exchange, use the Get-ClassificationRuleCollection cmdlet in the Exchange Management Shell in the following manner:

```
$rule_collection = Get-ClassificationRuleCollection

$rule_collection[0].SerializedClassificationRuleCollection
| Set-Content oob_classifications.xml -Encoding byte
```

For more information

Data loss prevention

Transport rules

Exchange Management Shell

Exchange Online PowerShell

Sensitive information types inventory

Messaging policy and compliance > Data loss prevention > DLP policy templates >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-11*

Data loss prevention (DLP) includes 51 sensitive information type rules that are ready to use right out of the box. You can use any sensitive information type in a DLP policy, and in that policy you can use other conditions, actions, and exceptions to solve your DLP requirements. The information types provided by Microsoft and listed here can help you detect personally identifiable information (PII), financial information, and health information across a number of regions.

This topic has the following reference information to help you understand the included sensitive information types, their properties, and where they are used in the XML.

- Sensitive information type definitions
- Regex definitions contained in DLP rule pack
- Keyword list used for DLP rules
- XML rule definitions

If you would like to learn how to adjust the sensitivity of the included rules, follow the example in **Customize the built-in DLP rules to detect more credit card data**. You can also extend and customize your DLP solution with information types that you provide yourself. To learn more about creating your own specialized information types, see Define your own DLP templates and information types. For more information about modifying your DLP policies by using sensitive information detection, see Integrating sensitive information rules with transport rules and Manage DLP policies.

Sensitive information type definitions

The following table lists the properties of the included sensitive information types, such as the name, the term used in the XML, the primary geographical region that uses it, and the type of information that it detects. You can use the XML terms to find the corresponding sensitive information in the code snippets in the other sections of this topic.

Information type name	XML term	Primary region	Category
ABA Routing Number	`aba_routing`	United States	finance

Australia Bank Account Number	`australia_bank_account_number`	Australia	finance
Australia Driver's License Number	`australia_drivers_license_number`	Australia	PII
Australia Medical Account Number	`australian_medical_account_number`	Australia	health
Australia Passport Number	`australia_passport_number`	Australia	PII
Australia Tax File Number	`australian_tax_file_number`	Australia	finance
Canada Bank Account Number	`canada_bank_account_number`	Canada	finance
Canada Driver's License Number	`canada_drivers_license`	Canada	PII
Canada Health Service Number	`canada_health_service_number`	Canada	health
Canada Passport Number	`canada_passport_number`	Canada	PII
Canada Personal Health Identification Number (PHIN)	`canada_phin`	Canada	health
Canada Social Insurance Number	`canadian_sin`	Canada	PII
Credit Card Number	`credit_card`	All	finance
Drug Enforcement Agency (DEA) Number	`dea_number`	United States	PII

EU Debit Card Number	`eu_debit_card`	European Union	finance
Finland National ID[1]	`finnish_national_id`	Finland	PII
France Driver's License Number	`french_drivers_license`	France	PII
France National ID Card (CNI)	`france_cni`	France	PII
France Passport Number	`fr_passport`	France	PII
France Social Security Number (INSEE)	`french_insee`	France	PII
German Driver's License Number	`german_drivers_license`	Germany	PII
German Passport Number	`german_passport`	Germany	PII
International Banking Account Number (IBAN)	`iban`	All	finance
IP Address	`ipaddress`	All	PII
Israel Bank Account Number	`israel_bank_account_number`	Israel	finance
Israel National ID	`israeli_national_id_number`	Israel	PII
Italy Driver's License Number	`italy_drivers_license_number`	Italy	PII
Japan Bank Account Number	`jp_bank_account`	Japan	finance

Japan Driver's License Number	`jp_drivers_license_number`	Japan	PII
Japan Passport Number	`jp_passport`	Japan	PII
Japan Resident Registration Number	`jp_resident_registration_number`	Japan	PII
Japan Social Insurance Number (SIN)	`jp_sin`	Japan	PII
New Zealand Ministry of Health Number	`new_zealand_ministry_of_health_number`	New Zealand	health
Saudi Arabia National ID	`saudi_arabia_national_id`	Saudi Arabia	PII
Poland National ID (PESEL)[1]	`pesel_identification_number`	Poland	PII
Poland Identity Card[1]	`polish_national_id`	Poland	PII
Poland Passport[1]	`polish_passport_number`	Poland	PII
Spain National ID		Spain	PII
Spain Social Security Number (SSN)	`spanish_social_security_number`	Spain	PII
Sweden National ID	`swedish_national_identifier`	Sweden	PII
Sweden Passport Number	`sweden_passport_number`	Sweden	PII
SWIFT Code	`swift`	All	finance
Taiwan National ID[1]	`taiwanese_national_id`	Taiwan	PII

U.K. Driver's License Number	uk_drivers_license	United Kingdom	PII
U.K. Electoral Roll Number	uk_electoral	United Kingdom	PII
U.K. National Health Service Number	uk_nhs_number	United Kingdom	health
U.K. National Insurance Number (NINO)	uk_nino	United Kingdom	health
U.S. / U.K. Passport Number	usa_uk_passport	United States and United Kingdom	PII
U.S. Bank Account Number	usa_bank_account_number	United States	finance
U.S. Driver's License Number	us_drivers_license	United States	PII
U.S. Individual Taxpayer Identification Number (ITIN)	itin	United States	finance
U.S. Social Security Number (SSN)	ssn	United States	health

Note:

[1]Available in Exchange 2013 SP1 and Exchange Online.

Regex definitions contained in DLP rule pack

The following code snippet contains the regular expressions, or *regex*, that are used within DLP to detect common character patterns in sensitive information.

```
    <Regex id="Regex_france_cni">(^|\s)(\d{12})($|\s|\.\s)
</Regex>
    <Regex id="Regex_uk_electoral">(^|\s)([a-zA-Z]{2}
```

```
\d{1,4})($|\s|\.\s)</Regex>
    <Regex id="Regex_canada_health_service_number">(^|\s)
(\d{10})($|\s|\.\s)</Regex>
    <Regex id="Regex_canada_phin">(^|\s)(\d{9})($|\s|\.\s)
</Regex>
    <Regex id="Regex_canada_passport_number">(^|\s)(\D{2})
(\d{6})($|\s|\.\s)</Regex>
    <Regex id="Regex_canada_bank_account_number">(^|\s)
(\d{7})($|\s|\.\s)</Regex>
    <Regex id="Regex_australia_passport_number">(^|\s)([A-
Za-z]\d{7})($|\s|\.\s)</Regex>
    <Regex id="Regex_australia_drivers_license_number">(?
ix)(?:^|\s)(?=(?:[A-Z\d]{2}\d{2}[A-Z\d]{5})(?:$|\s|.\s))(?
=(?:[A-Z]{0,2}\d){4,9})(?!(?:\d{0,9}[A-Z]){3,9})[A-Z\d]{9}
</Regex>
    <Regex id="Regex_australia_bank_account_number">(^|\s)
([0-9]{6,10})($|\s|\.\s)</Regex>
    <Regex id="Regex_sweden_passport_number">(^|\s)(\d{8})
($|\s|\.\s)</Regex>
    <Regex id="Regex_italy_drivers_license_number">(^|\s)
(\D{1}[^b-uw-zB-UW-Z])((\w{7})(\D))($|\s|\.\s)</Regex>
    <Regex id="Regex_ipv4_address">(^|\s)((?:[0-9]\.)|(?:
[0-9]))(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?
:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)(?!(?:\.[0-9])|(?:[0-
9]))($|\s|\.\s)</Regex>
    <Regex id="Regex_ipv6_address">(^|\s)((?:[A-F0-9]
{1,4}:){7}[A-F0-9]{1,4})($|\s|\.\s)</Regex>
    <Regex id="Regex_israel_bank_account_number">(^|\s)
(\d{2}-\d{3}-\d{8}|\d{13})($|\s|\.\s)</Regex>
    <Regex id="Regex_saudi_arabia_national_id">(^|\s)
(\d{10})($|\s|\.\s)</Regex>
    <Regex id="Regex_usa_bank_account_number">(^|\s)
(\d{4,17})($|\s|\.\s)</Regex>
```

Keyword list used for DLP rules

The following code snippet shows the keywords that are used for each sensitive information type rule. Keyword information is valuable to help you determine why a rule might not be detecting information that you expect to be detected. However, keep in mind that the service evaluates more

than just keywords when detecting information. Confidence and proximity are also important parts of the evaluation process. For more information about how DLP rules are applied, see How DLP rules are applied to evaluate messages.

```
<Keyword id="Keyword_nz_terms">
      <Group matchStyle="word">
        <Term caseSensitive="false">NHI</Term>
        <Term caseSensitive="false">New Zealand</Term>
        <Term caseSensitive="false">Health</Term>
        <Term caseSensitive="false">treatment</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_canada_provinces">
      <Group matchStyle="word">
        <Term caseSensitive="false">Nunavut</Term>
        <Term caseSensitive="false">Quebec</Term>
        <Term caseSensitive="false">Northwest Territories</
Term>
        <Term caseSensitive="false">Ontario</Term>
        <Term caseSensitive="false">British Columbia</Term>
        <Term caseSensitive="false">Alberta</Term>
        <Term caseSensitive="false">Saskatchewan</Term>
        <Term caseSensitive="false">Manitoba</Term>
        <Term caseSensitive="false">Yukon</Term>
        <Term caseSensitive="false">Newfoundland and
Labrador</Term>
        <Term caseSensitive="false">New Brunswick</Term>
        <Term caseSensitive="false">Nova Scotia</Term>
        <Term caseSensitive="false">Prince Edward Island</
Term>
        <Term caseSensitive="false">Canada</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_number_exclusions">
      <Group matchStyle="word">
        <Term caseSensitive="false">00000000</Term>
        <Term caseSensitive="false">11111111</Term>
        <Term caseSensitive="false">22222222</Term>
        <Term caseSensitive="false">33333333</Term>
```

```
        <Term caseSensitive="false">44444444</Term>
        <Term caseSensitive="false">55555555</Term>
        <Term caseSensitive="false">66666666</Term>
        <Term caseSensitive="false">77777777</Term>
        <Term caseSensitive="false">88888888</Term>
        <Term caseSensitive="false">99999999</Term>
        <Term caseSensitive="false">000000000</Term>
        <Term caseSensitive="false">111111111</Term>
        <Term caseSensitive="false">222222222</Term>
        <Term caseSensitive="false">333333333</Term>
        <Term caseSensitive="false">444444444</Term>
        <Term caseSensitive="false">555555555</Term>
        <Term caseSensitive="false">666666666</Term>
        <Term caseSensitive="false">777777777</Term>
        <Term caseSensitive="false">888888888</Term>
        <Term caseSensitive="false">999999999</Term>
        <Term caseSensitive="false">0000000000</Term>
        <Term caseSensitive="false">1111111111</Term>
        <Term caseSensitive="false">2222222222</Term>
        <Term caseSensitive="false">3333333333</Term>
        <Term caseSensitive="false">4444444444</Term>
        <Term caseSensitive="false">5555555555</Term>
        <Term caseSensitive="false">6666666666</Term>
        <Term caseSensitive="false">7777777777</Term>
        <Term caseSensitive="false">8888888888</Term>
        <Term caseSensitive="false">9999999999</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_uk_nino">
      <Group matchStyle="word">
        <Term caseSensitive="false">national insurance
number</Term>
        <Term caseSensitive="false">national insurance
contributions</Term>
        <Term caseSensitive="false">protection act</Term>
        <Term caseSensitive="false">insurance</Term>
        <Term caseSensitive="false">social security
number</Term>
        <Term caseSensitive="false">insurance application</
```

```
Term>
        <Term caseSensitive="false">medical application</
Term>
        <Term caseSensitive="false">social insurance</Term>
        <Term caseSensitive="false">medical attention</
Term>
        <Term caseSensitive="false">social security</Term>
        <Term caseSensitive="false">great britain</Term>
        <Term caseSensitive="false">insurance</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_uk_drivers_license">
      <Group matchStyle="word">
        <Term caseSensitive="false">DVLA</Term>
        <Term caseSensitive="false">light vans</Term>
        <Term caseSensitive="false">quadbikes</Term>
        <Term caseSensitive="false">motor cars</Term>
        <Term caseSensitive="false">125cc</Term>
        <Term caseSensitive="false">sidecar</Term>
        <Term caseSensitive="false">tricycles</Term>
        <Term caseSensitive="false">motorcycles</Term>
        <Term caseSensitive="false">photocard licence</
Term>
        <Term caseSensitive="false">learner drivers</Term>
        <Term caseSensitive="false">licence holder</Term>
        <Term caseSensitive="false">licence holders</Term>
        <Term caseSensitive="false">driving licences</Term>
        <Term caseSensitive="false">driving licence</Term>
        <Term caseSensitive="false">dual control car</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_Australia_Medical_Account_Number">
      <Group matchStyle="word">
        <Term caseSensitive="false">bank account details</
Term>
        <Term caseSensitive="false">medicare payments</
Term>
        <Term caseSensitive="false">mortgage account</Term>
        <Term caseSensitive="false">bank payments</Term>
```

```
        <Term caseSensitive="false">information branch</
Term>
        <Term caseSensitive="false">credit card loan</Term>
        <Term caseSensitive="false">department of human
services</Term>
        <Term caseSensitive="false">local service</Term>
        <Term caseSensitive="false">medicare</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_Australia_Tax_File_Number">
      <Group matchStyle="word">
        <Term caseSensitive="false">australian business
number</Term>
        <Term caseSensitive="false">marginal tax rate</
Term>
        <Term caseSensitive="false">medicare levy</Term>
        <Term caseSensitive="false">portfolio number</Term>
        <Term caseSensitive="false">service veterans</Term>
        <Term caseSensitive="false">withholding tax</Term>
        <Term caseSensitive="false">individual tax return</
Term>
        <Term caseSensitive="false">tax file number</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_Israel_National_ID">
      <Group matchStyle="string">
        <Term caseSensitive="false">□□□□□□□□</Term>
        <Term caseSensitive="false">National ID Number</
Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_uk_electoral">
      <Group matchStyle="word">
        <Term caseSensitive="false">council nomination</
Term>
        <Term caseSensitive="false">nomination form</Term>
        <Term caseSensitive="false">electoral register</
Term>
        <Term caseSensitive="false">electoral roll</Term>
```

```
      </Group>
    </Keyword>
    <Keyword id="Keyword_canada_health_service_number">
      <Group matchStyle="word">
        <Term caseSensitive="false">personal health
number</Term>
        <Term caseSensitive="false">patient information</
Term>
        <Term caseSensitive="false">health services</Term>
        <Term caseSensitive="false">speciality services</
Term>
        <Term caseSensitive="false">automobile accident</
Term>
        <Term caseSensitive="false">patient hospital</Term>
        <Term caseSensitive="false">psychiatrist</Term>
        <Term caseSensitive="false">workers compensation</
Term>
        <Term caseSensitive="false">disability</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_canada_phin">
      <Group matchStyle="word">
        <Term caseSensitive="false">social insurance
number</Term>
        <Term caseSensitive="false">health information
act</Term>
        <Term caseSensitive="false">income tax
information</Term>
        <Term caseSensitive="false">manitoba health</Term>
        <Term caseSensitive="false">health registration</
Term>
        <Term caseSensitive="false">prescription
purchases</Term>
        <Term caseSensitive="false">benefit eligibility</
Term>
        <Term caseSensitive="false">personal health</Term>
        <Term caseSensitive="false">power of attorney</
Term>
        <Term caseSensitive="false">registration number</
```

```
Term>
        <Term caseSensitive="false">personal health
number</Term>
        <Term caseSensitive="false">practitioner referral</
Term>
        <Term caseSensitive="false">wellness professional</
Term>
        <Term caseSensitive="false">patient referral</Term>
        <Term caseSensitive="false">health and wellness</
Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_canada_passport_number">
      <Group matchStyle="word">
        <Term caseSensitive="false">canadian citizenship</
Term>
        <Term caseSensitive="false">canadian passport</
Term>
        <Term caseSensitive="false">passport application</
Term>
        <Term caseSensitive="false">passport photos</Term>
        <Term caseSensitive="false">certified translator</
Term>
        <Term caseSensitive="false">canadian citizens</
Term>
        <Term caseSensitive="false">processing times</Term>
        <Term caseSensitive="false">renewal application</
Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_canada_bank_account_number">
      <Group matchStyle="word">
        <Term caseSensitive="false">canada savings bonds</
Term>
        <Term caseSensitive="false">canada revenue agency</
Term>
        <Term caseSensitive="false">canadian financial
institution</Term>
        <Term caseSensitive="false">direct deposit form</
```

```
Term>
        <Term caseSensitive="false">canadian citizen</Term>
        <Term caseSensitive="false">legal representative</
Term>
        <Term caseSensitive="false">notary public</Term>
        <Term caseSensitive="false">commissioner for
oaths</Term>
        <Term caseSensitive="false">child care benefit</
Term>
        <Term caseSensitive="false">universal child care</
Term>
        <Term caseSensitive="false">canada child tax
benefit</Term>
        <Term caseSensitive="false">income tax benefit</
Term>
        <Term caseSensitive="false">harmonized sales tax</
Term>
        <Term caseSensitive="false">social insurance
number</Term>
        <Term caseSensitive="false">income tax refund</
Term>
        <Term caseSensitive="false">child tax benefit</
Term>
        <Term caseSensitive="false">territorial payments</
Term>
        <Term caseSensitive="false">institution number</
Term>
        <Term caseSensitive="false">deposit request</Term>
        <Term caseSensitive="false">banking information</
Term>
        <Term caseSensitive="false">direct deposit</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_australia_passport_number">
      <Group matchStyle="word">
        <Term caseSensitive="false">passport</Term>
        <Term caseSensitive="false">passport details</Term>
        <Term caseSensitive="false">immigration and
citizenship</Term>
```

```
        <Term caseSensitive="false">commonwealth of
australia</Term>
        <Term caseSensitive="false">department of
immigration</Term>
        <Term caseSensitive="false">residential address</
Term>
        <Term caseSensitive="false">department of
immigration and citizenship</Term>
        <Term caseSensitive="false">visa</Term>
        <Term caseSensitive="false">national identity
card</Term>
        <Term caseSensitive="false">passport number</Term>
        <Term caseSensitive="false">travel document</Term>
        <Term caseSensitive="false">issuing authority</
Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_australia_drivers_license_number">
      <Group matchStyle="word">
        <Term caseSensitive="false">international driving
permits</Term>
        <Term caseSensitive="false">australian automobile
association</Term>
        <Term caseSensitive="false">sydney nsw</Term>
        <Term caseSensitive="false">international driving
permit</Term>
        <Term caseSensitive="false">aaa</Term>
        <Term caseSensitive="false">DriverLicense</Term>
        <Term caseSensitive="false">DriverLicenses</Term>
        <Term caseSensitive="false">Driver Lic</Term>
        <Term caseSensitive="false">Driver Lics</Term>
        <Term caseSensitive="false">Driver License</Term>
        <Term caseSensitive="false">Driver Licenses</Term>
        <Term caseSensitive="false">DriversLic</Term>
        <Term caseSensitive="false">DriversLics</Term>
        <Term caseSensitive="false">DriversLicense</Term>
        <Term caseSensitive="false">DriversLicenses</Term>
        <Term caseSensitive="false">Drivers Lic</Term>
        <Term caseSensitive="false">Drivers Lics</Term>
```

```
        <Term caseSensitive="false">Drivers License</Term>
        <Term caseSensitive="false">Drivers Licenses</Term>
        <Term caseSensitive="false">Driver'Lic</Term>
        <Term caseSensitive="false">Driver'Lics</Term>
        <Term caseSensitive="false">Driver'License</Term>
        <Term caseSensitive="false">Driver'Licenses</Term>
        <Term caseSensitive="false">Driver' Lic</Term>
        <Term caseSensitive="false">Driver' Lics</Term>
        <Term caseSensitive="false">Driver' License</Term>
        <Term caseSensitive="false">Driver' Licenses</Term>
        <Term caseSensitive="false">Driver'sLic</Term>
        <Term caseSensitive="false">Driver'sLics</Term>
        <Term caseSensitive="false">Driver'sLicense</Term>
        <Term caseSensitive="false">Driver'sLicenses</Term>
        <Term caseSensitive="false">Driver's Lic</Term>
        <Term caseSensitive="false">Driver's Lics</Term>
        <Term caseSensitive="false">Driver's License</Term>
        <Term caseSensitive="false">Driver's Licenses</
Term>
        <Term caseSensitive="false">DriverLic#</Term>
        <Term caseSensitive="false">DriverLics#</Term>
        <Term caseSensitive="false">DriverLicense#</Term>
        <Term caseSensitive="false">DriverLicenses#</Term>
        <Term caseSensitive="false">Driver Lic#</Term>
        <Term caseSensitive="false">Driver Lics#</Term>
        <Term caseSensitive="false">Driver License#</Term>
        <Term caseSensitive="false">Driver Licenses#</Term>
        <Term caseSensitive="false">DriversLic#</Term>
        <Term caseSensitive="false">DriversLics#</Term>
        <Term caseSensitive="false">DriversLicense#</Term>
        <Term caseSensitive="false">DriversLicenses#</Term>
        <Term caseSensitive="false">Drivers Lic#</Term>
        <Term caseSensitive="false">Drivers Lics#</Term>
        <Term caseSensitive="false">Drivers License#</Term>
        <Term caseSensitive="false">Drivers Licenses#</
Term>
        <Term caseSensitive="false">Driver'Lic#</Term>
        <Term caseSensitive="false">Driver'Lics#</Term>
        <Term caseSensitive="false">Driver'License#</Term>
```

```
        <Term caseSensitive="false">Driver'Licenses#</Term>
        <Term caseSensitive="false">Driver' Lic#</Term>
        <Term caseSensitive="false">Driver' Lics#</Term>
        <Term caseSensitive="false">Driver' License#</Term>
        <Term caseSensitive="false">Driver' Licenses#</
Term>
        <Term caseSensitive="false">Driver'sLic#</Term>
        <Term caseSensitive="false">Driver'sLics#</Term>
        <Term caseSensitive="false">Driver'sLicense#</Term>
        <Term caseSensitive="false">Driver'sLicenses#</
Term>
        <Term caseSensitive="false">Driver's Lic#</Term>
        <Term caseSensitive="false">Driver's Lics#</Term>
        <Term caseSensitive="false">Driver's License#</
Term>
        <Term caseSensitive="false">Driver's Licenses#</
Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_australia_bank_account_number">
      <Group matchStyle="word">
        <Term caseSensitive="false">swift bank code</Term>
        <Term caseSensitive="false">correspondent bank</
Term>
        <Term caseSensitive="false">base currency</Term>
        <Term caseSensitive="false">usa account</Term>
        <Term caseSensitive="false">holder address</Term>
        <Term caseSensitive="false">bank address</Term>
        <Term caseSensitive="false">information account</
Term>
        <Term caseSensitive="false">fund transfers</Term>
        <Term caseSensitive="false">bank charges</Term>
        <Term caseSensitive="false">bank details</Term>
        <Term caseSensitive="false">banking information</
Term>
        <Term caseSensitive="false">full names</Term>
        <Term caseSensitive="false">iaea</Term>
      </Group>
    </Keyword>
```

```
    <Keyword id="Keyword_sweden_passport">
      <Group matchStyle="word">
        <Term caseSensitive="false">visa requirements</
Term>
        <Term caseSensitive="false">Alien Registration
Card</Term>
        <Term caseSensitive="false">Schengen visas</Term>
        <Term caseSensitive="false">Schengen visa</Term>
        <Term caseSensitive="false">Visa Processing</Term>
        <Term caseSensitive="false">Visa Type</Term>
        <Term caseSensitive="false">Single Entry</Term>
        <Term caseSensitive="false">Multiple Entry</Term>
        <Term caseSensitive="false">G3 Processing Fees</
Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_italy_drivers_license_number">
      <Group matchStyle="word">
        <Term caseSensitive="false">numero di patente di
guida</Term>
        <Term caseSensitive="false">patente di guida</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_israel_bank_account_number">
      <Group matchStyle="string">
        <Term caseSensitive="false">Bank Account Number</
Term>
        <Term caseSensitive="false">Bank Account</Term>
        <Term caseSensitive="false">Account Number</Term>
        <Term caseSensitive="false">□□□ □□□□ □ □□□□</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_saudi_arabia_national_id">
      <Group matchStyle="word">
        <Term caseSensitive="false">Identification Card</
Term>
        <Term caseSensitive="false">I card number</Term>
        <Term caseSensitive="false">ID number</Term>
        <Term caseSensitive="false">□□ □□□□□□□□□□□□□□□□</Term>
```

```
      </Group>
    </Keyword>
    <Keyword id="Keyword_ipaddress">
      <Group matchStyle="string">
        <Term caseSensitive="false">ip address</Term>
        <Term caseSensitive="false">internet protocol</
Term>
        <Term caseSensitive="false">IP-□□□□□</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_pesel_identification_number">
      <Group matchStyle="word">
        <Term caseSensitive="false">Nr PESEL</Term>
        <Term caseSensitive="false">PESEL</Term>
      </Group>
    </Keyword>
    <Keyword
id="Keyword_polish_national_id_passport_number">
      <Group matchStyle="word">
        <Term caseSensitive="false">Nazwa i nr dowodu
tozsamosci</Term>
        <Term caseSensitive="false">Dowód Tozsamosci</Term>
        <Term caseSensitive="false">dow. os.</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_finnish_national_id">
      <Group matchStyle="word">
        <Term caseSensitive="false">Sosiaaliturvatunnus</
Term>
        <Term caseSensitive="false">SOTU</Term>
        <Term caseSensitive="false">Henkilötunnus</Term>
        <Term caseSensitive="false">HETU</Term>
        <Term caseSensitive="false">Personbeteckning</Term>
        <Term caseSensitive="false">Personnummer</Term>
      </Group>
    </Keyword>
    <Keyword id="Keyword_taiwanese_national_id">
      <Group matchStyle="string">
        <Term caseSensitive="false">身份證字號</Term>
```

```
        <Term caseSensitive="false">身份證</Term>
        <Term caseSensitive="false">身份證號碼</Term>
        <Term caseSensitive="false">身份證號</Term>
        <Term caseSensitive="false">身分證字號</Term>
        <Term caseSensitive="false">身分證</Term>
        <Term caseSensitive="false">身分證號碼</Term>
        <Term caseSensitive="false">身份證號</Term>
        <Term caseSensitive="false">身分證統一編號</Term>
        <Term caseSensitive="false">國民身分證統一編號</Term>
        <Term caseSensitive="false">簽名</Term>
        <Term caseSensitive="false">蓋章</Term>
        <Term caseSensitive="false">簽名或蓋章</Term>
        <Term caseSensitive="false">簽章</Term>
      </Group>
    </Keyword>
```

XML rule definitions

The following code sample includes all the default, out-of-box rules, which, to be honest, make the XML sample quite long. To make it easier to navigate the sample, use the table in Sensitive information type definitions to find the corresponding XML term for the sensitive information type that you want to locate. Then use the search feature of your web browser (usually **Ctrl+F**) to find the XML term in the sample, or you can copy the sample to a text editor and search from there. For an overview of how rules are applied to evaluate messages, see How DLP rules are applied to evaluate messages.

```
  <Rules>
    <Entity id="a181a86b-b800-4700-a3bd-6b07aa5d20c1"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_credit_card" />
        <Any minMatches="1">
          <Match idRef="Keyword_cc_verification" />
          <Match idRef="Keyword_cc_name" />
          <Match idRef="Func_expiration_date" />
      </Pattern>
    </Entity>
    <Entity id="0e9b3178-9678-47dd-a509-37222ca96b42"
patternsProximity="300" recommendedConfidence="85">
```

```
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_eu_debit_card" />
        <Any minMatches="1">
          <Match idRef="Keyword_eu_debit_card" />
          <Match idRef="Keyword_card_terms_dict" />
          <Match idRef="Keyword_card_security_terms_dict" /
>
          <Match idRef="Keyword_card_expiration_terms_dict"
/>
          <Match idRef="Func_expiration_date" />
          <Match idRef="Func_eu_date" />
          <Match idRef="Func_eu_date1" />
          <Match idRef="Func_eu_date2" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="a44669fe-0d48-453d-a9b1-2cc83f2cba77"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_ssn" />
        <Any minMatches="1">
          <Match idRef="Keyword_ssn" />
          <Match idRef="Func_us_date" />
          <Match idRef="Func_us_address" />
        </Any>
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_unformatted_ssn" />
        <Match idRef="Keyword_ssn" />
        <Any minMatches="1">
          <Match idRef="Func_us_date" />
          <Match idRef="Func_us_address" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="e55e2a32-f92d-4985-a35d-a0b269eb687b"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_formatted_itin" />
```

```
        <Any minMatches="1">
          <Match idRef="Keyword_itin" />
          <Match idRef="Func_us_address" />
          <Match idRef="Func_us_date" />
          <Match idRef="Keyword_itin_collaborative" />
        </Any>
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_unformatted_itin" />
        <Match idRef="Keyword_itin" />
        <Any minMatches="1">
          <Match idRef="Keyword_itin_collaborative" />
          <Match idRef="Func_us_address" />
          <Match idRef="Func_us_date" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="a2f29c85-ecb8-4514-a610-364790c0773e"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_canadian_sin" />
        <Any minMatches="2">
          <Match idRef="Keyword_sin" />
          <Match idRef="Keyword_sin_collaborative" />
          <Match idRef="Func_us_date" />
        </Any>
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_unformatted_canadian_sin" />
        <Match idRef="Keyword_sin" />
      </Pattern>
    </Entity>
    <Entity id="16c07343-c26f-49d2-a987-3daf717e94cc"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_uk_nino" />
        <Any minMatches="0" maxMatches="0">
          <Match idRef="Keyword_uk_nino" />
        </Any>
```

```
      </Pattern>
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_uk_nino" />
        <Any minMatches="1">
          <Match idRef="Keyword_uk_nino" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="f93de4be-d94c-40df-a8be-461738047551"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_uk_drivers_license" />
        <Match idRef="Keyword_uk_drivers_license" />
      </Pattern>
    </Entity>
    <Entity id="91da9335-1edb-45b7-a95f-5fe41a16c63c"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_german_drivers_license" />
        <Any minMatches="1">
          <Match
idRef="Keyword_german_drivers_license_number" />
          <Match
idRef="Keyword_german_drivers_license_collaborative" />
          <Match idRef="Keyword_german_drivers_license" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="2e3da144-d42b-47ed-b123-fbf78604e52c"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_german_passport" />
        <Any minMatches="1">
          <Match idRef="Keyword_german_passport" />
          <Match
idRef="Keyword_german_passport_collaborative" />
          <Match idRef="Keyword_german_passport_number" />
          <Match idRef="Keyword_german_passport1" />
          <Match idRef="Keyword_german_passport2" />
```

```
        </Any>
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_german_passport_data" />
        <Any minMatches="1">
          <Match idRef="Keyword_german_passport" />
          <Match
idRef="Keyword_german_passport_collaborative" />
          <Match idRef="Keyword_german_passport_number" />
          <Match idRef="Keyword_german_passport1" />
          <Match idRef="Keyword_german_passport2" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="3192014e-2a16-44e9-aa69-4b20375c9a78"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_uk_nhs_number" />
        <Any minMatches="1">
          <Match idRef="Keyword_uk_nhs_number" />
          <Match idRef="Keyword_uk_nhs_number1" />
          <Match idRef="Keyword_uk_nhs_number_dob" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="71f62b97-efe0-4aa1-aa49-e14de253619d"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_french_insee" />
        <Match idRef="Func_fr_insee" />
        <Any minMatches="0" maxMatches="0">
          <Match idRef="Keyword_fr_insee" />
        </Any>
      </Pattern>
      <Pattern confidenceLevel="95">
        <IdMatch idRef="Func_french_insee" />
        <Match idRef="Func_fr_insee" />
        <Any minMatches="1">
          <Match idRef="Keyword_fr_insee" />
```

```
        </Any>
      </Pattern>
    </Entity>
    <Entity id="18e55a36-a01b-4b0f-943d-dc10282a1824"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_french_drivers_license" />
        <Any minMatches="1">
          <Match idRef="Keyword_french_drivers_license" />
          <Match idRef="Func_eu_date" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="37186abb-8e48-4800-ad3c-e3d1610b3db0"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_alberta_drivers_license_number" />
        <Match idRef="Keyword_alberta_drivers_license_name"
/>
        <Match idRef="Keyword_canada_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_british_columbia_drivers_license_number" />
        <Match
idRef="Keyword_british_columbia_drivers_license_name" />
        <Match idRef="Keyword_canada_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_manitoba_drivers_license_number" />
        <Match
idRef="Keyword_manitoba_drivers_license_name" />
        <Match idRef="Keyword_canada_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_new_brunswick_drivers_license_number" />
```

```
        <Match
idRef="Keyword_new_brunswick_drivers_license_name" />
        <Match idRef="Keyword_canada_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_newfoundland_labrador_drivers_license_number" /
>
        <Match
idRef="Keyword_newfoundland_labrador_drivers_license_name"
/>
        <Match idRef="Keyword_canada_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_nova_scotia_drivers_license_number" />
        <Match
idRef="Keyword_nova_scotia_drivers_license_name" />
        <Match idRef="Keyword_canada_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_ontario_drivers_license_number" />
        <Match idRef="Keyword_ontario_drivers_license_name"
/>
        <Match idRef="Keyword_canada_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_prince_edward_island_drivers_license_number" />
        <Match
idRef="Keyword_prince_edward_island_drivers_license_name" /
>
        <Match idRef="Keyword_canada_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_quebec_drivers_license_number"
/>
        <Match
```

```
idRef="Keyword_quebec_drivers_license_name" />
        <Match idRef="Keyword_canada_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_saskatchewan_drivers_license_number" />
        <Match
idRef="Keyword_saskatchewan_drivers_license_name" />
        <Match idRef="Keyword_canada_drivers_license" />
      </Pattern>
    </Entity>
    <Entity id="dfeb356f-61cd-459e-bf0f-7c6d28b458c6"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_alabama_drivers_license_number" />
        <Match idRef="Keyword_alabama_drivers_license_name"
/>
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_alaska_delaware_oregon_drivers_license_number"
/>
        <Match
idRef="Keyword_alaska_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_arizona_drivers_license_number" />
        <Match idRef="Keyword_arizona_drivers_license_name"
/>
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_arkansas_drivers_license_number" />
        <Match
```

```
idRef="Keyword_arkansas_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_california_drivers_license_number" />
        <Match
idRef="Keyword_california_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_colorado_drivers_license_number" />
        <Match
idRef="Keyword_colorado_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_connecticut_drivers_license_number" />
        <Match
idRef="Keyword_connecticut_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_alaska_delaware_oregon_drivers_license_number"
/>
        <Match
idRef="Keyword_delaware_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_district_of_columbia_drivers_license_number" />
        <Match
idRef="Keyword_district_of_columbia_drivers_license_name" /
>
        <Match idRef="Keyword_us_drivers_license" />
```

```
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_florida_maryland_michigan_minnesota_drivers_lic
ense_number" />
        <Match idRef="Keyword_florida_drivers_license_name"
/>
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_georgia_drivers_license_number" />
        <Match idRef="Keyword_georgia_drivers_license_name"
/>
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_hawaii_drivers_license_number"
/>
        <Match
idRef="Keyword_hawaii_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_idaho_drivers_license_number" />
        <Match idRef="Keyword_idaho_drivers_license" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_illinois_drivers_license_number" />
        <Match
idRef="Keyword_illinois_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_indiana_drivers_license_number" />
```

```
        <Match idRef="Keyword_indiana_drivers_license" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_iowa_drivers_license_number" /
>
        <Match idRef="Keyword_iowa_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_kansas_drivers_license_number"
/>
        <Match
idRef="Keyword_kansas_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_kentucky_massachusetts_virginia_drivers_license
_number" />
        <Match
idRef="Keyword_kentucky_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_louisiana_drivers_license_number" />
        <Match
idRef="Keyword_louisiana_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_maine_drivers_license_number" />
        <Match idRef="Keyword_maine_drivers_license_name" /
>
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
```

```
        <IdMatch
idRef="Func_florida_maryland_michigan_minnesota_drivers_lic
ense_number" />
        <Match
idRef="Keyword_maryland_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_kentucky_massachusetts_virginia_drivers_license
_number" />
        <Match
idRef="Keyword_massachusetts_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_florida_maryland_michigan_minnesota_drivers_lic
ense_number" />
        <Match
idRef="Keyword_michigan_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_florida_maryland_michigan_minnesota_drivers_lic
ense_number" />
        <Match
idRef="Keyword_minnesota_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_mississippi_oklahoma_drivers_license_number" />
        <Match
idRef="Keyword_mississippi_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
```

```
        <IdMatch
idRef="Func_missouri_drivers_license_number" />
        <Match
idRef="Keyword_missouri_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_montana_drivers_license_number" />
        <Match idRef="Keyword_montana_drivers_license_name"
/>
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_nebraska_drivers_license_number" />
        <Match
idRef="Keyword_nebraska_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_nevada_drivers_license_number"
/>
        <Match
idRef="Keyword_nevada_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_new_hampshire_drivers_license_number" />
        <Match
idRef="Keyword_new_hampshire_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_new_jersey_drivers_license_number" />
        <Match
idRef="Keyword_new_jersey_drivers_license_name" />
```

```
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_new_mexico_drivers_license_number" />
        <Match
idRef="Keyword_new_mexico_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_new_york_drivers_license_number" />
        <Match
idRef="Keyword_new_york_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_north_carolina_drivers_license_number" />
        <Match
idRef="Keyword_north_carolina_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_north_dakota_drivers_license_number" />
        <Match
idRef="Keyword_north_dakota_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_ohio_drivers_license_number" /
>
        <Match idRef="Keyword_ohio_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_mississippi_oklahoma_drivers_license_number" />
```

```
        <Match
idRef="Keyword_oklahoma_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_alaska_delaware_oregon_drivers_license_number"
/>
        <Match
idRef="Keyword_oregon_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_pennsylvania_drivers_license_number" />
        <Match
idRef="Keyword_pennsylvania_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_rhode_island_drivers_license_number" />
        <Match
idRef="Keyword_rhode_island_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_south_carolina_drivers_license_number" />
        <Match
idRef="Keyword_south_carolina_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_south_dakota_drivers_license_number" />
        <Match
idRef="Keyword_south_dakota_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
```

```
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_tennessee_drivers_license_number" />
        <Match
idRef="Keyword_tennessee_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_texas_drivers_license_number" />
        <Match idRef="Keyword_texas_drivers_license_name" /
>
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_utah_drivers_license_number" /
>
        <Match idRef="Keyword_utah_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_vermont_drivers_license_number" />
        <Match idRef="Keyword_vermont_drivers_license_name"
/>
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_kentucky_massachusetts_virginia_drivers_license
_number" />
        <Match
idRef="Keyword_virginia_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_washington_drivers_license_number" />
```

```
        <Match
idRef="Keyword_washington_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_west_virginia_drivers_license_number" />
        <Match
idRef="Keyword_west_virginia_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_wisconsin_drivers_license_number" />
        <Match
idRef="Keyword_wisconsin_drivers_license_name" />
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_wyoming_drivers_license_number" />
        <Match idRef="Keyword_wyoming_drivers_license_name"
/>
        <Match idRef="Keyword_us_drivers_license" />
      </Pattern>
    </Entity>
    <Entity id="c6011143-d087-451c-8313-7f6d4aed2270"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_jp_drivers_license_number" />
        <Match idRef="Keyword_jp_drivers_license_number" />
      </Pattern>
    </Entity>
    <Entity id="01c1209b-6389-4faf-a5f8-3f7e13899652"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Func_jp_resident_registration_number" />
        <Match
```

```
idRef="Keyword_jp_resident_registration_number" />
      </Pattern>
    </Entity>
    <Entity id="c840e719-0896-45bb-84fd-1ed5c95e45ff"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_jp_sin" />
        <Match idRef="Keyword_jp_sin" />
      </Pattern>
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_jp_sin_pre_1997" />
        <Match idRef="Keyword_jp_sin" />
      </Pattern>
    </Entity>
    <Entity id="75177310-1a09-4613-bf6d-833aae3743f8"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_jp_passport" />
        <Match idRef="Keyword_jp_passport" />
      </Pattern>
    </Entity>
    <Entity id="d354f95b-96ee-4b80-80bc-4377312b55bc"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_jp_bank_account" />
        <Match idRef="Keyword_jp_bank_account" />
      </Pattern>
    </Entity>
    <Entity id="3008b884-8c8c-4cd8-a289-99f34fc7ff5d"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_fr_passport" />
        <Match idRef="Keyword_passport" />
      </Pattern>
    </Entity>
    <Entity id="178ec42a-18b4-47cc-85c7-d62c92fd67f8"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_usa_uk_passport" />
```

```
        <Match idRef="Keyword_passport" />
      </Pattern>
    </Entity>
    <Entity id="cb2ab58c-9cb8-4c81-baf8-a4e106791df4"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_swift" />
        <Match idRef="Keyword_swift" />
      </Pattern>
    </Entity>
    <Entity id="a2ce32a8-f935-4bb6-8e96-2a5157672e2c"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_usa_bank_account_number" />
        <Match idRef="Keyword_usa_Bank_Account" />
      </Pattern>
    </Entity>
    <Entity id="cb353f78-2b72-4c3c-8827-92ebe4f69fdf"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_aba_routing" />
        <Match idRef="Keyword_ABA_Routing" />
      </Pattern>
    </Entity>
    <Entity id="9a5445ad-406e-43eb-8bd7-cac17ab6d0e4"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_dea_number" />
      </Pattern>
    </Entity>
    <Entity id="104a99a0-3d3b-4542-a40d-ab0b9e1efe63"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch
idRef="Func_australian_medical_account_number" />
        <Any minMatches="0" maxMatches="0">
          <Match
idRef="Keyword_Australia_Medical_Account_Number" />
        </Any>
```

```
      </Pattern>
      <Pattern confidenceLevel="95">
        <IdMatch
idRef="Func_australian_medical_account_number" />
        <Any minMatches="1">
          <Match
idRef="Keyword_Australia_Medical_Account_Number" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="e29bc95f-ff70-4a37-aa01-04d17360a4c5"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_australian_tax_file_number" />
        <Any minMatches="0" maxMatches="0">
          <Match
idRef="Keyword_Australia_Tax_File_Number" />
          <Match idRef="Keyword_number_exclusions" />
        </Any>
      </Pattern>
      <Pattern confidenceLevel="95">
        <IdMatch idRef="Func_australian_tax_file_number" />
        <Any minMatches="1">
          <Match
idRef="Keyword_Australia_Tax_File_Number" />
        </Any>
        <Any minMatches="0" maxMatches="0">
          <Match idRef="Keyword_number_exclusions" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="e05881f5-1db1-418c-89aa-a3ac5c5277ee"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Func_israeli_national_id_number" />
        <Any minMatches="1">
          <Match idRef="Keyword_Israel_National_ID" />
        </Any>
      </Pattern>
```

```
    </Entity>
    <Entity id="2b71c1c8-d14e-4430-82dc-fd1ed6bf05c7"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch
idRef="Func_new_zealand_ministry_of_health_number" />
        <Any minMatches="1">
          <Match idRef="Keyword_nz_terms" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="5df987c0-8eae-4bce-ace7-b316347f3070"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch
idRef="Func_spanish_social_security_number" />
      </Pattern>
    </Entity>
    <Entity id="f69aaf40-79be-4fac-8f05-fd1910d272c8"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_swedish_national_identifier" /
>
      </Pattern>
    </Entity>
    <Entity id="74a54de9-2a30-4aa0-a8aa-3d9327fc07c7"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch
idRef="Regex_australia_bank_account_number" />
        <Any minMatches="1">
          <Match
idRef="Keyword_australia_bank_account_number" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="1cbbc8f5-9216-4392-9eb5-5ac2298d1356"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
```

```
        <IdMatch
idRef="Regex_australia_drivers_license_number" />
        <Any minMatches="1">
          <Match
idRef="Keyword_australia_drivers_license_number" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="29869db6-602d-4853-ab93-3484f905df50"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_australia_passport_number" />
        <Any minMatches="1">
          <Match idRef="Keyword_passport" />
          <Match
idRef="Keyword_australia_passport_number" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="552e814c-cb50-4d94-bbaa-bb1d1ffb34de"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_canada_bank_account_number" /
>
        <Any minMatches="1">
          <Match idRef="Keyword_canada_bank_account_number"
/>
        </Any>
      </Pattern>
    </Entity>
    <Entity id="14d0db8b-498a-43ed-9fca-f6097ae687eb"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_canada_passport_number" />
        <Any minMatches="1">
          <Match idRef="Keyword_canada_passport_number" />
          <Match idRef="Keyword_passport" />
        </Any>
      </Pattern>
```

```
    </Entity>
    <Entity id="722e12ac-c89a-4ec8-a1b7-fea3469f89db"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_canada_phin" />
        <Any minMatches="2">
          <Match idRef="Keyword_canada_phin" />
          <Match idRef="Keyword_canada_provinces" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="59c0bf39-7fab-482c-af25-00faa4384c94"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_canada_health_service_number"
/>
        <Any minMatches="1">
          <Match
idRef="Keyword_canada_health_service_number" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="f741ac74-1bc0-4665-b69b-f0c7f927c0c4"
patternsProximity="300" recommendedConfidence="65">
      <Pattern confidenceLevel="65">
        <IdMatch idRef="Regex_france_cni" />
      </Pattern>
    </Entity>
    <Entity id="1daa4ad5-e2dd-4ca4-a788-54722c09efb2"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Regex_ipv4_address" />
        <Any minMatches="0" maxMatches="0">
          <Match idRef="Keyword_ipaddress" />
        </Any>
      </Pattern>
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Regex_ipv6_address" />
        <Any minMatches="0" maxMatches="0">
```

```
          <Match idRef="Keyword_ipaddress" />
        </Any>
      </Pattern>
      <Pattern confidenceLevel="95">
        <IdMatch idRef="Regex_ipv4_address" />
        <Any minMatches="1">
          <Match idRef="Keyword_ipaddress" />
        </Any>
      </Pattern>
      <Pattern confidenceLevel="95">
        <IdMatch idRef="Regex_ipv6_address" />
        <Any minMatches="1">
          <Match idRef="Keyword_ipaddress" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="e7dc4711-11b7-4cb0-b88b-2c394a771f0e"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_iban" />
      </Pattern>
    </Entity>
    <Entity id="7d08b2ff-a0b9-437f-957c-aeddbf9b2b25"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_israel_bank_account_number" /
>
        <Any minMatches="1">
          <Match idRef="Keyword_israel_bank_account_number"
/>
        </Any>
      </Pattern>
    </Entity>
    <Entity id="97d6244f-9157-41bd-8e0c-9d669a5c4d71"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_italy_drivers_license_number"
/>
        <Any minMatches="1">
```

```
          <Match
idRef="Keyword_italy_drivers_license_number" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="8c5a0ba8-404a-41a3-8871-746aa21ee6c0"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_saudi_arabia_national_id" />
        <Any minMatches="1">
          <Match idRef="Keyword_saudi_arabia_national_id" /
>
        </Any>
      </Pattern>
    </Entity>
    <Entity id="ba4e7456-55a9-4d89-9140-c33673553526"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_sweden_passport_number" />
        <Any minMatches="1">
          <Match idRef="Keyword_passport" />
          <Match idRef="Keyword_sweden_passport" />
        </Any>
      </Pattern>
    </Entity>
    <Entity id="a3eea206-dc0c-4f06-9e22-aa1be3059963"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_uk_electoral" />
        <Any minMatches="1">
          <Match idRef="Keyword_uk_electoral" />
        </Any>
      </Pattern>
    </Entity>
<Version minEngineVersion="15.0.847.013">
  <Entity id="338FD995-4CB5-4F87-AD35-79BD1DD926C1"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_finnish_national_id" />
```

```
        <Match idRef="Keyword_finnish_national_id" />
      </Pattern>
    </Entity>
  <Entity id="4C7BFC34-8DD1-421D-8FB7-6C6182C2AF03"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_taiwanese_national_id" />
        <Match idRef="Keyword_taiwanese_national_id" />
      </Pattern>
    </Entity>
  <Entity id="E3AAF206-4297-412F-9E06-BA8487E22456"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_pesel_identification_number" /
>
        <Match
idRef="Keyword_pesel_identification_number" />
      </Pattern>
    </Entity>
  <Entity id="25E64989-ED5D-40CA-A939-6C14183BB7BF"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_polish_national_id" />
        <Match
idRef="Keyword_polish_national_id_passport_number" />
      </Pattern>
    </Entity>
  <Entity id="03937FB5-D2B6-4487-B61F-0F8BFF7C3517"
patternsProximity="300" recommendedConfidence="85">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Func_polish_passport_number" />
        <Match
idRef="Keyword_polish_national_id_passport_number" />
      </Pattern>
    </Entity>
</Version>
```

For more information

Manage DLP policies

Integrating sensitive information rules with transport rules

What's new for transport rules

Data loss prevention

DLP policy templates supplied in Exchange

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-04*

In Microsoft Exchange Server 2013 and Exchange Online, you can use data loss prevention (DLP) policy templates as a starting point for building DLP policies that help you meet your specific regulatory and business policy needs. You can modify the templates to meet the specific needs of your organization.

Caution:
You should enable your DLP policies in test mode before running them in your production environment. During such tests, it is recommended that you configure sample user mailboxes and send test messages that invoke your test policies in order to confirm the results. Use of these policies does not ensure compliance with any regulation. After your testing is complete, make the necessary configuration changes in Exchange so the transmission of information complies with your organization's policies. For example, you might need to configure TLS with known business partners or add more restrictive transport rule actions, such as adding rights protection to messages that contain a certain type of data.

Templates available for DLP

The following table lists the DLP policy templates in Exchange. To learn more about customizing these templates to create DLP policies, see Manage DLP policies.

Template	Description
Australia Financial Data	Helps detect the presence of information commonly considered to be financial data in Australia, including credit cards, and SWIFT codes.

Australia Health Records Act (HRIP Act)	Helps detect the presence of information commonly considered to be subject to the Health Records and Information Privacy (HRIP) act in Australia, like medical account number and tax file number.
Australia Personally Identifiable Information (PII) Data	Helps detect the presence of information commonly considered to be personally identifiable information (PII) in Australia, like tax file number and driver's license.
Australia Privacy Act	Helps detect the presence of information commonly considered to be subject to the privacy act in Australia, like driver's license and passport number.
Canada Financial Data	Helps detect the presence of information commonly considered to be financial data in Canada, including bank account numbers and credit cards.
Canada Health Information Act (HIA)	Helps detect the presence of information subject to Canada Health Information Act (HIA) for Alberta, including data like passport numbers and health information.
Canada Personal Health Act (PHIPA) - Ontario	Helps detect the presence of information subject to Canada Personal Health Information Protection Act (PHIPA) for Ontario, including data like passport numbers and health information.
Canada Personal Health Information Act (PHIA) - Manitoba	Helps detect the presence of information subject to Canada Personal Health Information Act (PHIA) for Manitoba, including data like health information.

Canada Personal Information Protection Act (PIPA)	Helps detect the presence of information subject to Canada Personal Information Protection Act (PIPA) for British Columbia, including data like passport numbers and health information.
Canada Personal Information Protection Act (PIPEDA)	Helps detect the presence of information subject to Canada Personal Information Protection and Electronic Documents Act (PIPEDA), including data like passport numbers and health information.
Canada Personally Identifiable Information (PII) Data	Helps detect the presence of information commonly considered to be personally identifiable information (PII) in Canada, like health ID number and social insurance number.
France Data Protection Act	Helps detect the presence of information commonly considered to be subject to the Data Protection Act in France, like the health insurance card number.
France Financial Data	Helps detect the presence of information commonly considered to be financial information in France, including information like credit card, account information, and debit card numbers.
France Personally Identifiable Information (PII) Data	Helps detect the presence of information commonly considered to be personally identifiable information (PII) in France, including information like passport numbers.
Germany Financial Data	Helps detect the presence of information commonly considered to be financial data in Germany like EU debit card numbers.

Germany Personally Identifiable Information (PII) Data	Helps detect the presence of information commonly considered to be personally identifiable information (PII) in Germany, including information like driver's license and passport numbers.
Israel Financial Data	Helps detect the presence of information commonly considered to be financial data in Israel, including bank account numbers and SWIFT codes.
Israel Personally Identifiable Information (PII) Data	Helps detect the presence of information commonly considered to be personally identifiable information (PII) in Israel, like national ID number.
Israel Protection of Privacy	Helps detect the presence of information commonly considered to be subject to the Protection of Privacy in Israel, including information like bank account numbers or national ID.
Japan Financial Data	Helps detect the presence of information commonly considered to be financial information in Japan, including information like credit card, account information, and debit card numbers.
Japan Personally Identifiable Information (PII) Data	Helps detect the presence of information commonly considered to be personally identifiable information (PII) in Japan, including information like driver's license and passport numbers.
Japan Protection of Personal Information	Helps detect the presence of information subject to Japan Protection of Personal

	Information, including data like resident registration numbers.
PCI Data Security Standard (PCI DSS)	Helps detect the presence of information subject to PCI Data Security Standard (PCI DSS), including information like credit card or debit card numbers.
Saudi Arabia - Anti-Cyber Crime Law	Helps detect the presence of information commonly considered to be subject to the Anti-Cyber Crime Law in Saudi Arabia, including international bank account numbers and SWIFT codes.
Saudi Arabia Financial Data	Helps detect the presence of information commonly considered to be financial data in Saudi Arabia, including international bank account numbers and SWIFT codes.
Saudi Arabia Personally Identifiable Information (PII) Data	Helps detect the presence of information commonly considered to be personally identifiable information (PII) in Saudi Arabia, like national ID number.
U.K. Access to Medical Reports Act	Helps detect the presence of information subject to United Kingdom Access to Medical Reports Act, including data like National Health Service numbers.
U.K. Data Protection Act	Helps detect the presence of information subject to United Kingdom Data Protection Act, including data like national insurance numbers.
U.K. Financial Data	Helps detect the presence of information commonly considered to be financial information in United Kingdom, including

	information like credit card, account information, and debit card numbers.
U.K. Personal Information Online Code of Practice (PIOCP)	Helps detect the presence of information subject to United Kingdom Personal Information Online Code of Practice, including data like health information.
U.K. Personally Identifiable Information (PII) Data	Helps detect the presence of information commonly considered to be personally identifiable information (PII) in United Kingdom, including information like driver's license and passport numbers.
U.K. Privacy and Electronic Communications Regulations	Helps detect the presence of information subject to United Kingdom Privacy and Electronic Communications Regulations, including data like financial information.
U.S. Federal Trade Commission (FTC) Consumer Rules	Helps detect the presence of information subject to U.S. Federal Trade Commission (FTC) Consumer Rules, including data like credit card numbers.
U.S. Financial Data	Helps detect the presence of information commonly considered to be financial information in United States, including information like credit card, account information, and debit card numbers.
U.S. Gramm-Leach-Bliley Act (GLBA)	Helps detect the presence of information subject to Gramm-Leach-Bliley Act (GLBA), including information like social security numbers or credit card numbers.
U.S. Health Insurance Act (HIPAA)	Helps detect the presence of information

	subject to United States Health Insurance Portability and Accountability Act (HIPAA),including data like social security numbers and health information.
U.S. Patriot Act	Helps detect the presence of information commonly subject to U.S. Patriot Act, including information like credit card numbers or tax identification numbers.
U.S. Personally Identifiable Information (PII) Data	Helps detect the presence of information commonly considered to be personally identifiable information (PII) in the United States, including information like social security numbers or driver's license numbers.
U.S. State Breach Notification Laws	Helps detect the presence of information subject to U.S. State Breach Notification Laws, including data like social security and credit card numbers.
U.S. State Social Security Number Confidentiality Laws	Helps detect the presence of information subject to U.S. State Social Security Number Confidentiality Laws, including data like social security numbers.

For more information

Data loss prevention

Create a DLP policy from a template

Sensitive information types inventory

Manage DLP policies

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-03*

You can view, change, or remove existing data loss prevention (DLP) policies in Microsoft Exchange Server 2013, using the Exchange Administration Center (EAC) or the Exchange Management Shell.

For additional management tasks related to DLP, see DLP procedures.

For more information about The Exchange Management Shell, see Exchange Management Shell.

What do you need to know before you begin?

- Estimated time to complete each procedure: 15-60 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.
- For any DLP policy, you can select one of three modes:
-
- **Enforce** Rules within the policy are evaluated for all messages and supported file types. Mail flow can be disrupted if data is detected that meets the conditions of the policy. All actions described within the policy are taken.
-
- **Test DLP policy with Policy Tips** Rules within the policy are evaluated for all messages and supported file types. Mail flow will not be disrupted if data is detected that meets the conditions of the policy. That is, messages are not blocked. If Policy Tips are configured, they are shown to users.
-
- **Test DLP policy without Policy Tips** Rules within the policy are evaluated for all messages and supported file types. Mail flow will not be disrupted if data is detected that meets the conditions of the policy. That is, messages are not blocked. If Policy Tips are configured, they are not shown to users.
- An individual rule within a DLP policy can have its own mode settings. When the mode of a policy is different than the mode of a rule within that policy, the rule setting has priority and will be evaluated according to its mode.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

View the details of an existing DLP policy

You may need to view the rules and actions of an existing DLP policy that you have already established for your organization. This can be useful if you experience unexpected mail flow issues or if your organization changes the way sensitive information needs to be monitored.

Use the EAC to view the details within an existing DLP policy

1. In the EAC, navigate to **Compliance management** > **Data loss prevention**.
2. Double-click one of the policies that appear in your list of policies, or highlight one item and click **Edit** ✎.
3. On the **Edit DLP policy** page, click **Rules**.

Tip:
You can create a DLP policy and leave it in a non-activated or disabled mode. In this mode, a policy is not enforced and you can change any predicates, actions, or values associated with its rules before you test or begin enforcing it.

Use the Shell to view the details within an existing DLP policy

This example returns information about the fictitious DLP policy named Employee Numbers. The command is piped to the **Format-List** cmdlet to display the detailed configuration of the specified DLP policy.

```
Get-DlpPolicy "Employee Numbers" | Format-List
```

For syntax and parameter information, see Get-DlpPolicy.

Change a DLP policy

You can change an existing DLP policy by modifying either the name of the policy or the rules that govern the effects of the policy. An example rule change might include adding custom disclaimer text to a message body and RMS protection for messages sent within a specific domain and that are detected to have sensitive information. If you are using DLP policy templates, keep in mind that these are only one of the features in Exchange 2013 that can help you design and apply a robust policy and compliance system for your messaging environment.

Use the EAC to change an existing DLP policy

1. In the EAC, navigate to **Compliance management** > **Data loss prevention**.
2. Double-click one of the template-based policies that appear in your list of policies or highlight one item and click **Edit** ✎.
3. On the **Edit DLP policy** page, click **Rules**.
4. To change an existing rule, highlight the rule and click **Edit** ✎.
5. To add a new blank rule that you can fully customize, click **New** +.
6. To add a rule about sender notification, blocking messages, or allowing overrides, click the arrow

next to the **New +** icon.

7. To remove a rule, highlight the rule and click **Delete** 🗑.
8. Click **Save** to finish modifying the policy and save your changes.

Use the Shell to change an existing DLP policy

You can specify the action and notification level of a policy using the Exchange Management Shell. This example sets the mode for a fictitious DLP policy named Employee Numbers so that the actions are not enforced and notification messages are not displayed.

```
Set-DlpPolicy "Employee Numbers" -Mode Audit
```

For syntax and parameter information, see Set-DlpPolicy.

Delete a DLP policy

You can permanently remove a DLP policy using the EAC. Once you've deleted a policy, it will no longer be enforced and none of the rules and actions will be saved.

Alternatively, you can set the operational state or mode of a policy to **Test DLP policy without Policy Tips**. This stops it from being enforced in your message environment, but preserves the detailed configuration settings of the policy itself. This can be useful if there is a possibility that you will need to enforce the policy again in the future.

Use the EAC to delete an existing DLP policy

1. In the EAC, navigate to **Compliance management** > **Data loss prevention**.
2. Select the policy you want to remove in your list of policies, and then click **Delete** 🗑.

Use the Shell to delete an existing DLP policy

This example removes the fictitious DLP policy named Employee Numbers.

```
Remove-DlpPolicy "Employee Numbers"
```

For syntax and parameter information, see Remove-DlpPolicy.

For more information

Data loss prevention

Policy Tips

Create a DLP policy from a template

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-07-31*

In Microsoft Exchange Server 2013, you can use data loss prevention (DLP) policy templates to help meet the messaging policy and compliance needs of your organization. These templates contain pre-built sets of rules that can help you manage message data that is associated with several common legal and regulatory requirements. To see a list of all the templates supplied by Microsoft, see DLP policy templates supplied in Exchange. Example DLP templates that are supplied can help you manage:

- Gramm-Leach-Bliley Act (GLBA) data
- Payment Card Industry Data Security Standard (PCI-DSS)
- United States Personally Identifiable Information (U.S. PII)

You can customize any of these DLP templates or use them as-is. DLP policy templates are built on top of transport rules that include new conditions or predicates and actions. DLP policies support the full range of traditional transport rules, and you can add the additional rules after a DLP policy has been established. For more information about policy templates, see DLP policy templates. To learn more about transport rule capabilities, see Transport rules. Once you have started enforcing a policy, you can learn about how to observe the results by reviewing the following topics:

Exchange 2013: DLP policy detection reports

Exchange Online: **Summary data reports for DLP policies**

Caution:

You should enable your DLP policies in test mode before running them in your production environment. During such tests, it is recommended that you configure sample user mailboxes and send test messages that invoke your test policies in order to confirm the results.

For additional management tasks related to creating a DLP policy from a template, see DLP procedures.

What do you need to know before you begin?

- Estimated time to complete: 30 minutes
- Ensure that Exchange 2013 is setup as described in Planning and deployment.
- Configure both administrator and user accounts within your organization and validate basic mail flow.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the “Data loss prevention (DLP)” entry in the Messaging policy and compliance permissions topic
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to configure a DLP policy from a template

1. In the EAC, navigate to **Compliance management** > **Data loss prevention**, and then click **Add +**
   .

Note:
You can also select this action if you click the arrow next to the **Add +** icon and select **New DLP policy from template** from the drop down menu.

2. On the **Create a new DLP policy from a template** page, complete the following fields:
   a. **Name** Add a name that will distinguish this policy from others.
   b. **Description** Add an optional description that summarizes this policy.
   c. **Choose a template** Select the appropriate template to begin creating a new policy.
   d. **More options** Select the mode or state. The new policy is not fully enabled until you specify that it should be. The default mode for a policy is test without notifications.
   e. Click **Save** to finish creating the policy.

Note:
In addition to the rules within a specific template, your organization may have additional expectations or company policies that apply to regulated data within your messaging environment. Exchange 2013 makes it easy for you to change the basic template in order to add actions so that your Exchange messaging environment complies with your own requirements.

You can modify policies by editing the rules within them once the policy has been saved in your Exchange 2013 environment. An example rule change might include making specific people exempt from a policy or sending a notice and blocking message delivery if a message is found to have sensitive content. For more information about editing policies and rules, see Manage DLP policies.

You have to navigate to the specific policy's rule set on the **Edit DLP policy** page and use the tools available on that page in order to change a DLP policy you have already created in Exchange 2013.

Some policies allow the addition of rules that invoke RMS for messages. You must have RMS configured on the Exchange server before adding the actions to make use of these types of rules.

For any of the DLP policies, you can change the rules, actions, exceptions, enforcement time period or whether other rules within the policy are enforced and you can add your own custom conditions for each.

For more information

Data loss prevention

Create a custom DLP policy

Messaging policy and compliance > Data loss prevention > DLP policy templates >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-09-26*

A custom data loss prevention (DLP) policy allows you to establish conditions, rules, and actions that can help meet the specific needs of your organization, and which may not be covered in one of the pre-existing DLP templates.

The rule conditions that are available to you in a single policy include all the traditional transport rules in addition to the sensitive information types presented in Sensitive information types inventory. For more information about transport rules in Exchange 2013, see Transport rules.

Caution:

You should enable your DLP policies in test mode before running them in your production environment. During such tests, it is recommended that you configure sample user mailboxes and send test messages that invoke your test policies in order to confirm the results.

For additional management tasks related to creating a custom DLP policy, see DLP procedures.

What do you need to know before you begin?

- Estimated time to complete: 60 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.
- In order to create a new custom DLP policy, you must follow the installation instructions for Exchange 2013. For more information about deployment, see Planning and deployment.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to create a custom DLP policy without any existing rules

1. In the EAC, navigate to **Compliance management** > **Data loss prevention**. Any existing policies

that you have configured are shown in a list.

2. Click the arrow that is beside the **Add +** icon, and select **New custom policy**.

Important:
If you click **Add +** icon instead of the arrow, you will create a new policy based on a template. For more information about using templates, see Create a DLP policy from a template.

3. On the **New custom policy** page, complete the following fields:
   a. **Name** Add a name that will distinguish this policy from others.
   b. **Description** Add an optional description that summarizes this policy.
   c. **Choose a state** Select the mode or state for this policy. The new policy is not fully enabled until you specify that it should be. The default mode for a policy is test without notifications.
4. Click **Save** to finish creating the new policy reference information. The policy is added to the list of all policies that you have configured, although there are not yet any rules or actions associated with this new custom policy.
5. Double-click the policy that you just created or select it and click **Edit** .
6. On the **Edit DLP policy** page, click **Rules**.

Click **Add +** to add a new blank rule. You can establish conditions using all the traditional transport rules in addition to the sensitive information types.

In order to avoid confusion, supply a unique name for each part of your policy or rule when you have the option to provide your own character string. There are several options additional options available to you:

   a. Click the arrow that is beside the **Add +** icon to add a rule about sender notification or allowing overrides.
   b. To remove a rule, highlight the rule and click **Delete** .
   c. Click **More options** ••• to add additional conditions and actions for this rule including time-bound limits of enforcement or effects on other rules in this policy.
7. Click **Save** to finish modifying the policy and save your changes.

DLP policy templates are one type of feature Microsoft Exchange that can help you design and apply a robust policy and compliance system for your messaging environment. For more information about compliance features, see Messaging policy and compliance.

For more information

Data loss prevention

Transport rules

Integrating sensitive information rules with transport rules

Define your own DLP templates and

information types

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-23*

You can develop data loss prevention (DLP) policy templates as XML files independent of Microsoft Exchange Server 2013 and then import them using the Exchange Administration Center or the Exchange management shell. This section describes the process and details of authoring and tuning DLP XML files for use within a DLP solution. You are not required to develop your own DLP XML files because the Exchange Administration Center provides a way for you to get started quickly with existing DLP policy templates and transport rules in order to scan messages.

Looking for management tasks related to DLP policy templates? See DLP procedures.

Note:

Exchange 2013: DLP is a premium feature that requires an Exchange Enterprise Client Access License (CAL). For more information about CALs and server licensing, see Exchange Server Licensing.
Exchange Online: DLP is a premium feature that requires an Exchange Online Plan 2 subscription. For more information, see Exchange Online Licensing.

Important:

It's beyond the scope of this documentation to recommend a business model or information about file packaging or deployment guidelines for the sensitive information rules or to discuss how such rules would be distributed. Furthermore, this documentation does not discuss protection mechanisms, such as encryption, for custom developed rules, nor does it discuss how such mechanism would be employed.

Extend the information types to meet your needs

The following sections describe concepts and the XML schema definition that you must understand in order to begin creating your own XML files for both DLP policy templates and sensitive information rule packages that can be imported into Exchange 2013 and used as DLP policies.

DLP in Microsoft Exchange helps you to apply organizational-specific policy to sensitive information. A key factor in the strength of a DLP solution is the ability to correctly identify confidential or sensitive information that may be unique to the organization, regulatory needs, geography, or other aspects of business. Although Microsoft has provided policy templates and sensitive information types within the product for you to get started, your unique business needs can require a custom data loss prevention solution. For this reason, Microsoft provides a way for you to create and import your own DLP policy templates or your own sensitive information definitions within classification rule packages. An accurate DLP solution relies on configuring the correct set of rules for the sensitive information detection engine that provide high degree of

protection while minimizing false positives and negatives.

Develop your own DLP policy templates

You can write your own DLP policy template XML file and import it. This approach to extending the DLP solution provided in Exchange will allow you to build DLP policies that closely match your DLP requirements.

Managing custom templates and their related policies is similar to managing the DLP policies that you create based on Microsoft-supplied templates. In a typical DLP policy lifecycle, you would do the following:

1. Create your own DLP policy template, a custom XML file. To learn more, see Developing DLP policy template files.
2. Import your custom template. To learn more, see Import a DLP policy from a file.
3. Create a DLP Policy based on your custom template. To learn more, see Create a DLP policy from a template.
4. Update your custom template by repeating steps 1 and 2.
5. Remove your custom template. To learn more, see Remove-DlpPolicyTemplate.

For more information about the XML schema definition and concepts related to developing your own templates, see Developing DLP policy template files.

Develop your own sensitive information types and matching logic in classification rule packages

You can write your own sensitive information definitions in a classification rule package, which is an XML file, and import it as part of your DLP solution. The sensitive information detection engine provides the deep content analysis capabilities for identifying sensitive information like credit card numbers, social security numbers, and company intellectual property. The engine is controlled by a configurable set of instructions, or rules, for scanning and analyzing the content. The rules are combined together into a classification rule package, an XML document that adheres to a standardized rules package XML schema definition. Here's how you can develop your own.

1. Create your own sensitive information types, a custom XML file. To learn more, see Developing sensitive information rule packages.
2. Import your sensitive information type. To learn more, see New-ClassificationRuleCollection.
3. Create custom template based on your information types. To learn more, see Developing sensitive information rule packages
4. Update your custom template by repeating steps 1 and 2.
5. Remove your custom template. To learn more, see Remove-ClassificationRuleCollection

For more information about the rule packages, see Developing sensitive information rule packages and Matching methods and techniques for rule packages.

Understanding rule types in rule packages

The rules within a rule package configure the process for detecting well-defined content characteristics; for example, rules for finding a driver's license number. Two main rule types are available: Entity and Affinity.

Entity rules are targeted toward well-defined (and oftentimes regulated) identifiers such as U.S. social security numbers. An Entity is represented by a collection of countable patterns. A pattern defines a collection of matches with an explicit primary match identifier. An example Entity is a driver's license.

Affinity rules are targeted toward a certain type of document such as a corporate financial statement. An Affinity is represented as a collection of independent evidences. Evidence is an aggregation of required matches within certain proximity. An example of an Affinity is the U.S. Sarbanes-Oxley Act.

For more information

Data loss prevention

Import a DLP policy from a file

New-ClassificationRuleCollection

Transport rules

Sensitive information types inventory

Developing DLP policy template files

Data loss prevention > DLP policy templates > Define your own DLP templates and information types >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-16*

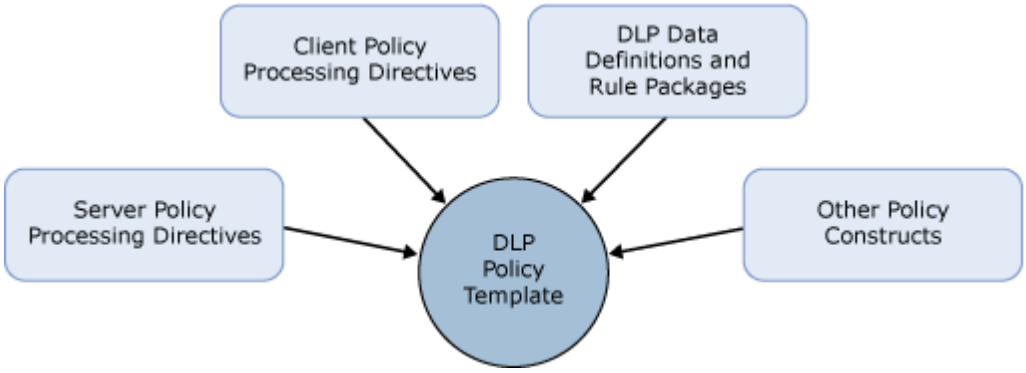
This overview explains the components of an XML schema definition for data loss prevention (DLP) policy template files and also provides an XML-format example policy file. It will be helpful to understand the overall DLP architecture and rule-development process before you begin. For more information, see Data loss prevention and Define your own DLP templates and information types.

In order to make data loss prevention solutions easy to adopt and manage, a conceptual model known as DLP policies and policy templates is introduced in Microsoft Exchange Server 2013. DLP policy templates provide a preliminary design for your intended DLP policy. In order to be valuable,

a DLP policy template must encapsulate all the directives and data objects that are required to meet a specific policy objective, such as a regulation or business need. The template is not environment-specific. It is simply a definition or model of a policy that can be provided as part of the product configuration or supplied by independent software vendors and partners. DLP policies on the other hand, are run-time instantiations of the templates that are specific to the deployment environment. Your existing messaging policy framework can incorporate DLP policies through the use of transport rules. Transport rules provide great flexibility in adapting and expressing the richness of your DLP solutions.

Policy template sources and structure

DLP policy templates are typically influenced from multiple sources such as server-based processing directives, client computer policies, or other policy constructs as shown in the following image:



Simple management operations are available for DLP policy templates though both the Exchange Management Shell and Internet-based interfaces, such as the Exchange Administration Center, which include Import, Export, Deletion and Query capabilities. A DLP policy is created by referencing a DLP policy template as part of the creation process. These referenced DLP policy templates may be references to ones installed in the system, which are stored in active directory domain services, or be provided as input directly from externally supplied policies.

DLP policy templates are represented as XML documents. A single XML schema is used for policies provided within Exchange and externally also. The conceptual structure of the XML document is represented in the table below, which shows the major elements. The set of policy component definitions help you achieve a specific policy objective such as a regulation or business need.

Structural Element	Meaning or Example
Publisher	Microsoft or Partner
Version	15.0.1.0
Policy Name	PCI-DSS
Description	The PCI-DSS DLP policy helps detect the presence

	of information subject to PCI Data Security Standard (PCI DSS), including information like credit card or debit card numbers. Use of this policy does not ensure compliance with any regulation. After your testing is complete, make the necessary configuration changes in Exchange so the transmission of information complies with your organization's policies. Examples include configuring TLS with known business partners or adding more restrictive transport rule actions, such as adding rights protection to messages that contain this type of data.
Metadata	Tags to describe the local regulation, country or region, keywords and more.
Set of policy constructs	1. Transport rule definitions, such as conditions and actions. 2. Email client behavior definitions that control client experience through interactive notifications. 3. Optionally, configuration references that need to be coordinated with client environment-specific settings.
Set of data classifications	1. Specifies classification entities or affinities. 2. Entities have count and confidence; affinities only have a confidence. 3. Comes with its own set of properties and classification schema.

Policy template format definition

DLP Policy templates are expressed as XML documents which adhere to the following schema. Note that the XML is case-sensitive. For instance, `dlpPolicyTemplates` will work, but `DlpPolicyTemplates` won't work.

```
<?xml version="1.0" encoding="UTF-8"?>
<dlpPolicyTemplates>
  <dlpPolicyTemplate id="F7C29AEC-A52D-4502-9670-
141424A83FAB" mode="Audit" state="Enabled"
version="15.0.2.0">
    <contentVersion>4</contentVersion>
    <publisherName>Microsoft</publisherName>
    <name>
      <localizedString lang="en">PCI-DSS</localizedString>
    </name>
    <description>
      <localizedString lang="en">Detects the presence of
information subject to Payment Card Industry Data Security
Standard (PCI-DSS) compliance requirements.</
localizedString>
    </description>
    <keywords></keywords>
    <ruleParameters></ruleParameters>
    <ruleParameters/>
    <policyCommands>
      <!-- The contents below are applied/executed as rules
directly in PS - -->
      <commandBlock>
        <![CDATA[ new-transportRule "PCI-DSS: Monitor
Payment Card Information Sent To Outside" -DlpPolicy "%%
DlpPolicyName%%" -SentToScope NotInOrganization -
SetAuditSeverity High -MessageContainsDataClassifications
@{Name="Credit Card Number"; MinCount="1" } -Comments
"Monitors payment card information sent to outside the
organization as part of the PCI-DSS DLP Policy."]]>
      </commandBlock>
      <commandBlock>
        <![CDATA[ new-transportRule "PCI-DSS: Monitor
Payment Card Information Sent To Within" -DlpPolicy "%%
DlpPolicyName%%" -Comments "Monitors payment card
information sent inside the organization as part of the
PCI-DSS DLP Policy." -SentToScope InOrganization -
SetAuditSeverity Low -MessageContainsDataClassifications
@{Name="Credit Card Number"; MinCount="1" } ]]>
```

```
        </commandBlock>
      </policyCommands>
      <policyCommandsResources></policyCommandsResources>
    </dlpPolicyTemplate>
</dlpPolicyTemplates>
```

If a parameter you include in your XML file for any element includes a space, the parameter has to be surrounded by double quotes or it will not work properly. In the example below, the parameter that follows `-SentToScope` is acceptable and does not include double quotes because it is one continuous string of characters without a space. However, the parameter provided for `–Comments` will not appear in the Exchange Administration Center because there are no double quotes and it includes spaces.

```
<CommandBlock><![CDATA[ new-transportRule "PCI-DSS: Monitor Payment Card Information Sent To Within" -DlpPolicy "PCI-DSS" -Comments Monitors payment card information sent inside the organization -SentToScope InOrganization -SetAuditSeverity Low -MessageContainsDataClassifications @{Name="Credit Card Number"; MinCount="1" } ]]> </CommandBlock>
```

localizedString Element

The template format offers the capability to localize strings in the template which may be presented to the end-user, for example as part of selecting which DLP policy templates are installed. The localizedString element can be used to supply multiple definitions for the Name and Description fields.

ruleParameters Node

This is an optional set of parameters that need to be supplied during the template instantiation phase when creating a DLP policy to map to deployment specific objects. For example an actual distribution group that is available in the deployment.

dlpPolicyTemplate Element

This is the root element for the DLP policy template and is required for every template. Available attributes are shown in the following table:

Attribute Name	Required?	Description
Version	Yes	The version number used in

		this DLP policy template document.
State	No	Optional default configuration for the state of the policy.
Mode	No	Optional default configuration for the mode of the policy.
Id	No	A GUID to uniquely identify this DLP policy template definition in the following format:"A29C69BF-4F98-47F1-9A99-5771DFD2C27F".

Child elements include the following sequence of elements.

Child Element	(minimum, maximum)	Description
PublisherName	(1, 1)	Meta data for the template's publisher
Name	(1, 1)	Localizable name for this template.
Description	(1, 1)	Localizable description for this template.
Keywords	(1, 1)	List of keywords that applies to this template. A template may have an empty list of keywords.
RuleParameters	(0, 1)	List of template parameters that are used in the policy definition.
PolicyCommands	(0, 1)	List of Transport rule definitions for this policy. This is an optional element.

DLP Policy Template: PolicyCommands

This part of the policy template contains the list of the Exchange Management Shell commands that are used to instantiate the policy definition. The import process will execute each of the commands as part of the instantiation process. Sample policy commands are provided here.

```
<PolicyCommands>
    <!-- The contents below are applied/executed as rules
directly in PS - -->
      <CommandBlock> <![CDATA[ new-transportRule "PCI-DSS:
Monitor Payment Card Information Sent To Outside" -
DlpPolicy "PCI-DSS" -SentToScope NotInOrganization -
SetAuditSeverity High -MessageContainsDataClassifications
@{Name="Credit Card Number"; MinCount="1" } -Comments
"Monitors payment card information sent to outside the
organization as part of the PCI-DSS DLP policy."]]></
CommandBlock>
      <CommandBlock><![CDATA[ new-transportRule "PCI-DSS:
Monitor Payment Card Information Sent To Within" -DlpPolicy
"PCI-DSS" -Comments "Monitors payment card information sent
inside the organization as part of the PCI-DSS DLP policy."
-SentToScope InOrganization -SetAuditSeverity Low -
MessageContainsDataClassifications @{Name="Credit Card
Number"; MinCount="1" } ]]> </CommandBlock>
  </PolicyCommands>
```

The format of the cmdlets is the standard Exchange Management Shell cmdlet syntax for the cmdlets used. The commands are executed in sequence. It is possible for each of the command nodes to contain a script block which would be composed of multiple commands. Below is example that illustrates how to embed classification rule pack inside of a dlp policy template, and installing the rule pack as part of the policy creation process. The classification rule pack is embedded in the policy template, and then passed as a parameter to the cmdlet in the template:

```
<CommandBlock>
  <![CDATA[
$rulePack = [system.Text.Encoding]::Unicode.GetBytes('<?xml
version="1.0" encoding="utf-16"?>
<rulePackage xmlns="http://schemas.microsoft.com/
office/2011/mce">
  <RulePack id="c3f021a3-c265-4dc2-b3a7-41a1800bf518">
```

```
    <Version major="1" minor="0" build="0" revision="0"/>
    <Publisher id="e17451d3-9648-4117-a0b1-493a6d5c73ad"/>
    <Details defaultLangCode="en-us">
      <LocalizedDetails langcode="en-us">
        <PublisherName>Contoso</PublisherName>
        <Name>Contoso Sample Rule Pack</Name>
        <Description>This is a sample rule package</
Description>
      </LocalizedDetails>
    </Details>
  </RulePack>
  <Rules>
    <Entity id="7cc35258-6b35-4415-baff-a76d1a018980"
patternsProximity="300" recommendedConfidence="85"
workload="Exchange">
      <Pattern confidenceLevel="85">
        <IdMatch idRef="Regex_Contoso" />
        <Any minMatches="1">
          <Match idRef="Regex_conf" />
        </Any>
      </Pattern>
    </Entity>
    <Regex id="Regex_Contoso">(?i)(\bContoso\b)</Regex>
    <Regex id="Regex_conf">(?i)(\bConfidential\b)</Regex>
    <LocalizedStrings>
      <Resource idRef="7cc35258-6b35-4415-baff-
a76d1a018980">
        <Name default="true" langcode="en-us">
          Confidential Information Rule
        </Name>
        <Description default="true" langcode="en-us">
          Sample rule pack - Detects Contoso confidential
information
        </Description>
      </Resource>
    </LocalizedStrings>
  </Rules>
</RulePackage>
')
```

```
New-ClassificationRuleCollection -FileData $rulePack
New-TransportRule -name "customEntity" -DlpPolicy "%%
DlpPolicyName%%" -SentToScope NotInOrganization -
MessageContainsDataClassifications @{Name="Confidential
Information Rule"} -SetAuditSeverity High]]>
</CommandBlock>
```

Child elements include the following ordered sequence of elements.

Child Element	(Minimum, Maximum)	Description
CommandBlock	(1, n)	A command block that is executed in the PowerShell. The command blocks are each executed in sequence.

For more information

Data loss prevention

Define your own DLP templates and information types

Import a DLP policy from a file

Developing sensitive information rule packages

Data loss prevention > DLP policy templates > Define your own DLP templates and information types >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-19*

The XML schema and guidance in this topic will help you get started creating your own basic data loss prevention (DLP) XML files that define your own sensitive information types in a classification rule package. After you have created a well-formed XML file, you can import it by using either the Exchange Administration Center or the Exchange management shell in order to help create a Microsoft Exchange Server 2013 DLP solution. An XML file that is a custom DLP policy template can contain the XML that is your classification rule package. For an overview about defining your own DLP templates as XML files, see Define your own DLP templates and information types.

Contents

Overview of the rule authoring process

The rule authoring process is made up of the following general steps.

1. Prepare a set of test documents representative of their target environment. Key characteristics to consider for the set of test documents include: A subset of the documents contain the entity or affinity for which the rule is being authored, and a subset of the documents do not contain the entity or affinity for which the rule is being authored.
2. Identify the rules that meet acceptance requirements (precision and recall) to identify qualifying content. This may require the development of multiple conditions within a rule, bound with Boolean logic, which together satisfy the minimum match requirements to identify target documents.
3. Establish the recommended confidence level for the rules based on the acceptance requirements (precision and recall). The recommended confidence element can be thought of as the default confidence level for the rule.
4. Validate the rules by instantiating a policy with them and monitoring the sample test content. Based on the results, adjust the rules, or the confidence level to maximize the detected content while minimizing false positives and negatives. Continue the cycle of validation and rule adjustments until a satisfactory level of content detection is reached for both positive and negative samples.

For information about the XML schema definition for policy template files, see Developing DLP policy template files.

Rule description

Two main Rule types can be authored for the DLP sensitive information detection engine: Entity and Affinity. The Rule type chosen is based on the type of processing logic that should be applied to the processing of the content as described in the previous sections. The rule definitions are configured in an XML document in the format described by the standardized Rules XSD. The rules describe both the type of content to match and the confidence level that the described match represents the target content. Confidence level specifies the probability for the Entity to be present if a Pattern is found in the content or the probability for the Affinity to be present if Evidence is found in the content.

Basic rule structure

The Rule definition is constructed from three main components:

1. **Entity** defines the matching and counting logic for that rule
2. **Affinity** defines the matching logic for the rule
3. **Localized Strings** localization for rule names and their descriptions

Three additional supporting elements are used that define the details of the processing and referenced within the main components: Keyword, Regex, and Function. By using references, a single definition of the supporting elements, like a social security number, can be used to in multiple Entity or Affinity rules. The basic rule structure in XML format can be seen as follows.

```
<?xml version="1.0" encoding="utf-16"?>
<RulePackage xmlns="http://schemas.microsoft.com/
office/2011/mce">
  <RulePack id="DAD86A92-AB18-43BB-AB35-96F7C594ADAA">
    <Version major="1" minor="0" build="0" revision="0"/>
    <Publisher id="619DD8C3-7B80-4998-A312-4DF0402BAC04"/>
    <Details defaultLangCode="en-us">
      <LocalizedDetails langcode="en-us">
        <PublisherName>DLP by EPG</PublisherName>
        <Name>CSO Custom Rule Pack</Name>
        <Description>This is a rule package for a EPG
demo.</Description>
      </LocalizedDetails>
    </Details>
  </RulePack>
  <Rules>
    <!-- Employee ID -->
    <Entity id="E1CC861E-3FE9-4A58-82DF-4BD259EAB378"
patternsProximity="300" recommendedConfidence="75">
      <Pattern confidenceLevel="75">
        <IdMatch idRef="Regex_employee_id" />
        <Match idRef="Keyword_employee" />
      </Pattern>
    </Entity>
    <Regex id="Regex_employee_id">(\s)(\d{9})(\s)</Regex>
    <Keyword id="Keyword_employee">
      <Group matchStyle="word">
        <Term>Identification</Term>
```

```
        <Term>Contoso Employee</Term>
      </Group>
    </Keyword>
    <LocalizedStrings>
      <Resource idRef="E1CC861E-3FE9-4A58-82DF-
4BD259EAB378">
        <Name default="true" langcode="en-us">
          Employee ID
        </Name>
        <Description default="true" langcode="en-us">
          A custom classification for detecting Employee
ID's
        </Description>
      </Resource>
    </LocalizedStrings>
  </Rules>
</RulePackage>
```

Entity rules

Entity Rules are targeted towards well defined identifiers, such as Social Security Number, and are represented by a collection of countable patterns. Entity Rules returns a count and the confidence level of a match, where Count is the total number of instances of the entity that were found and the Confidence Level is the probability that the given entity exists in the given document. Entity contains the "id" attribute as its unique identifier. The identifier is used for localization, versioning, and querying. The Entity id must be a GUID and should not be duplicated in other entities or affinities. It is referenced in the localized strings section.

Entity rules contains optional patternsProximity attribute (default = 300) which is used when applying Boolean logic to specify the adjacency of multiple patterns required to satisfy the match condition. Entity element contains 1 or more child Pattern elements, where each pattern is a distinct representation of the Entity like Credit Card Entity or Driver's License Entity. The Pattern element has a required attribute of confidenceLevel which represents the pattern's precision based on sample dataset. Pattern element can have three child elements:

1. IdMatch - This is required.
2. Match
3. Any

If any of the Pattern elements return "true," the Pattern is satisfied. The count for the Entity element equals the sum of all detected Pattern counts.

$$\text{Entity Count} = \sum_{n=1}^{k} (Pattern\ n = true)$$

where k is the number of Pattern elements for the Entity.

A Pattern element must have exactly one IdMatch element. IdMatch represents the identifier that the Pattern is to match, for example a credit card number or ITIN number. The Count for a pattern is the number of IdMatches matched with the Pattern element. IdMatch element anchors the proximity window for the Match elements.

Another optional sub-element of the Pattern element is the Match element which represents corroborative evidence that is required to be matched to support finding the IdMatch element. For example, the higher confidence rule may require that, in addition to finding a credit card number, additional artifacts exist in the document, within a proximity window of the credit card, like address and name. These additional artifacts would be represented through the Match element or Any element (these are described in detail in Matching Methods and Techniques section). Multiple Match elements can be included in a Pattern definition which can be included directly in the Pattern element or combined using the Any element to define matching semantics. It returns true if a match is found in the proximity window anchored around the IdMatch content. An optional attribute of minOccurs can be used to specify the minimum number of occurrences that need to be satisfied for a successful match.

Both the IdMatch and Match elements do not define the details of what content needs to be matched but instead reference it through the idRef attribute. This promotes reusability of definitions in multiple Pattern constructs.

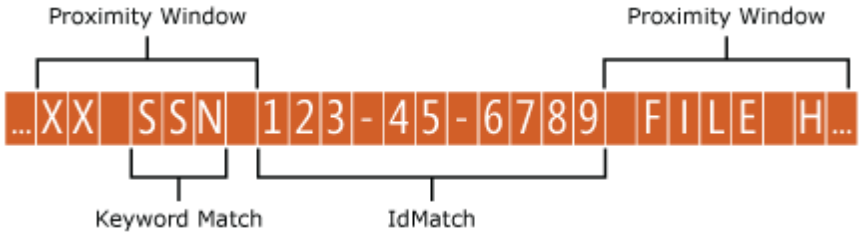
```
<Entity id="..." patternsProximity="300" >
    <Pattern confidenceLevel="85">
        <IdMatch idRef="FormattedSSN" />
        <Any minMatches="1">
            <Match idRef="SSNKeyword" />
            <Match idRef="USDate" />
            <Match idRef="USAddress" />
            <Match idRef="Name" />
        </Any>
    </Pattern>
    <Pattern confidenceLevel="65">
        <IdMatch idRef="UnformattedSSN" />
        <Match idRef="SSNKeyword" />
        <Any minMatches="1">
            <Match idRef="USDate" />
            <Match idRef="USAddress" />
            <Match idRef="Name" />
        </Any>
    </Pattern>
```

```
</Entity>
```

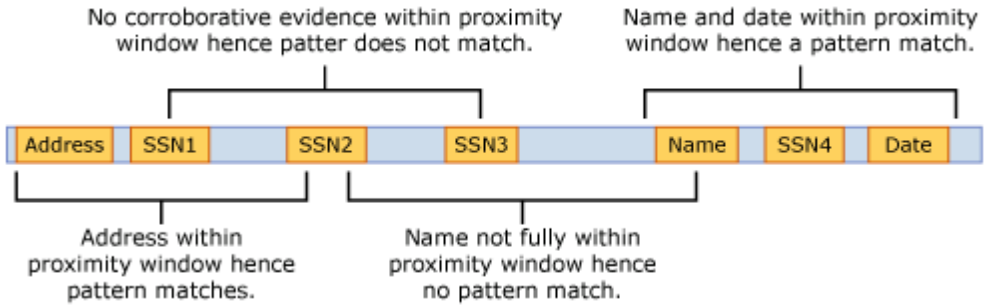
The Entity id element, represented in the previous XML by "…" should be a GUID and it is referenced in the Localized Strings section.

Entity pattern proximity window

Entity holds optional patternsProximity attribute value (integer, default = 300) used to find the Patterns. For each pattern the attribute value defines the distance (in Unicode characters) from the IdMatch location for all other Matches specified for that Pattern. The proximity window is anchored by the IdMatch location, with the window extending to the left and right of the IdMatch.



The example below illustrates how the proximity window affects the matching algorithm where the SSN IdMatch element requires at least 1 of address, name or date corroborating matches. Only SSN1 and SSN4 match because for SSN2 and SSN3, either no or only partial corroborating evidence is found within the proximity window.



Entity confidence level

Entity element's confidence level is the combination of all the satisfied Pattern's confidence levels. They are combined using the following equation:

$$Entity\ Confidence\ Level = 1 - \prod_{i=1}^{k}(1 - Confidence\ Level\ (Pattern\ i))$$

where k is the number of Pattern elements for the Entity and a Pattern that does not match returns a confidence level of 0.

Referring back to the example entity element structure code sample, if both patterns are matched, the entity's confidence level is 94.75% based on the following calculation:

$CL_{Entity} = 1-[(1-CL_{Pattern1}) \times (1-CL_{Pattern1})]$

$= 1-[(1-0.85) \times (1-0.65)]$

$= 1-(0.15 \times 0.35)$

$= 94.75\%$

Similarly, if only the second pattern matches, the Entity's confidence level is 65% based on the following calculation:

$CL_{Entity} = 1 - [(1 - CL_{Pattern1}) \text{ X } (1 - CL_{Pattern1})]$

$= 1 - [(1 - 0) \text{ X } (1 - 0.65)]$

$= 1 - (1 \text{ X } 0.35)$

$= 65\%$

These confidence values are assigned in the rules for individual patterns based on the set of test documents validated as part of the rule authoring process.

Affinity rules

Affinity rules are targeted towards content without well-defined identifiers, for example Sarbanes-Oxley or corporate financial content. For this content no single consistent identifier can be found and instead the analysis requires determining if a collection of evidence is present. Affinity rules do not return a count, instead they return if found and the associated confidence level. Affinity content is represented as a collection of independent evidences. Evidence is an aggregation of required matches within certain proximity. For Affinity rule, the proximity is defined by the evidencesProximity attribute (default is 600) and the minimum confidence level by the thresholdConfidenceLevel attribute.

Affinity rules contains the id attribute for its unique identifier that is used for localization, versioning and querying. Unlike Entity rules, because Affinity rules do not rely on well-defined identifiers, they do not contain the IdMatch element.

Each Affinity rule contains one or more child Evidence elements which define the evidence that is to be found and the level of confidence contributing to the Affinity rule. The affinity is not considered found if the resulting confidence level is below the threshold level. Each Evidence logically represents corroborative evidence for this "type" of document and the confidenceLevel attribute is the precision for that Evidence on the test dataset.

Evidence elements have one or more of Match or Any child elements. If all child Match and Any elements match, the Evidence is found and the confidence level is contributed to the rules confidence level calculation. The same description applies to the Match or Any elements for Affinity rules as for Entity rules.

```
<Affinity id="..."
          evidencesProximity="1000"
          thresholdConfidenceLevel="65">
    <Evidence confidenceLevel="40">
```

```
            <Any>
                <Match idRef="AssetsTerms" />
                <Match idRef="BalanceSheetTerms" />
                <Match idRef="ProfitAndLossTerms" />
            </Any>
        </Evidence>
        <Evidence confidenceLevel="40">
            <Any minMatches="2">
                <Match idRef="TaxTerms" />
                <Match idRef="DollarAmountTerms" />
                <Match idRef="SECTerms" />
                <Match idRef="SECFilingFormTerms" />
                <Match idRef="DollarTotalRegex" />
            </Any>
        </Evidence>
</Affinity>
```

Affinity proximity window

The proximity window for Affinity is calculated differently than for Entity patterns. Affinity proximity follows a sliding window model. The affinity proximity algorithm attempts to find the maximum number of matching evidences in the given window. Evidences in the proximity window must have a confidence level greater than the threshold defined for the Affinity rule to be found.

Affinity confidence level

Confidence level for the Affinity equals the combination of found Evidences within the proximity window for the Affinity rule. While similar to the confidence level of Entity rule, the key difference is the application of proximity window. Similar to the Entity rules, Affinity element's confidence level is the combination of all the satisfied Evidence confidence levels, but for Affinity rule it only represents the highest combination of Evidence elements found within the proximity window. The Evidence confidence levels are combined using the following equation:

$$Affinity\ Confidence\ Level = 1 - \prod_{i=1}^{k} (1 - Confidence\ Level\ (Evidence\ i))$$

where k is the number of Evidence elements for the Affinity matched within the proximity window.

Referring back to Figure 4 Example Affinity rule structure, if all three evidences are matched within the proximity sliding window, the affinity confidence level is 85.6% based on the calculation below.

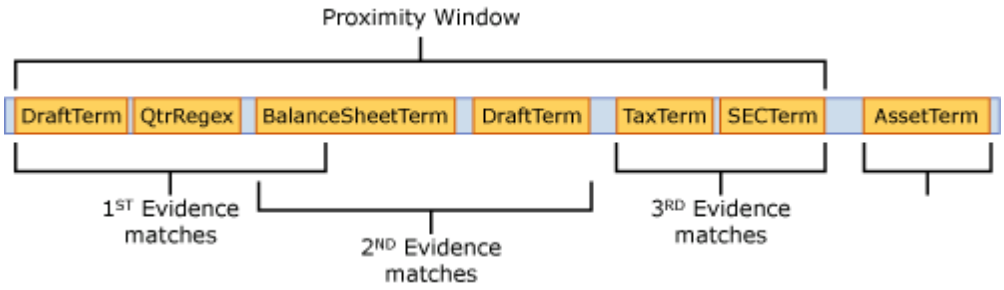
This exceeds the Affinity rule threshold of 65 which results in the rule matching.

$CL_{Affinity} = 1 - [(1 - CL_{Evidence\ 1}) X (1 - CL_{Evidence\ 2}) X (1 - CL_{Evidence\ 2})]$

$= 1 - [(1 - 0.6) X (1 - 0.4) X (1 - 0.4)]$

$= 1 - (0.4 X 0.6 X 0.6)$

$= 85.6\%$



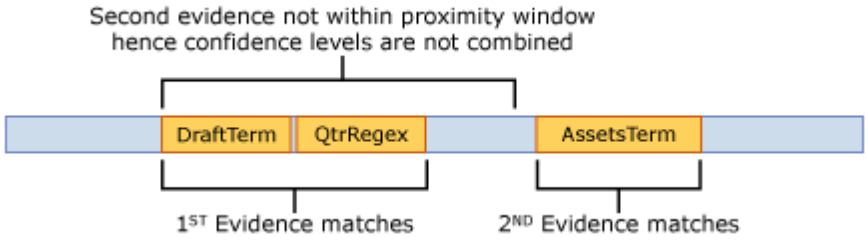
Using the same example rule definition, if only the first evidence matches because the second Evidence is outside of the proximity window, the highest Affinity confidence level is 60% based on the calculation below and the Affinity rule does not match since the threshold of 65 was not met.

$CL_{Affinity} = 1 - [(1 - CL_{Evidence\ 1}) X (1 - CL_{Evidence\ 2}) X (1 - CL_{Evidence\ 2})]$

$= 1 - [(1 - 0.6) X (1 - 0) X (1 - 0)]$

$= 1 - (0.4 X 1 X 1)$

$= 60\%$



Tuning confidence levels

One of the key aspects of the rule authoring process is the tuning of confidence levels for both Entity and Affinity rules. After creating the rule definitions, run the rule against the representative content and collect the accuracy data. Compare the returned results for each pattern or evidence against the expected results for the test documents.

Test Content	P_1 or E_1	P_2 or E_2	P_n or E_n	Returned Result	Expected Result
Content 1	+	-	-	+	+
Content 2	-	-	+	+	-
Content n	-	-	-	-	+

If the rules meet acceptance requirements, that is, the Pattern or Evidence has a confidence rate above an established threshold (e.g. 75%)—the match expression is complete and it can be moved to the next step.

If the Pattern or Evidence do not meet the confidence level, then re-author it (e.g. add more corroborative evidence; remove or add additional Patterns/Evidences; etc.) and repeat this step.

Next, tune the confidence level for each Pattern or Evidence in your rules based on the results from the previous step. For each Pattern or Evidence, aggregate the number of True Positives (TP), subset of the documents that contain the entity or affinity for which the rule is being authored and that resulted in a match and the number of False Positives (FP), a subset of documents that do not contain the entity or affinity for which the rule is being authored and that also returned a match. Set confidence level for each Pattern/Evidence using the following calculation:

Confidence Level = True Positives / (True Positives + False Positives)

Pattern or Evidence	True Positives	False Positives	Confidence Level
P_1or E_1	4	1	80%
P_2or E_2	2	2	50%
$P_n$or E_n	9	10	47%

Using local languages in your XML file

The rule schema supports storing of localized name and description for each of Entity and Affinity elements. Each Entity and Affinity element must contain a corresponding element in the LocalizedStrings section. To localize each element, include a Resource element as a child of the LocalizedStrings element to store name and descriptions for multiple locales for each element. The Resource element includes a required idRef attribute which matches the corresponding idRef attribute for each element that is being localized. The Locale child elements of the Resource element contains the localized name and descriptions for each specified locale.

```
<LocalizedStrings>
    <Resource idRef="guid">
```

```
        <Locale langcode="en-US" default="true">
            <Name>affinity name en-us</Name>
            <Description>
                affinity description en-us
            </Description>
        </Locale>
        <Locale langcode="de">
            <Name>affinity name de</Name>
            <Description>
                affinity description de
            </Description>
        </Locale>
    </Resource>
</LocalizedStrings>
```

Classification rule pack XML schema definition

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:mce="http://schemas.microsoft.com/
office/2011/mce"
           targetNamespace="http://schemas.microsoft.com/
office/2011/mce"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           elementFormDefault="qualified"
           attributeFormDefault="unqualified"
           id="RulePackageSchema">
  <xs:simpleType name="LangType">
    <xs:union memberTypes="xs:language">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value=""/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>
  <xs:simpleType name="GuidType" final="#all">
    <xs:restriction base="xs:token">
      <xs:pattern value="[0-9a-fA-F]{8}\-([0-9a-fA-F]{4}\-)
```

```
{3}[0-9a-fA-F]{12}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="RulePackageType">
    <xs:sequence>
      <xs:element name="RulePack" type="mce:RulePackType"/>
      <xs:element name="Rules" type="mce:RulesType">
        <xs:key name="UniqueRuleId">
          <xs:selector xpath="mce:Entity|mce:Affinity"/>
          <xs:field xpath="@id"/>
        </xs:key>
        <xs:key name="UniqueProcessorId">
          <xs:selector xpath="mce:Regex|mce:Keyword"></
xs:selector>
          <xs:field xpath="@id"/>
        </xs:key>
        <xs:key name="UniqueResourceIdRef">
          <xs:selector xpath="mce:LocalizedStrings/
mce:Resource"/>
          <xs:field xpath="@idRef"/>
        </xs:key>
        <xs:keyref name="ReferencedRuleMustExist"
refer="mce:UniqueRuleId">
          <xs:selector xpath="mce:LocalizedStrings/
mce:Resource"/>
          <xs:field xpath="@idRef"/>
        </xs:keyref>
        <xs:keyref name="RuleMustHaveResource"
refer="mce:UniqueResourceIdRef">
          <xs:selector xpath="mce:Entity|mce:Affinity"/>
          <xs:field xpath="@id"/>
        </xs:keyref>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RulePackType">
    <xs:sequence>
      <xs:element name="Version" type="mce:VersionType"/>
      <xs:element name="Publisher"
```

```
type="mce:PublisherType"/>
      <xs:element name="Details" type="mce:DetailsType">
        <xs:key name="UniqueLangCodeInLocalizedDetails">
          <xs:selector xpath="mce:LocalizedDetails"/>
          <xs:field xpath="@langcode"/>
        </xs:key>
        <xs:keyref name="DefaultLangCodeMustExist"
refer="mce:UniqueLangCodeInLocalizedDetails">
          <xs:selector xpath="."/>
          <xs:field xpath="@defaultLangCode"/>
        </xs:keyref>
      </xs:element>
      <xs:element name="Encryption"
type="mce:EncryptionType" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="id" type="mce:GuidType"
use="required"/>
  </xs:complexType>
  <xs:complexType name="VersionType">
    <xs:attribute name="major" type="xs:unsignedShort"
use="required"/>
    <xs:attribute name="minor" type="xs:unsignedShort"
use="required"/>
    <xs:attribute name="build" type="xs:unsignedShort"
use="required"/>
    <xs:attribute name="revision" type="xs:unsignedShort"
use="required"/>
  </xs:complexType>
  <xs:complexType name="PublisherType">
    <xs:attribute name="id" type="mce:GuidType"
use="required"/>
  </xs:complexType>
  <xs:complexType name="LocalizedDetailsType">
    <xs:sequence>
      <xs:element name="PublisherName" type="mce:NameType"/
>
      <xs:element name="Name" type="mce:RulePackNameType"/>
      <xs:element name="Description"
type="mce:OptionalNameType"/>
```

```
    </xs:sequence>
    <xs:attribute name="langcode" type="mce:LangType"
use="required"/>
  </xs:complexType>
  <xs:complexType name="DetailsType">
    <xs:sequence>
      <xs:element name="LocalizedDetails"
type="mce:LocalizedDetailsType" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="defaultLangCode"
type="mce:LangType" use="required"/>
  </xs:complexType>
  <xs:complexType name="EncryptionType">
    <xs:sequence>
      <xs:element name="Key" type="xs:normalizedString"/>
      <xs:element name="IV" type="xs:normalizedString"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="RulePackNameType">
    <xs:restriction base="xs:token">
      <xs:minLength value="1"/>
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="NameType">
    <xs:restriction base="xs:normalizedString">
      <xs:minLength value="1"/>
      <xs:maxLength value="256"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="OptionalNameType">
    <xs:restriction base="xs:normalizedString">
      <xs:minLength value="0"/>
      <xs:maxLength value="256"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="RestrictedTermType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
```

```
      <xs:maxLength value="512"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="RulesType">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Entity" type="mce:EntityType"/>
        <xs:element name="Affinity"
type="mce:AffinityType"/>
      </xs:choice>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Regex" type="mce:RegexType"/>
        <xs:element name="Keyword" type="mce:KeywordType"/>
      </xs:choice>
      <xs:element name="LocalizedStrings"
type="mce:LocalizedStringsType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EntityType">
    <xs:sequence>
      <xs:element name="Pattern" type="mce:PatternType"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="mce:GuidType"
use="required"/>
    <xs:attribute name="patternsProximity"
type="mce:ProximityType" use="required"/>
    <xs:attribute name="recommendedConfidence"
type="mce:ProbabilityType"/>
    <xs:attribute name="workload" type="mce:WorkloadType"/>
  </xs:complexType>
  <xs:complexType name="PatternType">
    <xs:sequence>
      <xs:element name="IdMatch" type="mce:IdMatchType"/>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Match" type="mce:MatchType"/>
        <xs:element name="Any" type="mce:AnyType"/>
      </xs:choice>
    </xs:sequence>
```

```
    <xs:attribute name="confidenceLevel"
type="mce:ProbabilityType" use="required"/>
  </xs:complexType>
  <xs:complexType name="AffinityType">
    <xs:sequence>
      <xs:element name="Evidence" type="mce:EvidenceType"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="mce:GuidType"
use="required"/>
    <xs:attribute name="evidencesProximity"
type="mce:ProximityType" use="required"/>
    <xs:attribute name="thresholdConfidenceLevel"
type="mce:ProbabilityType" use="required"/>
    <xs:attribute name="workload" type="mce:WorkloadType"/>
  </xs:complexType>
  <xs:complexType name="EvidenceType">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Match" type="mce:MatchType"/>
        <xs:element name="Any" type="mce:AnyType"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="confidenceLevel"
type="mce:ProbabilityType" use="required"/>
  </xs:complexType>
  <xs:complexType name="IdMatchType">
    <xs:attribute name="idRef" type="xs:string"
use="required"/>
  </xs:complexType>
  <xs:complexType name="MatchType">
    <xs:attribute name="idRef" type="xs:string"
use="required"/>
  </xs:complexType>
  <xs:complexType name="AnyType">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Match" type="mce:MatchType"/>
        <xs:element name="Any" type="mce:AnyType"/>
```

```
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="minMatches"
type="xs:nonNegativeInteger" default="1"/>
    <xs:attribute name="maxMatches"
type="xs:nonNegativeInteger" use="optional"/>
  </xs:complexType>
  <xs:simpleType name="ProximityType">
    <xs:restriction base="xs:positiveInteger">
      <xs:minInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ProbabilityType">
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="WorkloadType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Exchange"/>
      <xs:enumeration value="Outlook"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="RegexType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="id" type="xs:token"
use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="KeywordType">
    <xs:sequence>
      <xs:element name="Group" type="mce:GroupType"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:token" use="required"/
>
```

```
  </xs:complexType>
  <xs:complexType name="GroupType">
    <xs:sequence>
      <xs:choice>
        <xs:element name="Term" type="mce:TermType"
maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="matchStyle" default="word">
      <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="word"/>
          <xs:enumeration value="string"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="TermType">
    <xs:simpleContent>
      <xs:extension base="mce:RestrictedTermType">
        <xs:attribute name="caseSensitive"
type="xs:boolean" default="false"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="LocalizedStringsType">
    <xs:sequence>
      <xs:element name="Resource" type="mce:ResourceType"
maxOccurs="unbounded">
        <xs:key name="UniqueLangCodeUsedInNamePerResource">
          <xs:selector xpath="mce:Name"/>
          <xs:field xpath="@langcode"/>
        </xs:key>
        <xs:key
name="UniqueLangCodeUsedInDescriptionPerResource">
          <xs:selector xpath="mce:Description"/>
          <xs:field xpath="@langcode"/>
        </xs:key>
      </xs:element>
```

```
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ResourceType">
    <xs:sequence>
      <xs:element name="Name" type="mce:ResourceNameType"
maxOccurs="unbounded"/>
      <xs:element name="Description"
type="mce:DescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="idRef" type="mce:GuidType"
use="required"/>
  </xs:complexType>
  <xs:complexType name="ResourceNameType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="default" type="xs:boolean"
default="false"/>
        <xs:attribute name="langcode" type="mce:LangType"
use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="DescriptionType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="default" type="xs:boolean"
default="false"/>
        <xs:attribute name="langcode" type="mce:LangType"
use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```

For more information

Data loss prevention

Matching methods and techniques for rule packages

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-21*

This topic describes techniques for matching pattern and evidence elements within a data loss prevention (DLP) XML file that is designed to contain your own custom sensitive information type rule package. After you have created a well-formed XML file, you can import the file by using the Exchange admin center (EAC) or Exchange Management Shell in order to help create your Microsoft Exchange Server 2013 DLP solution. Before you can make use of the matching methods described here, you should already have a DLP XML file started. For more information about defining your own DLP templates and XML files, see Define your own DLP templates and information types.

The Match element

The `Match` element is used within the `Pattern` and `Evidence` elements to represents the underlying keyword, regex or function that is to be matched. The definition of the match itself is stored outside of the `Rule` element and is referenced through the `idRef` required attribute. Multiple `Match` elements can be included in a Pattern definition which can be included directly in the `Pattern` element or combined using the `Any` element to define matching semantics.

An optional attribute of `minoccurs` can be used to specify the minimum number of occurrences that need to be satisfied for a successful match of each of the `Match` elements.

```
<?xml version="1.0" encoding="utf-8"?>
<Rules packageId="...">
        ...
<Entity id="...">
  <Pattern confidenceLevel="85">
            <IdMatch idRef="FormattedSSN" />
            <Match idRef="USDate" minOccurs="2"/>
            <Match idRef="USAddress" />
```

```
  </Pattern>
</Entity>
        ...
         <Keyword id="FormattedSSN "> ... </Keyword>
         <Regex id=" USDate "> ... </Regex>
         <Regex id="USAddress"> ... </Regex>
        ...
</Rules>
```

Defining keyword-based matches

A common Rule requirement is to match based on well-known keyword string expressions. This is accomplished by using the `Keyword` element. The Keyword element has an "id" attribute that is used as a reference in the corresponding Entity or Affinity rules. A single Keyword element can be referenced in multiple Entity and Affinity rules.

The matching can be performed using either an exact match or word match based algorithms. Exact match uses a case-sensitive algorithm that searches for the text in the specified language. Word match applies a matching algorithm based on word boundaries. The terms to be matched can be included inline in the Keyword definition by using the Term sub-element or in an external dictionary file by specifying the Dictionary sub-element. This is configured by Term and Dictionary sub-elements.

Tip:

Use the constant based match style over regex for better efficiency and performance. Use regex matching only in cases where constant based matches are not sufficient and flexibility of regular expressions is required.

```
<Keyword id="Word_Example">
    <Group matchStyle="word">
        <Term>card verification</Term>
        <Term>cvn</Term>
        <Term>cid</Term>
        <Term>cvc2</Term>
        <Term>cvv2</Term>
        <Term>pin block</Term>
        <Term>security code</Term>
    </Group>
</Keyword>
...
<Keyword id="String_Example">
```

```
    <Group matchStyle="string">
        <Term>card</Term>
        <Term>pin</Term>
        <Term>security</Term>
    </Group>
</Keyword>
```

Defining regular expression based matches

Another common method of matching is based on regular expressions. The flexibility of regular expression matching makes this a common choice for implementing matches for data such as driver's license numbers, addresses. Common regular expression syntax is used to define the regex patterns. The table here provides examples of some of the most common regular expression tokens available.

Tip:
Use the constant based match style over regex for better efficiency and performance. Use regex matching only in cases where constant based matches are not sufficient and flexibility of regular expressions is required.

Symbol	Meaning
c	Match the literal character c once, unless it is one of the special characters.
^	Match the beginning of a line.
.	Match any character that isn't a new line.
$	Match the end of a line.
|	Logical OR between expressions.
()	Group sub-expressions.
[]	Define a character class.
*	Match the preceding expression zero or more times.
+	Match the preceding expression one or more times.

?	Match the preceding expression zero or one time.
{*n*}	Match the preceding expression *n* times.
{*n*,}	Match the preceding expression at least *n* times.
{*n*, *m*}	Match the preceding expression at least *n* times and at most *m* times.
\d	Match a digit.
\D	Match a character that is not a digit.
\w	Match an alpha character, including the underscore.
\W	Match a character that is not an alpha character.
\s	Match a whitespace character (any of \t, \n, \r, or \f).
\S	Match a non-whitespace character.
\t	Tab.
\n	New line.
\r	Carriage return.
\f	Form feed.
m	Escape *m*, where *m* is one of the meta characters described above: ^, ., $, |, (), [], *, +, ?, \, or /.

The Regex element has an "id" attribute that is used as a reference in the corresponding Entity or Affinity rules. A single Regex element can be referenced in multiple Entity and Affinity rules. The Regex expression is defined as the value of the Regex element.

```
<Regex id="CCRegex">
     \bcc\#\s|\bcc\#\:\s
</Regex>
...
<Regex id="ItinFormatted">
    (?:^|[\s\,\:])(?:9\d{2})[- ](?:[78]\d[-
     ]\d{4})(?:$|[\s\,]|\.\s)
</Regex>
...
<Regex id="NorthCarolinaDriversLicenseNumber">
    (^|\s|\:)(\d{1,8})($|\s|\.\s)
</Regex>
```

Combining multiple match elements

A common technique to increase the matching confidence is to define semantics between multiple Match elements, for example that one or more of the matches need to occur. The Any element allows the definition of the logic based between multiple matches. The Any element can be used as sub-element in the Pattern element. It contains one or more Match children elements and defines the logic of matches between them. See below for XML code examples for the Any elements with all matches, “not” logic, and exact match count.

Optional minMatches attribute can be used (default = 1) to define the minimum number of Match elements that must be met to satisfy a match. Similarly, optional maxMatches attribute can be used (default = number of children Match elements) to define the maximum number of Match elements that must be met to satisfy a match. These conditions are combined using logical operators based on the usage of the minMatches and maxMatches attributes, allowing following semantics:

- Matching all children Match elements

Not matching any children Match elements

Matching an exact subset of any children Match elements

```
<Any minMatches="3" maxMatches="3">
    <Match idRef="USDate" />
    <Match idRef="USAddress" />
    <Match idRef="Name" />
</Any>

<Any maxMatches="0">
    <Match idRef="USDate" />
    <Match idRef="USAddress" />
```

```
    <Match idRef="Name" />
</Any>

<Any minMatches="1" maxMatches="1">
    <Match idRef="USDate" />
    <Match idRef="USAddress" />
    <Match idRef="Name" />
</Any>
```

Increasing confidence level with more evidence

For entity base rules, another option of increasing confidence is to define multiple Pattern elements, each with increasing number of corroborative evidence. This is achieved by using minMatches and maxMatches for Any element to create independent patterns with increasing confidence level based on increasing number of corroborative evidence. For example:

- if one piece of corroborative evidence is found: confidence level is 65%
- if two pieces: confidence level is 75%
- if three pieces: confidence level is 85%

```
<Entity id="..." patternsProximity="300" >
    <Pattern confidenceLevel="65">
        <IdMatch idRef="UnformattedSSN" />
        <Any maxMatches="1">
            <Match idRef="USDate" />
            <Match idRef="USAddress" />
            <Match idRef="Name" />
        </Any>
    </Pattern>
    <Pattern confidenceLevel="75">
        <IdMatch idRef="UnformattedSSN" />
        <Any minMatches="2" maxMatches="2">
            <Match idRef="USDate" />
            <Match idRef="USAddress" />
            <Match idRef="Name" />
        </Any>
    </Pattern>
    <Pattern confidenceLevel="85">
        <IdMatch idRef="UnformattedSSN" />
        <Any minMatches="3">
            <Match idRef="USDate" />
```

```
            <Match idRef="USAddress" />
            <Match idRef="Name" />
        </Any>
    </Pattern>
</Entity>
```

Example: US Social Security rule

This section includes an introduction description for authoring of a rule matching a US Social Security number. First, start with a description of how we identify content that contains social security number. A Social Security Number is found if:

1. Regex matches a formatted SSN (and it's in the valid SSN range)
2. Corroborative Evidence one of the following must occur nearby:
   a. Keyword match {Social Security, Soc Sec, SSN, SSNS, SSN#, SS#, SSID}
   b. Text representing a US address
   c. Text representing a date
   d. Text representing a name

Next, translate the description into the Rule schema representation:

```
<Entity id="a44669fe-0d48-453d-a9b1-2cc83f2cba77"
         patternsProximity="300"
RecommendedConfidence="85">
    <Pattern confidenceLevel="85">
      <IdMatch idRef="FormattedSSN" />
      <Any minMatches="1">
          <Match idRef="SSNKeywords" />
          <Match idRef="USDate" />
          <Match idRef="USAddress" />
          <Match idRef="Name" />
      </Any>
    </Pattern>
</Entity>
```

For more information

Data loss prevention

Define your own DLP templates and information types

Import a DLP policy from a file

Export a DLP policy from Exchange 2013

Messaging policy and compliance > Data loss prevention > DLP policy templates >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-10*

You can view or change the details within your DLP policies without using the Exchange admin center (EAC) or Exchange Management Shell cmdlets by exporting the policies, saving them as an XML file, and modifying that XML file. Typically you would then import the XML file back into Exchange. In this way, policies can be edited independent of Exchange. However, they must meet specific format requirements, also referred to as XML schema, in order to work correctly.

Caution:

When you export DLP policies by using the procedure described here, those policies are no longer available to Exchange and will not help you scan messages. There is no ability to copy the policy information while leaving it in-place within Exchange.

For additional management tasks related to DLP, see Manage DLP policies.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

The EAC doesn't provide a way to export DLP policies or templates to an external file. Use the Exchange Management Shell to perform this task.

Use the Shell to export a DLP policy

exCollapse

This example exports all DLP policies and associated transport rules along with their attributes to an XML file in the path C:\My Documents\exportedRules.xml. Exporting a DLP policy collection removes all existing DLP policies that are defined in your organization. This isn't the same as copying your polices into a file, which implies that they will still be active and filtering messages. After you export the DLP data using the cmdlet provided here, the DLP policies are no longer available to Exchange and will not help you scan messages. Make sure that you have a backup of your current DLP policy collection before you export and modify your current DLP policies. One way to achieve this is to export and then immediately import the same XML file.

1. Open the Exchange Management Shell.
2. Type `Get-ClassificationRuleCollection`, and your organization's rules should display on screen. If you haven't created any rules of your own, you'll see the default, out-of-box rules, labeled "Microsoft Rule Package."
3. Store that in a variable by typing `$ruleCollections = Get-ClassificationRuleCollection`.
4. Now make a formatted XML file with all that data by typing `Set-Content -path "C:\My Documents\exportedRules.xml" -Encoding Byte -Value $ruleCollections.SerializedClassificationRuleCollection`.

You can now edit the XML file to adjust the policies as needed. For details on importing policies back into Exchange, see Import a DLP policy from a file.

How do you know this worked?

exCollapse

To verify that you have successfully exported your DLP policies, do the following:

1. In the EAC, go to **Compliance** > **Data Loss Prevention** and observe the listed DLP policies in the center window.
2. You have successfully exported your DLP policies if there are no items listed in the policy summary view in the center window.

See Also

Import a DLP policy from a file

Import a DLP policy from a file

Messaging policy and compliance > Data loss prevention > DLP policy templates >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-03*

You can manage sensitive information through DLP policies by importing a file that contains policy

information settings. Policies can be developed independent of Exchange as XML files. However, they must meet specific format requirements in order to work correctly. Alternatively, policies that are exported from a previous version of Exchange can be imported into Microsoft Exchange Server 2013.

Caution:

You should enable your DLP policies in test mode before running them in your production environment. During such tests, it is recommended that you configure sample user mailboxes and send test messages that invoke your test policies in order to confirm the results.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to import a DLP policy from a file

Use the following procedure to import a DLP policy from a file. In order to avoid confusion, supply a unique name for each part of your policy or rule when you have the option to provide your own name.

1. In the EAC, navigate to **Compliance management** > **Data loss prevention**.
2. Click the arrow that is next to the **Add +** icon, then click **Import policy**.
3. On the **Import policy** page, complete the following fields:
   a. **Select the file to import** Add the name of the policy file you want to install.
   b. **Name** Add a name that will distinguish this policy from others.
   c. **Description** Optionally, add a description that summarizes this policy.
   d. **More options** Select the mode or state for this policy. The new policy is not fully enabled until you specify that it should be. The default mode for a policy is test without notifications.
   e. Click **Next** to validate and import the policy.

Use the Shell to import a DLP policy from a file

This example imports a DLP policy file in the file C:\My Documents\DLP Backup.xml. Importing a

DLP policy collection from an XML file removes or overwrites all pre-existing DLP policies that were defined in your organization. Make sure that you have a backup of your current DLP policy collection before you import and overwrite your current DLP policies.

```
Import-DlpPolicyCollection -FileData ([Byte[]]$(Get-Content
-Path " C:\My Documents\DLP Backup.xml " -Encoding Byte -
ReadCount 0))
```

For more information

Data loss prevention

Policy templates from Microsoft partners

Messaging policy and compliance > Data loss prevention > DLP policy templates >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-01-31*

The data loss prevention (DLP) features can help you identify and monitor many categories of sensitive information. As you configure your DLP polices, you have the option to import a file from a source outside Microsoft Exchange to use as a DLP policy template. Microsoft partner companies can develop and distribute these types of templates. This topic will be updated with a link to help you find these companies once they are available.

DLP policy template files that can be imported are XML documents that contain instructions for scanning and analyzing message data. These templates can contain new sensitive information definitions in classification rule packages, or they can utilize rule packages that exist within Exchange already. The XML documents must adhere to a Microsoft-defined XML schema definition. For more information about the schema definition, see Define your own DLP templates and information types.

For more information

Data loss prevention

Import a DLP policy from a file

Policy Tips

Exchange Server 2013 > Messaging policy and compliance > Data loss prevention >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-02-21*

You can help to prevent your organization's Microsoft Outlook, Outlook Web App (OWA), and OWA for Devices email users from inappropriately sending sensitive information by creating data loss prevention (DLP) policies that include *Policy Tip* notification messages. Similar to MailTips that were introduced in Microsoft Exchange Server 2010, Policy Tip notification messages are displayed to users in Outlook while they are composing an email message. Policy Tip notification messages only show up if something about the sender's email message seems to violate a DLP policy that you have in place and that policy includes a rule to notify the sender when the conditions that you establish are met. For more general information about data loss prevention, see Data loss prevention.

In order to show Policy Tips to your email senders, your rules must include the **Notify the sender with a Policy Tip** action. You can add this in the rules editor from the Exchange Administration Center. For more information, see Manage policy tips.

The transport rule agent that enforces DLP policies does not differentiate between email message attachments, body text, or subject lines while evaluating messages and the conditions within your policies. For example, if a user creates an email message that includes a credit card number in the body of the message and then attempts to address the message to a recipient outside your organization, then a Policy Tip notification message can be shown to that user in Outlook or Outlook Web App reminding them of your enterprise's expectations for such information. However, this type of notification will only show up if you have configured a DLP policy that restricts the example actions described; in this case adding an external email alias to the header of a message with credit card data. There is a great variety of conditions, actions, and exceptions you can choose from while creating DLP policies. This variety allows you to tailor your data loss prevention efforts in a way that meets your specific organization's needs.

Any time you use either the notify sender action or an override action within a rule, we recommend that you also include the condition that the message was sent from within your organization. You can do this by using the policy rules editor to add the following condition: **The sender is located...** > **inside the organization**. Learn more about changing existing DLP policies at Manage DLP policies. This is a best practice recommendation because the notify sender action is applied as part of your company's message creation experience. The senders referred to by the action are the authors of messages within your company. The user interaction presented by Policy Tips cannot be acted upon by your users for incoming messages and will be ignored when the sender is located outside your organization. You can apply DLP policies to scan incoming messages and take a variety of actions, but when you do this, don't add the notify sender action.

If email senders in your organization who are in the act of composing a message are made aware of your organizational expectations and standards in real time through Policy Tip notifications, then they are less likely to violate standards that your organization wants to enforce.

Note:

- **Exchange Online:** DLP is a premium feature that requires an Exchange Online Plan 2 subscription. For more information, see Exchange Online Licensing.
- **Exchange 2013:** DLP is a premium feature that requires an Exchange Enterprise Client Access License (CAL). For more information about CALs and server licensing, see Exchange Server Licensing.
- If your organization is using Exchange 2013 SP1 or Exchange Online, Policy Tips are available to people sending mail from Outlook 2013, Outlook Web App, or OWA for Devices. However, if your organization is currently using Exchange 2013, Policy Tips are only available to people sending email from Outlook 2013.

Default text for Policy Tips and rule options

You have a range of possible options when you add sender notification rules to DLP policies. When you add a rule to notify the sender by using the **Notify the sender with a Policy Tip** action within a DLP policy, you can choose how restrictive to be. The notification options in the following table are available. For general information about editing policies, see Manage DLP policies. For specific information about creating Policy Tips, see Manage policy tips.

Notification rule	Meaning	Default Policy Tip notification message that Outlook users will see
Notify only	Similar to MailTips, this causes an informative Policy Tip notification message about a policy violation. A sender can prevent this type of tip from showing up by using a Policy Tip options dialog box that can be accessed in Outlook.	This message may contain sensitive content. All recipients must be authorized to receive this content.
Reject message	The message will not be delivered until the condition is no longer present. The sender	This message may contain sensitive content. Your organization won't allow this

	is provided with an option to indicate that their email message does not contain sensitive content. This is also known as a false-positive override. If the sender indicates this, then Outlook will allow the message to leave the outbox so that the user's report may be audited, but Exchange will block the message from being sent.	message to be sent until that content is removed.
Reject unless false positive override	The result with this notification rule is similar to the **Reject message** notification rule. However, if you select this then Exchange will allow the message to be sent to the intended recipient, instead of blocking the message.	**Before the sender selects an option to override:** This message may contain sensitive content. Your organization won't allow this message to be sent until that content is removed. **After the sender selects an option override:** Your feedback will be submitted to your administrator when the message is sent.
Reject unless silent override	The message will not be delivered until the condition is no longer present or the sender indicates an override. The sender is provided with an option to indicate that they wish to override the policy.	**Before the sender selects an option to override:** This message may contain sensitive content. Your organization won't allow this message to be sent until that content is removed.

		After the sender selects an option override: You have overridden your organization's policy for sensitive content in this message. Your action will be audited by your organization.
Reject unless explicit override	The result with this notification rule is similar to the **Reject unless silent override** notification rule, except that in this case when the sender attempts to override the policy, they are required to provide a justification for overriding the policy.	**Before the sender selects an option to override:** This message may contain sensitive content. Your organization won't allow this message to be sent until that content is removed. **After the sender selects an option override:** You have overridden your organization's policy for sensitive content in this message. Your action will be audited by your organization.

Create your own Policy Tip notification messages

You can customize the text of a Policy Tip notification that email senders see in their email program. If you do this, keep in mind that your custom Policy Tip notification text will not appear unless you also configure a DLP policy with a rule that will cause the customized text to appear. For procedures that explain how to create your own Policy Tips, see Manage policy tips. The custom text that you create can replace the default text shown in the column of the table above that is labeled "Default Policy Tip notification message".

When you work in the Policy Tip setting editor—not the rules editor—the following four actions can be configured with custom text. As stated previously in this topic, in order for any of your custom text to appear, a DLP policy rule must include the **Notify the sender with a Policy Tip** action. Add the action to a rule by using the DLP rules editor.

Policy Tip Notification Action	Meaning

Notify the sender	Your text only appears when a **Notify the sender, but allow them to send** action is initiated. This type of action can be configured in the DLP rules editor.
Allow the sender to override	Your text only appears when the following actions are initiated: **Block the message unless it's a false positive**, **Block the message, but allow the sender to override and send**. These types of actions can be configured in the DLP rules editor.
Block the message	Your text only appears when a **Block the message** action is initiated. This type of action can be configured in the DLP rules editor.
Link to compliance URL	Your text only appears when a **Block the message, but allow the sender to override with a business justification and send** action is initiated. This type of action can be configured in the DLP rules editor.

For more information

Data loss prevention

Manage DLP policies

Manage policy tips

Manage policy tips

Messaging policy and compliance > Data loss prevention > Policy Tips >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-02-21*

Policy Tips are informative notices that are displayed to email senders while they're composing a

message. The purpose of the Policy Tip is to educate users that they might be violating the business practices or policies that you are enforcing with the data loss prevention (DLP) policies that you have established. The following procedures will help you begin using Policy Tips. For an overview about working with Policy Tips, see Policy Tips.

What do you need to know before you begin?

- Estimated time to complete each procedure: 30 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.
- Policy Tips will only show up for email senders when the following conditions are met:
  1. Sender's message client program is Microsoft Outlook 2013. If your organization has deployed Exchange 2013 SP1 or is using Exchange Online, Policy Tips also show up in Outlook Web App and OWA for Devices.
  2. A transport rule exists that invokes Policy Tip notifications. You can create such a transport rule by configuring a DLP policy that includes the action **Notify the sender with a Policy Tip**.
  3. The content of a message header, message body, or message attachment that is scanned by your transport agent meets the conditions established within the DLP policies or rules that also include Policy Tip notification rules. Put another way, the Policy Tip only shows up for end-users if they do something that causes the associated rule to take action.
- The default Policy Tip notification text that is built into the system will be shown if you don't use the Policy Tip settings feature to customize your Policy Tip text. To learn more about the default text, see Policy Tips.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create or modify a notify-only Policy Tip

This procedure results in an informational Policy Tip being shown to an email sender when the conditions of a specific rule are met. In Microsoft Outlook, the sender can prevent this tip from showing up by using a Policy Tip options dialog box. To configure custom Policy Tip text, see Create custom Policy Tip notification text.

Use the EAC to configure notify-only Policy Tips

1. In the EAC, go to **Compliance management** > **Data loss prevention**.
2. Double-click one of the policies that appear in your list of policies or highlight one item and

select **Edit** ✎.

3. On the **Edit DLP policy** page, select **Rules**.
4. To add Policy Tips to an existing rule, highlight the rule and select **Edit** ✎.

To add a new blank rule that you can fully customize, select **Add +** and then select **Create a new rule** .

5. In **Apply this rule if**, select, **The message contains sensitive information**. This condition is required.
6. Select **+**, select the sensitive information types, select **Add**, select **OK**, and then select **OK**.
7. In the **Do the following** box, select **Notify the sender with a Policy Tip**, and select an option in the **Choose whether the message is blocked or can be sent** drop-down list, and then select **OK**.
8. If you want to add additional conditions or actions, at the bottom of the window, select **More options**.

Note:

Only the following conditions can be used:

- **The recipient is**
- **The recipient is located**
- **The sender is**
- **The sender is a member of**
- **The sender is located**

The following actions can't be used:

- **Reject the message and include an explanation**
- **Reject the message with the enhanced status code of**
- **Delete the message without notifying anyone**

1. In the **Choose a mode for this rule** list, select whether you want the rule to be enforced. We recommend testing the rule first.
2. Select **Save** to finish modifying the rule and save your changes.

How do you know this worked?

To verify that you have successfully created a Policy Tip that will only notify a sender, do the following:

1. In the EAC, go to **Compliance management** > **Data loss prevention**.
2. Select the policy that you expect to contain a notification message.
3. Select **Edit** ✎ and then select **Rules**.
4. Select the specific rule that you expect to contain a notification message.
5. Confirm that your **Notify the sender** action appears in the lower portion of the rule summary.

Create or modify a block-message Policy Tip

This procedure results in a Policy Tip being shown to an email sender that indicates a message is rejected and it will not be delivered until the problematic condition is no longer present. The sender is provided with an option to indicate that their email message does not contain the

problematic condition. This is also known as a false-positive override. If the sender indicates this, then the message can leave the outbox and the user's report may be audited. However, Exchange will block the message from being sent. To configure custom Policy Tip text, see Create custom Policy Tip notification text.

Use the EAC to configure block-message Policy Tips

1. In the EAC, go to **Compliance management** > **Data loss prevention**.
2. Double-click one of the policies that appear in your list of policies or highlight one item and select **Edit** ✎.
3. On the **Edit DLP policy** page, select **Rules**.
4. To add Policy Tips to an existing rule, highlight the rule and select **Edit** ✎.
5. To add a new blank rule that you can fully customize, select **Add +**.
6. To add an action that will reveal a Policy Tip, select **More options...** and then select the **Add action** button.
7. From the drop down list, select **Notify the sender with a Policy Tip** and then select **Block the message**.
8. Select **OK**, then select **Save** to finish modifying the rule and save your changes.

How do you know this worked?

To verify that you have successfully created a reject message Policy Tip, do the following:

1. In the EAC, go to **Compliance management** > **Data loss prevention**.
2. Select one time to highlight the policy that you expect to contain a notification message.
3. Select **Edit** ✎ and then select **Rules**.
4. Select one time to highlight the specific rule that you expect to contain a notification message.
5. Confirm that your **Notify the sender that the message can't be sent** action appears in the lower portion of the rule summary.

Create or modify a block-unless-override Policy Tip

There are four options for Policy Tips that can reject messages or prevent messages from leaving the sender's outbox. To learn more about these options, see Policy Tips.

Use the EAC to configure block-unless override Policy Tips

1. In the EAC, go to **Compliance management** > **Data loss prevention**.
2. Double-select one of the policies that appear in your list of policies or highlight one item and select **Edit** ✎.
3. On the **edit DLP policy** page, select **Rules**.
4. To add Policy Tips to an existing rule, highlight the rule and select **Edit** ✎.

To add a new blank rule that you can fully customize, select **Add +** and then select **More options...**

.

5. To add the action that will reveal a Policy Tip, Select the **Add action** button.
6. From the drop down list, select **Notify the sender with a Policy Tip** and then select **Block the message, but allow the sender to override and send**.

7. Select **OK**, then select **Save** to finish modifying the rule and save your changes.

How do you know this worked?

To verify that you have successfully created a reject unless override Policy Tip, do the following:

1. In the EAC, go to **Compliance management** > **Data loss prevention**.
2. Select one time to highlight the policy that you expect to contain a notification message.
3. Select **Edit** and then select **Rules**.
4. Select one time to highlight the specific rule that you expect to contain a notification message.
5. Confirm that your **Block the message, but allow the sender to override and send** action appears in the lower portion of the rule summary.

Create custom Policy Tip notification text

This optional procedure will help you to customize the Policy Tip notification text that email senders see in their email program. If you do this, your custom Policy Tip notification text will not appear unless you also configure a DLP policy rule with an action that will cause the notification to appear. Keep in mind that there are default system Policy Tip notifications that can be shown if you do not customize your Policy Tip notification text. To learn more about the default text, see Policy Tips.

Use the EAC to create and manage custom Policy Tip notification text

1. In the EAC, go to **Compliance management** > **Data loss prevention**.
2. Select **Policy Tip settings**.
3. To add a new Policy Tip with your own customized message, select **Add +**. For more information about the action choices available, see Policy Tips.

To modify an existing Policy Tip, highlight the tip and select **Edit**.

To delete an existing Policy Tip, highlight it and select **Delete** and then confirm your action.

4. Select **Save** to finish modifying the Policy Tip and save your changes.
5. Select **Close** to finish managing your Policy Tips and save your changes.

Use the Shell to create custom Policy Tip notification text

The following example creates a new English-language Policy Tip that will block a message from being sent. The text of this custom Policy Tip is changed to the following value: "This message appears to contain restricted content and will not be delivered."

```
New-PolicyTipConfig -Name en\Reject -Value "This message appears to contain restricted content and will not be delivered."
```

For more information about DLP cmdlets, see Policy and compliance cmdlets.

Use the Shell to modify custom Policy Tip notification text

The following example modifies an existing English-language, notify-only Policy Tip. The text of

this custom Policy Tip is changed to "Sending bank account numbers in email is not recommended."

```
Set-PolicyTipConfig en\NotifyOnly "Sending bank account numbers in email is not recommended."
```

For more information about DLP cmdlets, see Policy and compliance cmdlets.

How do you know this worked?

To verify that you have successfully created custom Policy Tip text, do the following:
1. In the EAC, go to **Compliance management** > **Data loss prevention**.
2. Select **Policy Tip settings** .
3. Select **Refresh** .
4. Confirm that your action, locale and text for that locale appear in the list.

For more information

Data loss prevention

Policy Tips

Transport rules

Exchange 2010 MailTips

Document Fingerprinting

Exchange Server 2013 > Messaging policy and compliance > Data loss prevention >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-02-22*

Information workers in your organization handle many kinds of sensitive information during a typical day. *Document Fingerprinting* makes it easier for you to protect this information by identifying standard forms that are used throughout your organization. This topic describes the concepts behind Document Fingerprinting. If you'd like to learn how to create a document fingerprint, see Protect form data with document fingerprinting.

Basic scenario for Document Fingerprinting

Document Fingerprinting is a Data Loss Prevention (DLP) feature that converts a standard form into a sensitive information type, which you can use to define transport rules and DLP policies. For example, you can create a document fingerprint based on a blank patent template and then create

a DLP policy that detects and blocks all outgoing patent templates with sensitive content filled in. Optionally, you can set up Policy Tips to notify senders that they might be sending sensitive information, and the sender should verify that the recipients are qualified to receive the patents. This process works with any text-based forms used in your organization. Additional examples of forms that you can upload include:

- Government forms
- Health Insurance Portability and Accountability Act (HIPAA) compliance forms
- Employee information forms for Human Resources departments
- Custom forms created specifically for your organization

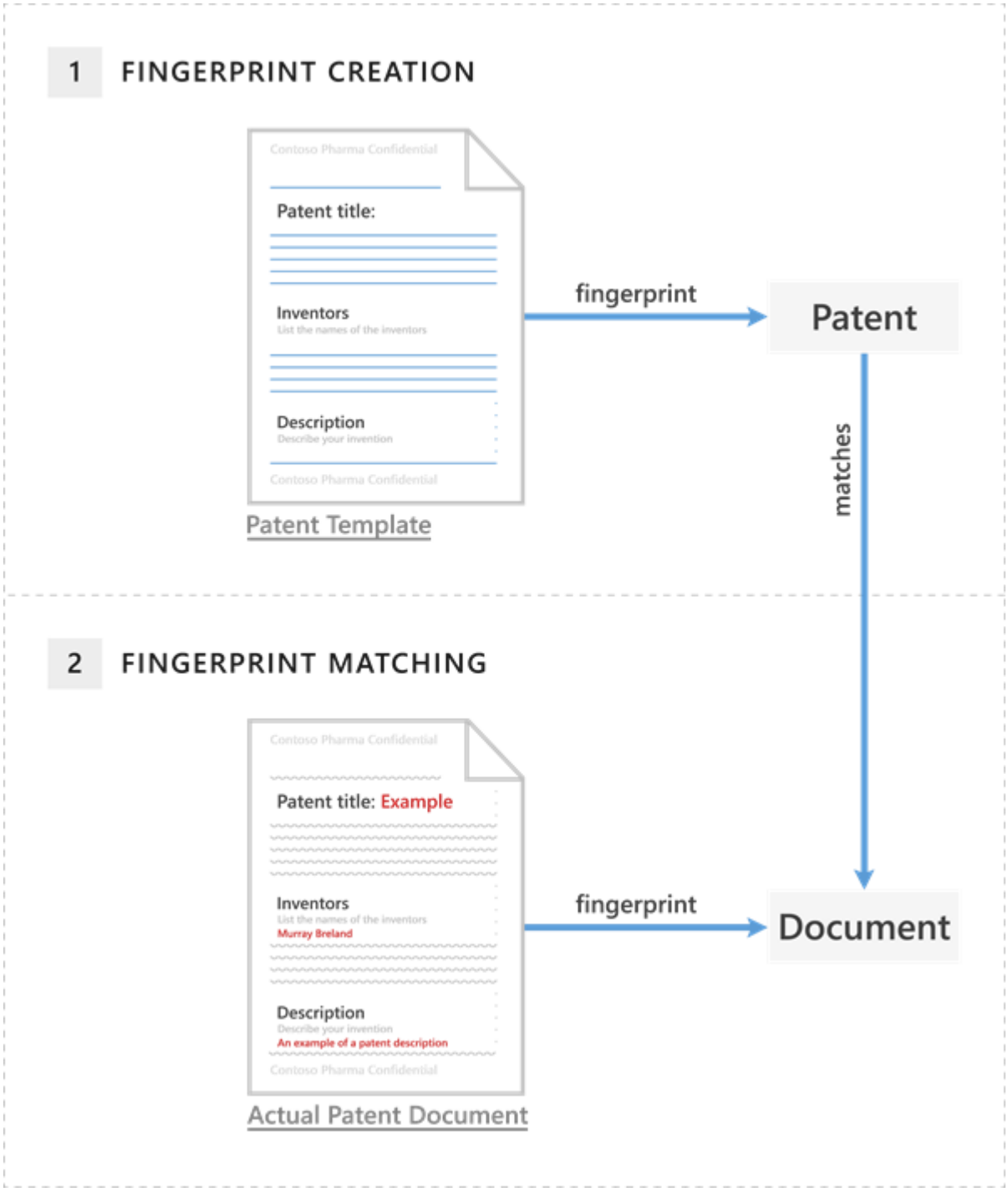
Ideally, your organization already has an established business practice of using certain forms to transmit sensitive information. After you upload an empty form to be converted to a document fingerprint and set up a corresponding policy, the DLP agent will detect any documents in outbound mail that match that fingerprint.

How Document Fingerprinting works

You've probably already guessed that documents don't have actual fingerprints, but the name helps explain the feature. In the same way that a person's fingerprints have unique patterns, documents have unique word patterns. When you upload a file, the DLP agent identifies the unique word pattern in the document, creates a document fingerprint based on that pattern, and uses that document fingerprint to detect outbound documents containing the same pattern. That's why uploading a form or template creates the most effective type of document fingerprint. Everyone who fills out a form uses the same original set of words and then adds his or her own words to the document. As long as the outbound document isn't password protected and contains all the text from the original form, the DLP agent can determine if the document matches the document fingerprint.

The following example shows what happens if you create a document fingerprint based on a patent template, but you can use any form as a basis for creating a document fingerprint.

Example of a patent document matching a document fingerprint of a patent template



The patent template contains the blank fields “Patent title,” “Inventors,” and “Description” and descriptions for each of those fields—that’s the word pattern. When you upload the original patent template, it’s in one of the supported file types and in plain text. The DLP agent uses an algorithm to convert this word pattern into a document fingerprint, which is a small Unicode XML file containing a unique hash value representing the original text, and the fingerprint is saved as a data classification in Active Directory. (As a security measure, the original document itself isn’t stored on the service; only the hash value is stored, and the original document can’t be reconstructed from the hash value.) The patent fingerprint then becomes a sensitive information type that you can associate with a DLP policy. After you associate the fingerprint with a DLP policy, the DLP agent detects any outbound emails containing documents that match the patent fingerprint and deals with them according to your organization’s policy. For example, you might want to set up a DLP policy that prevents regular employees from sending outgoing messages containing patents. The

DLP agent will use the patent fingerprint to detect patents and block those emails. Alternatively, you might want to let your legal department to be able to send patents to other organizations because it has a business need for doing so. You can allow specific departments to send sensitive information by creating exceptions for those departments in your DLP policy, or you can allow them to override a policy tip with a business justification. For more detailed information about creating DLP policy rules and exceptions, see **DLP procedures**, and to learn more about setting up policy tips that users can override, see Manage policy tips.

Supported file types

Document Fingerprinting supports the same **Using transport rules to inspect message attachments**. One quick note about file types: neither transport rules nor Document Fingerprinting supports the .dotx file type, which can be confusing because that's a template file in Word. When you see the word "template" in this and other Document Fingerprinting topics, it refers to a document that you have established as a standard form, not the template file type.

Limitations of document fingerprinting

The Document Fingerprinting DLP agent won't detect sensitive information in the following cases:

- Password protected files
- Files that contain only images
- Documents that don't contain all the text from the original form used to create the document fingerprint

For more information

Protect form data with document fingerprinting

Integrating sensitive information rules with transport rules

DLP procedures

Protect form data with document fingerprinting

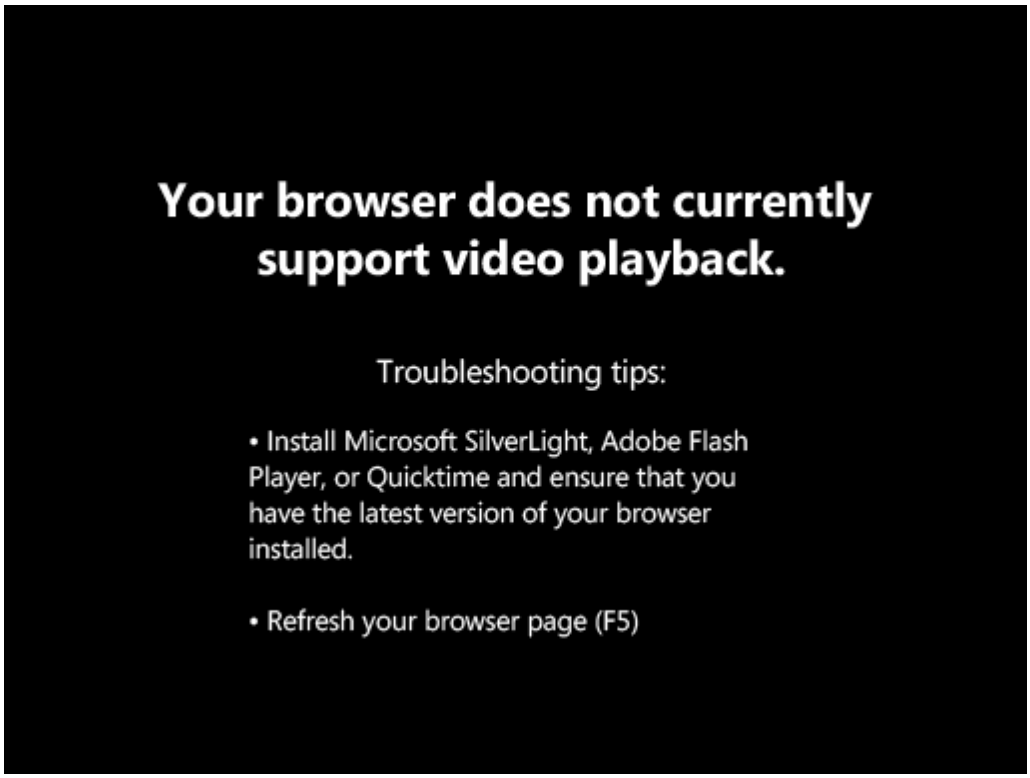
Messaging policy and compliance > Data loss prevention > Document Fingerprinting >

Applies to: *Exchange Server 2013, Exchange Online*

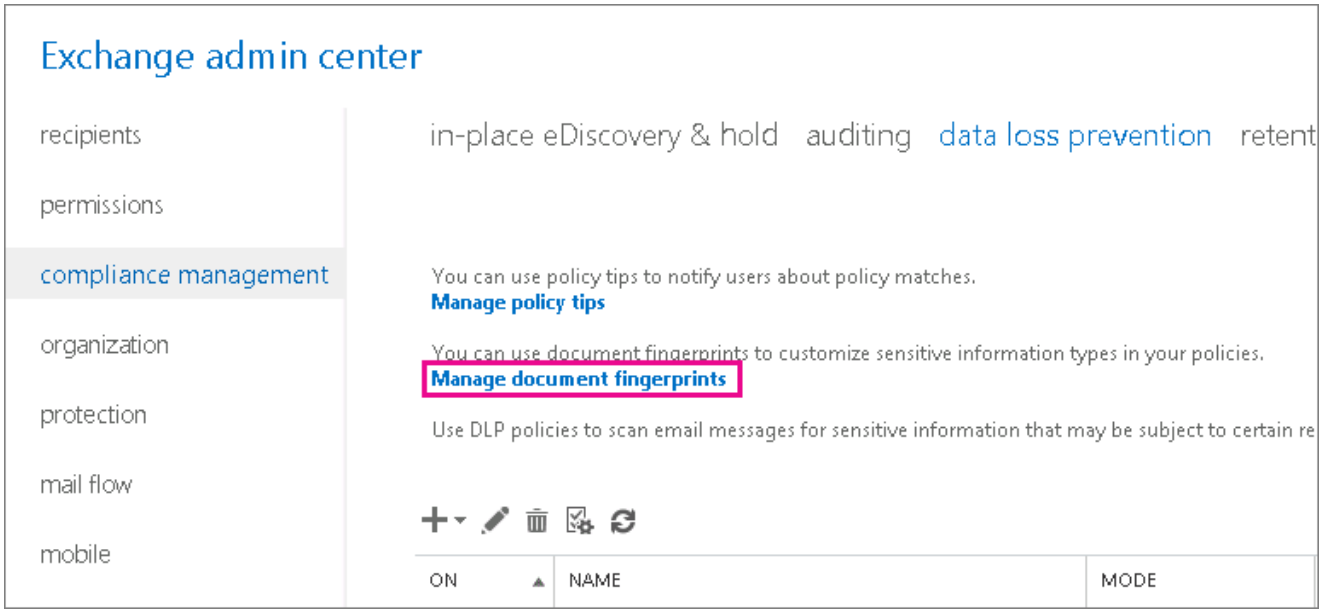
Topic Last Modified: *2014-09-01*

If your organization uses forms to collect sensitive information, users might try emailing those

forms to outside contacts, which creates a security risk. Data loss prevention (DLP) in Exchange helps you protect that information by detecting it with Document Fingerprinting. To use document fingerprinting, simply upload a blank form, such as an intellectual property document, government form, or other standard form used in your organization. Then, add the resulting document fingerprint to a DLP policy or transport rule. Here's how.



Use the EAC to create a document fingerprint



1. In the Exchange Administration Center EAC, go to **compliance management** > **data loss prevention**.
2. Click **Manage document fingerprints**.
3. On the document fingerprints page, click **New +** to create a new document fingerprint.
4. Give the document fingerprint a **Name** and **Description**. (The name you choose will appear in

the sensitive information types list.)

5. To upload a form, click **Add +**.
6. Choose a form, and click **Open**. (Make sure that the file you upload contains text, isn't password protected, and is in one of the **Using transport rules to inspect message attachments**. Otherwise, you'll get an error when you try creating the fingerprint.) Repeat for any additional files you want to add to the document list for this document fingerprint. You can also add or remove files from this document fingerprint later if you want.
7. Click **Save**.

The document fingerprint is now part of your sensitive information types, and you can add it to a DLP policy or add it to a transport rule via the **If the message contains…Sensitive Information** condition.

For more information about adding rules to a DLP policy, see the "Change a DLP policy" section of Manage DLP policies, and for more information about modifying transport rules, see Integrating sensitive information rules with transport rules. If you want to create a new policy, see Create a DLP policy from a template.

Use the Shell to create a classification rule package based on document fingerprinting

Tip:
Even though you can create and modify classification rule packages in the Shell, you might find that creating document fingerprints is a little simpler in the EAC. We recommend you try it there before trying this procedure in the Shell.

DLP uses classification rule packages to detect sensitive content in messages. To create a classification rule package based on a document fingerprint, use the **New-Fingerprint** and **New-DataClassification** cmdlets. Because the results of **New-Fingerprint** aren't stored outside the data classification rule, you always run **New-Fingerprint** and **New-DataClassification** or **Set-DataClassification** in the same PowerShell session. The following example creates a new document fingerprint based on the file C:\My Documents\Contoso Employee Template.docx. You store the new fingerprint as a variable so you can use it with the **New-DataClassification** cmdlet in the same PowerShell session.

```
$Employee_Template = Get-Content "C:\My Documents\Contoso
Employee Template.docx" -Encoding byte
$Employee_Fingerprint = New-Fingerprint -FileData
$Employee_Template -Description "Contoso Employee Template"
```

Now, let's create a new data classification rule named "Contoso Employee Confidential" that uses the document fingerprint of the file C:\My Documents\Contoso Customer Information Form.docx.

```
$Employee_Template = Get-Content "C:\My Documents\Contoso
Customer Information Form.docx" -Encoding byte
$Customer_Fingerprint = New-Fingerprint -FileData
$Customer_Form -Description "Contoso Customer Information
Form"
New-DataClassification -Name "Contoso Customer
Confidential" -Fingerprints $Customer_Fingerprint -
Description "Message contains Contoso customer
information."
```

You can now use the **Get-DataClassification** cmdlet to find all DLP data classification rule packages, and in this example, "Contoso Customer Confidential" is part of the data classification rule packages list.

Finally, add the "Contoso Customer Confidential" data classification rule package to a DLP policy.

```
New-TransportRule -Name "Notify :External Recipient Contoso
confidential" -NotifySender NotifyOnly -Mode Enforce -
SentToScope NotInOrganization -
MessageContainsDataClassification @{Name=" Contoso Customer
Confidential"}
```

The DLP agent now detects documents that match the Contoso Customer Form.docx document fingerprint.

For syntax and parameter information, see New-Fingerprint, New-DataClassification, Set-DataClassification, and Get-DataClassification.

For more information

Document Fingerprinting

Manage DLP policies

Integrating sensitive information rules with transport rules

DLP policy detection reports

Exchange Server 2013 > Messaging policy and compliance > Data loss prevention >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-25*

Data loss prevention (DLP) policy detection management broadly defines the activities that an organization performs in order to identify, investigate, and resolve DLP policy violations. In order to manage incidents, you need access to information that identifies what was detected by your DLP policies. This detection information is integrated with existing Microsoft Exchange Server 2013 data and log formats so that you can leverage an existing rich system of data to manage your mail flow incidents. Detection information is also presented in Microsoft Exchange Online through the **Reports** section, which is accessible from both the Office 365 admin center and the Exchange Admin Center. Learn more about this in Exchange Online at **Summary data reports for DLP policies**.

For information about creating an incident report along with a single policy detection event, see Create incident reports for DLP policy detections. For more information about message logs, see Track messages with delivery reports.

Note:

Exchange Online: DLP is a premium feature that requires an Exchange Online Plan 2 subscription. For more information, see Exchange Online Licensing.
Exchange 2013: DLP is a premium feature that requires an Exchange Enterprise Client Access License (CAL). For more information about CALs and server licensing, see Exchange Server Licensing.

Audit information

Data related to DLP detection management in Exchange is integrated into the message tracking logs, also known as delivery reports. The capabilities reuse much of the existing logging framework available in the system. For more information about the DLP reports in Exchange Online, see **Summary data reports for DLP policies**. For general information, including understanding the structure of the message tracking log files, please review existing content in Managing Message Tracking in the TechNet Library or Track messages with delivery reports.

The delivery report is a detailed log of all message activity as messages are transferred to and from a computer that is running the Transport service on a Mailbox server. For Exchange 2013, but not Exchange Online the message tracking log can be accessed through the Exchange Management Shell by using the **Get-MessageTrackingLog** cmdlet. DLP data is integrated into the delivery report following existing data formats and conventions.

Data logging format

Message tracking logs contain data from the agents involved in processing the mail flow content. For DLP, the transport rule agent (TRA) is used to invoke deep message content scanning and to apply the policies defined as part of the ETRs. The existing AgentInfo Event is used to add DLP related entries in the message tracking log.

The agent name will be **TRA** or **Transport Rule Agent** in the AgentInfo event. A single AgentInfo event will be logged per message describing the DLP processing applied to the message. The **CustomData** field of the message tracking log entry field is where the DLP data logged by the transport rule agent will appear. This field may contain multiple entries: one data classification and client information line for each data classification found in the message, one rule line for each rule that applies to the message, and one health monitoring line for each rule that exceeds the load or execution time threshold.

An example of the DLP log entry is displayed here. The output has been formatted to display strings in separate lines with new lines between.

Source: AGENT

EventId: AGENTINFO

CustomData: S:TRA=DC|dcid=41BFDBC6C9D811E0816A3CD34824019B|count=10|conf=77;

S:TRA=DC|dcid=C7ECCBA0CA0011E0B6C00B124924019B|count=3|conf=81;

S:TRA=CI|sndOverride=or|just=Business Reason;

S:TRA=CI|sndOverride=fp;

S:TRA=ETR|ruleId=FC2AA60C9D811E0AFC076D34824019B|
dlpid=1B81CC82C9DB11E09052C5D64824019B|st=2010-11-03 15:30T|action=PrependSubject|
action=Encrypt|sev=2|mode=audit|dcid=41BFDBC6C9D811E0816A3CD34824019B|sndOverride=or;

S:TRA=ETR|ruleId=AB2AA60C9D811E0AFC076D34824019B|
dlpid=1B81CC82C9DB11E09052C5D64824019B|st=2010-11-03 15:30T|action=Encrypt|sev=1|
mode=enabled|dcid=C7ECCBA0CA0011E0B6C00B124924019B|sndOverride=fp;

S:TRA=ETRP|ruleId=C27D21EECA0311E0BCB896154924019B|LoadW=200|LoadC=100|
ExecW=5500|ExecC=200;

The Transport Rule Agent requires grouping of the rule ID, DLP Policy ID (optional), last modified date, action, severity, mode, detected data classification (optional), and sender override (optional) based on rule ID (indicated by “TRA=ETR” in the log line). It also requires the data classification ID, count, and confidence level of classifications to be grouped by classification name (indicated by “TRA=DC” in the log line).

Additional groupings include data classification ID, sender override (optional), and override justification (optional) based on data classification ID for all classifications that were detected on the client (indicated by “TRA=CI” in the log line). The Transport Rule Agent also requires the rule ID,

load Wall clock (optional), load CPU clock (optional), execution Wall clock (optional), and execution CPU clock (optional) be grouped by rule ID for all rules that exceed the load or execution Wall or CPU clock thresholds (indicated by "TRA=ETRP" in the log line).

The following is a complete list of the data fields. All data in the MTL is type string. Format column describes how to recognize each field in the Message Tracking Log. Optional Field column specifies what fields might not be logged when a rule matches. DLP Specific column shows what fields are specific to the DLP feature.

Field name	Description	Format	Optional field	DLP specific
TRA	Transport Rule Agent; type AgentName	TRA=DC, ETR, CI, or ETRP	Mandatory	No
DC	Data Classification; type groupName	TRA=DC	Optional	Yes
ETR	Exchange Transport Rule; type groupName	TRA=ETR	Mandatory	No
CI	Client Information, type groupName	TRA=CI	Optional	Yes
ETRP	Exchange Transport Rule Performance; type groupName	TRA=ETRP	Optional	No
dcid	ID of the Data Classification	dcid=GUID	Optional	Yes
count	Count of the Data Classification	count=Integer	Optional	Yes
conf	Confidence level of the Data Classification	conf=Integer (Percent)	Optional	Yes

sndOverride	Sender override; the field is optional. In the TRA=CI line, when field is set to "or" signifies the data classification was overridden. If the field is set to "fp" signifies the data classification was reported as a false positive. In the TRA=ETR line, when the field is set to "or" signifies the rule or part of the rule was overridden. If the field is set to "fp" signifies the rule or part of the rule was reported as a false positive.	sndOverride=or or fp Where "or" represents override and "fp" means false positive. The sndOverride field is present when an end-user had reported either an override or false positive for a rule.	Optional	Yes
just	Justification; the field is optional and only available when the sender override field is equal to "or" in the TRA=CI line. Justification text	just=IW input justification string Justification field is only logged when end user reports an override.	Optional	Yes

	provided by the end user as the reason the data classification should be overridden.			
ruleId	ID for a rule	ruleId=GUID	Mandatory	No
dlpId	ID for a DLP Policy. The field is optional; if there is no dlpId then the rule doesn't belong to a DLP Policy.	dlpId=GUID	Optional	Yes
st	Last Modified Date of a rule	st=UTC date-time	Mandatory	No
action	Action taken by a rule; could have multiple actions per rule	action=single action If there are multiple actions applied for a rule, there will be multiple action fields.	Mandatory	No
sev	Audit severity of the rule	sev=1, 2, or 3 Where 1 represents low, 2 is medium, and 3 means high.	Optional	No

mode	State of the rule when it was hit (enforcement, audit, or auditandnotify).	mode=audit, auditandnotify, or enforcement	Mandatory	No
loadW	Load Wall Clock; the field is optional	loadW=time in ms	Optional	No
loadC	Load CPU Clock; the field is optional	loadC=time in ms	Optional	No
execW	Execute Wall Clock; the field is optional	execW=time in ms	Optional	No
execC	Execute CPU Clock; the field is optional	execC=time in ms	Optional	No
message-id	ID of the message	message-id=ID of message	Mandatory	No
date-time	Date and time the message was sent in universal time	date-time=UTC date-time	Mandatory	No
sender-address	Email address specified in the sender field	sender-address=Email address	Mandatory	No
recipient-address	Email address(es) of the message's recipient(s)	recipient-address=Email address	Mandatory	No

message-subject	Data found in the subject field of the message	message-subject=end-user input subject string	Mandatory	No

For more information

Data loss prevention

Create incident reports for DLP policy detections

Track messages with delivery reports

Create incident reports for DLP policy detections

Messaging policy and compliance > Data loss prevention > DLP policy detection reports >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-07*

In Microsoft Exchange, you can establish an action to create an incident report within a DLP policy rule set. Additionally, you can indicate to whom the report should be sent and what to do with the original message. The incident report can contain any of the following information.

Content of an incident management report

The **Generate Incident Report** action enables users to send incident reports to an incident management mailbox. A single incident report will be generated for each message only if the **Generate Incident Report** action is applied within a policy.

The following is a complete list of the line names in the incident report template. The format column describes how to recognize each field in the report. The optional field column specifies what fields might not be in the Report for each rule match. The DLP specific column shows what fields exist as a result of the DLP feature.

Line name	Description	Format	Optional field	DLP specific
Message-Id	ID of the original	Message-Id: ID of	Mandatory	No

	sent message	message		
Sender	True sender of the original message	Sender: Email address of sender	Mandatory	No
Subject	Subject of the original message	Subject: end-user input subject string	Mandatory	No
To	Recipient or recipients of the original message Each To line will only contain a single recipient, and there can be up to 10 recipients displayed in the Incident Report. If there are additional recipients, the next To line will display the remaining number of recipients.	To: Email address of recipient	Mandatory	No
CC	CC email address of the original message; the line is optional Each CC line will only contain a single CC email	CC: Email address of CC recipient	Optional	No

	address, and there can only be up to 10 CC email addresses that are displayed in the Incident Report. If there are additional CC addresses, the next CC line will display the remaining number of CC email addresses.			
BCC	BCC email address of the original message; the line is optional Each BCC line will only contain a single BCC email address, and there can only be up to 10 BCC addresses that are displayed in the Incident Report. If there are additional BCC email addresses, the following BCC line will display the remaining	BCC: Email address of BCC recipient	Optional	No

	number of BCC email addresses.			
Severity	Audit severity of the rule hit; displays the highest severity if multiple rules were hit.	Severity: Low, Medium, or High	Optional	No
Override	Displays if an override was reported for the message, and the justification of the override if provided.	Override: Yes, Justification: IW input justification string	Optional	Yes
False Positive	Displays if a false positive was reported for the message.	False Positive: Yes	Optional	Yes
Data Classification	Detected data classifications found in the original message; the line is optional. Each data classification line will only contain a single detected classification along with its count, confidence,	Data Classification: sensitive information type, Count: instances of the sensitive information found in the message, Confidence: percent value, Recommended Minimum Confidence:	Optional	Yes

	and recommended minimum confidence level. Up to 5 detected classifications will be displayed in the Incident Report. If the detected classification was an affinity, the count value does not apply and will not be shown.	percent value		
Rule Hit	Displays all the rules that hit the original message. Includes the name of the rule that was hit, the DLP Policy (optional) that the rule resides in, action(s) that were taken on the message because of the rule, data classification(s) in the rule that caused the rule to hit, and the definition of the	Rule Hit: rule name, DLP Policy: DLP Policy name if applicable, Action: single action, Data Classification: sensitive information type, Definition: rule definition if applicable	Mandatory	No

	rule.			
ID Match	Displays the matched data classification, the exact matched content from the message, and the primary evidence of the data classification match; the line is optional.	ID Match: sensitive information type, Value: actual value of the sensitive data, Context: text around the sensitive data in the message	Optional	Yes

For more information

DLP policy detection reports

Data loss prevention

DLP procedures

Exchange Server 2013 > Messaging policy and compliance > Data loss prevention >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-02*

You can begin using a data loss prevention (DLP) solution in your messaging environment by using the following procedures. To learn about concepts and objectives for DLP, see Data loss prevention.

Create a DLP policy from a template Information to help you configure a Microsoft-supplied, pre-built set of policy rules. Policy templates are an easy way to get started with managing message data that is associated with several common legal and regulatory requirements.

Create a custom DLP policy Information to help you configure policy rules to meet the specific needs of your organization which may not be covered in one of the pre-existing DLP templates. The rule conditions that are available to you in a single policy include all the traditional transport rules in addition to the new sensitive information types.

Import a DLP policy from a file Information to help you import a file that contains policy

information settings. Policies that are created independent of Exchange as XML files must meet specific format requirements in order to work correctly.

Manage DLP policies Information to help you view, change, or remove existing data loss prevention policies.

Note:
Data Loss Prevention is a premium feature that requires an Enterprise Client Access License (CAL).

For more information

Manage policy tips

Create incident reports for DLP policy detections

Learn more about modes for DLP policies and rules

Integrating sensitive information rules with transport rules

Exchange Server 2013 > Messaging policy and compliance > Data loss prevention >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-16*

In Microsoft Exchange Server 2013, you can create DLP policies that contain rules for not only traditional message classifications and existing transport rules but also combine these with rules for sensitive information found within messages. The existing transport rules framework offers rich capabilities to define messaging policies, covering the entire spectrum of soft to hard controls. Examples include:

- Limiting the interaction between recipients and senders, including interactions between departmental groups inside an organization.
- Applying separate policies for communications within and outside of an organization.
- Preventing inappropriate content from entering or leaving an organization.
- Filtering confidential information.
- Tracking or archiving messages that are sent to or received from specific individuals.
- Redirecting inbound and outbound messages for inspection before delivery.
- Applying disclaimers to messages as they pass through the organization.

Transport rules allow you to apply messaging policies to email messages that flow through the transport pipeline in the Transport service on Mailbox servers and on Edge Transport servers. These rules allow system administrators to enforce messaging policies, help keep messages more secure,

help to protect messaging systems, and help prevent accidental information loss. For more information about transport rules, see Transport rules.

Sensitive information rules within the transport rule framework

Sensitive information rules are integrated with the transport rules framework by introduction of a condition that you can customize: **If the message contains…Sensitive Information**. This condition can be configured with one or more sensitive information types that are contained within the messages. When multiple DLP policies or rules within a policy are configured with this condition, the policy or rule is satisfied when any of the conditions match. Exchange 2013 policy rules examine the subject, body and any attachments of a message. If the rule matches any of these message components, the rule actions will be applied.

The sensitive information condition may be combined with any of the already existing transport rules to define messaging policies. If combined, the condition works in conjunction with other rules and provides the AND semantics. For example, two different conditions are added together with an AND statement such that both need to match for the action to be applied. Any of the transport rule actions can be configured as result of rules containing the sensitive information type matching. Many different file types can be scanned by the transport rules agent, which scans messages to enforce transport rules. To learn more about the supported file types, see Using transport rules to inspect message attachments.

The rules can also be used in the exception part of a rule definition. Their use in the exception definition is independent of their use as a condition within the rule. This provides the flexibility to define rules that have the condition specifying multiple information types to be applied as part of the condition and also differing information types in the condition. This would allow policies such as matching specific traditional message-classification rules, but not matching other sensitive information types before performing actions that you define within a policy.

For more information

Data loss prevention

Sensitive information types inventory

Transport rules

Create a custom DLP policy

Information Rights Management

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-11-01*

Every day, information workers use e-mail to exchange sensitive information such as financial reports and data, legal contracts, confidential product information, sales reports and projections, competitive analysis, research and patent information, and customer and employee information. Because people can now access their e-mail from just about anywhere, mailboxes have transformed into repositories containing large amounts of potentially sensitive information. As a result, information leakage can be a serious threat to organizations. To help prevent information leakage, Microsoft Exchange Server 2013 includes Information Rights Management (IRM) features, which provide persistent online and offline protection of e-mail messages and attachments.

Contents

What is information leakage?

Leakage of potentially sensitive information can be costly for an organization and have wide-ranging impact on the organization and its business, employees, customers, and partners. Local and industry regulations increasingly govern how certain types of information are stored, transmitted, and secured. To avoid violating applicable regulations, organizations must protect themselves against intentional, inadvertent, or accidental information leakage.

The following are some consequences resulting from information leakage:

- **Financial damage** Depending on the size, industry, and local regulations, information leakage may result in financial impact due to loss of business or due to fines and punitive damages imposed by courts or regulatory authorities. Public companies may also risk losing market capitalization due to adverse media coverage.

- **Damage to image and credibility** Information leakage can damage an organization's image and credibility with customers. Moreover, depending on the nature of communication, leaked e-mail messages can potentially be a source of embarrassment for the sender and the organization.
- **Loss of competitive advantage** One of the most serious threats from information leakage is the loss of competitive advantage in business. Disclosure of strategic plans or disclosure of merger and acquisition information can potentially lead to loss of revenue or market capitalization. Other threats include loss of research information, analytical data, and other intellectual property.

Return to top

Traditional solutions to information leakage

Although traditional solutions to information leakage may protect initial access to data, they often don't provide constant protection. The following table lists some traditional solutions and their limitations.

Traditional solutions

Solution	Description	Limitations
Transport Layer Security (TLS)	TLS is an Internet standard protocol used to secure communications over a network by means of encryption. In a messaging environment, TLS is used to secure server/server and client/server communications. By default, Exchange 2010 uses TLS for all internal message transfers. Opportunistic TLS is also enabled by default for sessions with external hosts. Exchange first attempts to use TLS encryption for the session, but if a TLS connection can't be established with the destination server, Exchange uses SMTP. You can also	TLS only protects the SMTP session between two SMTP hosts. In other words, TLS protects information in motion, and it doesn't provide protection at the message-level or for information at rest. Unless the messages are encrypted using another method, messages in the sender's and recipients' mailboxes remain unprotected. For e-mail sent outside the organization, you can require TLS only for the first hop. After a remote SMTP host outside your organization receives the message, it can relay it to another SMTP host over an

	configure domain security to enforce mutual TLS with external organizations.	unencrypted session. Because TLS is a transport layer technology, it can't provide control over what the recipient does with the message.
E-mail encryption	Users can use technologies such as S/MIME to encrypt messages.	Users decide whether a message gets encrypted. There are additional costs of a public key infrastructure (PKI) deployment, with the accompanying overhead of certificate management for users and protection of private keys. After a message is decrypted, there's no control over what the recipient can do with the information. Decrypted information can be copied, printed, or forwarded. By default, saved attachments aren't protected. Messages encrypted using technologies such as S/MIME can't be accessed by your organization. The organization can't inspect message content, and therefore can't enforce messaging policies, scan messages for viruses or malicious content, or take any other action that requires accessing the content.

Finally, traditional solutions often lack enforcement tools that apply uniform messaging policies to prevent information leakage. For example, a user sends a message containing sensitive information and marks it as **Company Confidential** and **Do Not Forward**. After the message is delivered to the recipient, the sender or the organization no longer has control over the information. The recipient can willfully or inadvertently forward the message (using features such as automatic forwarding rules) to external e-mail accounts, subjecting your organization to substantial information leakage risks.

Return to top

IRM in Exchange 2013

In Exchange 2013, you can use IRM features to apply persistent protection to messages and attachments. IRM uses Active Directory Rights Management Services (AD RMS), an information protection technology in Windows Server 2008 and later. With the IRM features in Exchange 2013, your organization and your users can control the rights recipients have for e-mail. IRM also helps allow or restrict recipient actions such as forwarding a message to other recipients, printing a message or attachment, or extracting message or attachment content by copying and pasting. IRM protection can be applied by users in Microsoft Outlook or Microsoft Office Outlook Web App, or it can be based on your organization's messaging policies and applied using transport protection rules or Outlook protection rules. Unlike other e-mail encryption solutions, IRM also allows your organization to decrypt protected content to enforce policy compliance.

AD RMS uses extensible rights markup language (XrML)-based certificates and licenses to certify computers and users, and to protect content. When content such as a document or a message is protected using AD RMS, an XrML license containing the rights that authorized users have to the content is attached. To access IRM-protected content, AD RMS-enabled applications must procure a use license for the authorized user from the AD RMS cluster.

Note:

In Exchange 2013, the Prelicensing agent attaches a use license to messages protected using the AD RMS cluster in your organization. For more details, see Prelicensing later in this topic.

Applications used to create content must be RMS-enabled to apply persistent protection to content using AD RMS. Microsoft Office applications, such as Word, Excel, PowerPoint and Outlook are RMS-enabled and can be used to create and consume protected content.

IRM helps you do the following:

- Prevent an authorized recipient of IRM-protected content from forwarding, modifying, printing, faxing, saving, or cutting and pasting the content.
- Protect supported attachment file formats with the same level of protection as the message.
- Support expiration of IRM-protected messages and attachments so they can no longer be viewed after the specified period.
- Prevent IRM-protected content from being copied using the Snipping Tool in Microsoft Windows.

However, IRM can't prevent information from being copied using the following methods:

- Third-party screen capture programs
- Use of imaging devices such as cameras to photograph IRM-protected content displayed on the screen
- Users remembering or manually transcribing the information

To learn more about AD RMS, see Active Directory Rights Management Services.

AD RMS rights policy templates

AD RMS uses XrML-based rights policy templates to allow compatible IRM-enabled applications to apply consistent protection policies. In Windows Server 2008 and later, the AD RMS server exposes a Web service that can be used to enumerate and acquire templates. Exchange 2013 ships with the Do Not Forward template. When the Do Not Forward template is applied to a message, only the recipients addressed in the message can decrypt the message. The recipients can't forward the message, copy content from the message, or print the message. You can create additional RMS templates on the AD RMS server in your organization to meet your IRM protection requirements.

IRM protection is applied by applying an AD RMS rights policy template. Using policy templates, you can control permissions that recipients have on a message. Actions such as replying, replying to all, forwarding, extracting information from a message, saving a message, or printing a message can be controlled by applying the appropriate rights policy template to the message.

For more information about rights policy templates, see AD RMS Policy Template Considerations.

For more information about creating AD RMS rights policy templates, see AD RMS Rights Policy Templates Deployment Step-by-Step Guide.

Return to top

Applying IRM protection to messages

In Exchange 2010, IRM protection can be applied to messages using the following methods:

- **Manually by Outlook users** Your Outlook users can IRM-protect messages with the AD RMS rights policy templates available to them. This process uses the IRM functionality in Outlook, and not Exchange. However, you can use Exchange to access messages, and you can take actions (such as applying transport rules) to enforce your organization's messaging policy. For more information about using IRM in Outlook, see Introduction to using IRM for e-mail messages.
- **Manually by Outlook Web App users** When you enable IRM in Outlook Web App, users can IRM-protect messages they send, and view IRM-protected messages they receive. In Exchange 2013 Cumulative Update 1 (CU1), Outlook Web App users can also view IRM-protected attachments using Web-Ready Document Viewing. For more information about IRM in Outlook Web App, see Information Rights Management in Outlook Web App.
- **Manually by Windows Mobile and Exchange ActiveSync device users** In the release to manufacturing (RTM) version of Exchange 2010, users of Windows Mobile devices can view and create IRM-protected messages. This requires users to connect their supported Windows Mobile

devices to a computer and activate them for IRM. In Exchange 2010 SP1, you can enable IRM in Microsoft Exchange ActiveSync to allow users of Exchange ActiveSync devices (including Windows Mobile devices) to view, reply to, forward, and create IRM-protected messages. For more information about IRM in Exchange ActiveSync, see Information Rights Management in Exchange ActiveSync.

- **Automatically in Outlook 2010 and later** You can create Outlook protection rules to automatically IRM-protect messages in Outlook 2010 and later. Outlook protection rules are deployed automatically to Outlook 2010 clients, and IRM-protection is applied by Outlook 2010 when the user is composing a message. For more information about Outlook protection rules, see Outlook protection rules.
- **Automatically on Mailbox servers** You can create transport protection rules to automatically IRM-protect messages on Exchange 2013 Mailbox servers. For more information about transport protection rules, see Transport protection rules.

Note:
IRM protection isn't applied again to messages that are already IRM-protected. For example, if a user IRM-protects a message in Outlook or Outlook Web App, IRM protection isn't applied to the message using a transport protection rule.

Return to top

Scenarios for IRM protection

Scenarios for IRM protection are described in the following table.

Scenarios for IRM protection

Sending IRM-protected messages	Supported	Requirements
Within the same on-premises Exchange 2013 deployment	Yes	For requirements, see IRM Requirements later in this topic.
Between different forests in an on-premises deployment	Yes	For requirements, see Configuring AD RMS to Integrate with Exchange Server 2010 Across Multiple Forests.
Between an on-premises Exchange 2013 deployment and a cloud-based Exchange organization	Yes	• Use an on-premises AD RMS server. • Export the trusted publishing domain from your on-premises AD RMS server.

		• Import the trusted publishing domain in your cloud-based organization.
To external recipients	No	Exchange 2010 doesn't include a solution for sending IRM-protected messages to external recipients in a non-federated organization. AD RMS offers solutions using trust policies. You can configure a trust policy between your AD RMS cluster and Microsoft account (formerly known as Windows Live ID). For messages sent between two organizations, you can create a federated trust between the two Active Directory forests using Active Directory Federation Services (AD FS). To learn more, see Understanding AD RMS Trust Policies.

Return to top

Decrypting IRM-protected messages to enforce messaging policies

To enforce messaging policies and for regulatory compliance, you must be able to access encrypted message content. To meet eDiscovery requirements due to litigation, regulatory audits, or internal investigations, you must also be able to search encrypted messages. To help with these tasks, Exchange 2013 includes the following IRM features:

- **Transport decryption** To apply messaging policies, transport agents such as the Transport Rules agent should have access to message content. Transport decryption allows transport agents installed on Exchange 2013 servers to access message content. For more information, see Transport decryption.

- **Journal report decryption** To meet compliance or business requirements, organizations can use journaling to preserve messaging content. The Journaling agent creates a journal report for messages subject to journaling and includes metadata about the message in the report. The original message is attached to the journal report. If the message in a journal report is IRM-protected, journal report decryption attaches a cleartext copy of the message to the journal report. For more information, see Journal report decryption.
- **IRM decryption for Exchange Search** With IRM decryption for Exchange Search, Exchange Search can index content in IRM-protected messages. When a discovery manager performs an In-Place eDiscovery search, IRM-protected messages that have been indexed are returned in search results. For more information, see In-Place eDiscovery.

Note:

In Exchange 2010 SP1 and later, members of the Discovery Management role group can access IRM-protected messages returned by a discovery search and residing in a discovery mailbox. To enable this functionality, use the *EDiscoverySuperUserEnabled* parameter with Set-IRMConfiguration cmdlet. For more information, see Configure IRM for Exchange Search and In-Place eDiscovery.

To enable these decryption features, Exchange servers must have access to the message. This is accomplished by adding the Federation mailbox, a system mailbox created by Exchange Setup, to the super users group on the AD RMS server. For details, see Add the Federation Mailbox to the AD RMS Super Users Group.

Return to top

Prelicensing

To view IRM-protected messages and attachments, Exchange 2013 automatically attaches a prelicense to protected messages. This prevents the client from having to make repeated trips to the AD RMS server to retrieve a use license, and enables offline viewing of IRM-protected messages and attachments. Prelicensing also allows IRM-protected messages to be viewed in Outlook Web App. When you enable IRM features, prelicensing is enabled by default.

Return to top

IRM agents

In Exchange 2013, IRM functionality is enabled using transport agents in the Transport service on Mailbox servers. IRM agents are installed by Exchange Setup on a Mailbox server. You can't control IRM agents using the management tasks for transport agents.

Note:

In Exchange 2013, IRM agents are built-in agents. Built-in agents aren't included in the list of agents returned by the **Get-TransportAgent** cmdlet. For more information, see Transport agents.

The following table lists the IRM agents implemented in the Transport service on Mailbox servers.

IRM agents in the Transport service on Mailbox servers

Agent	Event	Function
RMS Decryption agent	OnEndOfData (SMTP) and OnSubmittedMessage	Decrypts messages to allow access to transport agents.
Transport Rules agent	OnRoutedMessage	Flags messages that match rule conditions in a transport protection rule to be IRM-protected by the RMS Encryption agent.
RMS Encryption agent	OnRoutedMessage	Applies IRM protection to messages flagged by the Transport Rules agent and re-encrypts transport decrypted messages.
Prelicensing agent	OnRoutedMessage	Attaches a prelicense to IRM-protected messages.
Journal Report Decryption agent	OnCategorizedMessage	Decrypts IRM-protected messages attached to journal reports and embeds cleartext versions along with the original encrypted messages.

For more information about transport agents, see Transport agents.

Return to top

IRM requirements

To implement IRM in your Exchange 2013 organization, your deployment must meet the requirements described in the following table.

IRM requirements

Server	Requirements

AD RMS cluster	• **Operating system** Windows Server 2012, Windows Server 2008 R2 or Windows Server 2008 SP2 with the hotfix Active Directory Rights Management Services role in Windows Server 2008 is required. • **Service connection point** Exchange 2010 and AD RMS-aware applications use the service connection point registered in Active Directory to discover an AD RMS cluster and URLs. AD RMS allows you to register the service connection point from within AD RMS Setup. If the account used to set up AD RMS isn't a member of the Enterprise Admins security group, service connection point registration can be performed after setup is complete. There is only one service connection point for AD RMS in an Active Directory forest. • **Permissions** Read and Execute permissions to the AD RMS server certification pipeline (`ServerCertification.asmx` file on AD RMS servers) must be assigned to the following: ○ Exchange Servers group or individual Exchange servers ○ AD RMS Service group on AD RMS servers By default, the ServerCertification.asmx file is located in the `\inetpub\wwwroot\_wmcs\certification\` folder on AD RMS servers. For details, see Set Permissions on the AD RMS Server Certification Pipeline. • **AD RMS super users** To enable transport decryption, journal report decryption, IRM in Outlook Web App, and IRM for Exchange Search, you must add the Federation mailbox, a system mailbox created by Exchange 2013

	Setup, to the super users group on the AD RMS cluster. For details, see Add the Federation Mailbox to the AD RMS Super Users Group.
Exchange	• Exchange 2010 or later is required. • The hotfix FIX: ArgumentNullException exception error message when a .NET Framework 2.0 SP2-based application tries to process a response with zero-length content to an asynchronous ASP.NET Web service request: "Value cannot be null" is recommended for Microsoft .NET Framework 2.0 SP2.
Outlook	• Users can IRM-protect messages in Outlook. Beginning with Outlook 2003, AD RMS templates for IRM-protecting messages is supported. • Outlook protection rules are an Exchange 2010 and Outlook 2010 feature. Previous versions of Outlook don't support this feature.
Exchange ActiveSync	• Devices supporting Exchange ActiveSync protocol version 14.1, including Windows Mobile devices, can support IRM in Exchange ActiveSync. The mobile e-mail application on a device must support the **RightsManagementInformation** tag defined in Exchange ActiveSync protocol version 14.1. In Exchange 2013, IRM in Exchange ActiveSync allows users with supported devices to view, reply to, forward, and create IRM-protected messages without requiring the user to connect the device to a computer and activate it for IRM. For details, see Information Rights Management in Exchange ActiveSync.

Note:

AD RMS cluster is the term used for an AD RMS deployment in an organization, including a

single server deployment. AD RMS is a Web service. It doesn't require that you set up a Windows Server failover cluster. For high availability and load-balancing, you can deploy multiple AD RMS servers in the cluster and use Network Load Balancing.

Important:
In a production environment, installing AD RMS and Exchange on the same server isn't supported.

Exchange 2013 IRM features support Microsoft Office file formats. You can extend IRM protection to other file formats by deploying custom protectors. For more information about custom protectors, see Information Protection and Control Partners in Independent Software Vendors.

Return to top

Configuring and testing IRM

You must use the Exchange Management Shell to configure IRM features in Exchange 2013. To configure individual IRM features, use the Set-IRMConfiguration cmdlet. You can enable or disable IRM for internal messages, transport decryption, journal report decryption, Exchange Search, and Outlook Web App. For more information about configuring IRM features, see Information Rights Management procedures.

After you set up an Exchange 2013 server, you can use the Test-IRMConfiguration cmdlet to perform end-to-end tests of your IRM deployment. These tests are useful to verify IRM functionality immediately after initial IRM configuration and on an ongoing basis. The cmdlet performs the following tests:

- Inspects IRM configuration for your Exchange 2013 organization.
- Checks the AD RMS server for version and hotfix information.
- Verifies whether an Exchange server can be activated for RMS by retrieving a Rights Account Certificate (RAC) and client licensor certificate.
- Acquires AD RMS rights policy templates from the AD RMS server.
- Verifies that the specified sender can send IRM-protected messages.
- Retrieves a super user use license for the specified recipient.
- Acquires a prelicense for the specified recipient.

Return to top

Extend Rights Management with the Rights Management connector

The Microsoft Rights Management connector (RMS connector) is an optional application that enhances data protection for your Exchange 2013 server by employing cloud-based Microsoft Rights Management services. Once you install the RMS connector, it provides continuous data

protection during the lifespan of the information and because these services are customizable, you can define the level of protection you need. For example, you can limit email message access to specific users or set view-only rights for certain messages.

To learn more about the RMS connector and how to install it, see Rights Management connector.

Transport protection rules

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-13*

Email messages and attachments increasingly contain business critical information such as product specifications, business strategy documents, and financial data, or personally identifiable information (PII) such as contact details, social security numbers, credit card numbers, and employee records. There are a number of industry-specific and local regulations in many parts of the world that govern the collection, storage, and disclosure of PII.

To help protect sensitive information, organizations create messaging policies that provide guidelines about how to handle this information. In Microsoft Exchange Server 2013, you can use transport protection rules to implement these messaging policies by inspecting message content, encrypting sensitive email content, and using rights management to control access to the content.

For management tasks related to managing IRM, see Information Rights Management procedures.

Transport protection rules and AD RMS

Transport protection rules allow you to use transport rules to IRM-protect messages by applying an Active Directory Rights Management Services (AD RMS) rights policy template.

Note:
AD RMS is an information protection technology that works with Rights Management Service (RMS)-enabled applications and clients to protect sensitive information online and offline. To use IRM protection in an on-premise Exchange deployment, Exchange 2013 requires an on-premises deployment of AD RMS running on Windows Server 2008 or later.

AD RMS uses XML-based policy templates to allow compatible IRM-enabled applications to apply consistent protection policies. In Windows Server 2008 and later, the AD RMS server exposes a Web service that can be used to enumerate and acquire templates. Exchange 2013 ships with the Do Not Forward template.

When the Do Not Forward template is applied to a message, only the recipients addressed in the message can decrypt the message. The recipients can't forward the message to anyone else, copy

content from the message, or print the message.

Additional RMS templates can be created in the on-premises AD RMS deployment to meet rights protection requirements in your organization.

Important:
If a rights policy template is removed from the AD RMS server, you must modify any transport protection rules that use the removed template. If a transport protection rule continues to use a rights policy template that's been removed, the AD RMS server will fail to license the content to any of the recipients, and a non-delivery report (NDR) will be delivered to the sender. In Windows Server 2008 and later, rights policy templates can be archived instead of deleted. Archived templates can still be used to license content, but when you create or modify a transport protection rule, archived templates aren't included in the list of templates.

For more information about creating AD RMS templates, see AD RMS Rights Policy Templates Deployment Step-by-Step Guide.

Automatic protection using transport protection rules

Messages containing business critical information or PII can be identified by using a combination of transport rule conditions, including regular expressions to identify text patterns such as social security numbers. Organizations require different levels of protection for sensitive information. Some information may be restricted to employees, contractors, or partners; while other information may be restricted only to full-time employees. The desired level of protection can be applied to messages by applying an appropriate rights policy template. For example, users may mark messages or email attachments as Company Confidential. As illustrated in the following figure, you can create a transport protection rule to inspect message content for the words "Company Confidential", and automatically IRM-protect the message.

For more information about creating transport rules to enforce rights protection, see Create a Transport Protection Rule.

Persistent protection of email attachments

Users send business critical information and PII in email attachments using common Microsoft Office file formats such as Microsoft Office Word, Excel, and PowerPoint. All of these file formats support persistent protection via IRM, and you can make sure that the business critical information and PII in these documents are properly protected. Transport protection rules apply the same protection to email messages and attachments in supported file formats.

Transport rules agent and encryption agent

When you use transport protection rules to IRM-protect messages based on rule conditions, the Transport Rules agent on the Transport service inspects messages. If they meet all the conditions and none of the exceptions, it flags the message to be IRM-protected. The Encryption agent, a

built-in transport agent that fires on the **OnRoutedMessage** event, actually applies IRM protection to the message. The Encryption agent acts on messages only if IRM is enabled for internal messages. For more information about enabling IRM, see Enable or Disable IRM for Internal Messages.

When the transport service is restarted, and it processes the first message that requires IRM encryption, the Encryption agent must be able to reach an AD RMS server in the organization. For subsequent messages, the agent doesn't need to contact the AD RMS server. Upon failure to encrypt a message due to transient conditions, Exchange retries the message three times at 10-minute intervals. After three retries, if the message can't be encrypted, it isn't delivered to recipients. An NDR is sent to the sender. We recommend that you plan your AD RMS deployment for high availability to make sure message flow isn't impacted.

When planning to use transport protection rules, you must consider the type of information you want to protect and plan on creating rules accordingly. In Exchange 2013, transport rules have a large number of predicates that allow you to inspect message content, including supported attachments, message headers, sender and recipient addresses, their Active Directory attributes such as department, distribution group membership, and management relationships of the sender with recipients. For more details about transport rule predicates available in Exchange 2013, see Transport rule conditions (predicates).

You must also consider the messaging traffic in your organization, and the number of messages that will be protected using transport protection rules. Applying IRM protection to a large number of messages requires more resources on the Mailbox server. Additionally, protecting a large number of messages or all messages also impacts the client experience, particularly for Microsoft Outlook users.

Outlook protection rules

Exchange Server 2013 > Messaging policy and compliance > Information Rights Management >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-13*

Every day, information workers exchange sensitive information by email, including financial reports and data, customer and employee information, and confidential product information and specifications. In Microsoft Exchange Server 2013, Microsoft Outlook, and Microsoft Office Outlook Web App, users can apply Information Rights Management (IRM) protection to messages by applying an Active Directory Rights Management Services (AD RMS) rights policy template. This requires an AD RMS deployment in the organization. For more information about AD RMS, see Active Directory Rights Management Services.

However, when left to the discretion of users, messages may be sent in clear text without IRM protection. In organizations that use email as a hosted service, there's a risk of information leakage as a message leaves the client and is routed and stored outside the boundaries of an organization. Although email hosting companies may have well-defined procedures and checks to help mitigate the risk of information leakage, after a message leaves the boundary of an organization, the organization loses control of the information. Outlook protection rules can help protect against this type of information leakage.

For management tasks related to managing IRM, see Information Rights Management procedures.

Automatic IRM protection in Outlook

In Exchange 2013, Outlook protection rules help your organization protect against the risk of information leakage by automatically applying IRM-protection to messages in Exchange 2013. Messages are IRM-protected before they leave the Outlook client. This protection is also applied to any attachments using supported file formats.

When you create Outlook protection rules on an Exchange 2013 server, the rules are automatically distributed to Outlook 2010 by using Exchange Web Services. For Outlook 2010 to apply the rule, the AD RMS rights policy template you specify must be available on users' computers.

Important:

If a rights policy template is removed from the AD RMS server, you must modify any Outlook protection rules that use the removed template. If an Outlook protection rule continues to use a rights policy template that's been removed, and transport decryption is enabled in the organization, the Decryption agent will fail to decrypt the message protected with a template that's no longer available. If transport decryption is configured as mandatory, the Transport service will reject the message and send a non-delivery report (NDR) to the sender. For more details about transport decryption, see Transport decryption. For more details about AD RMS rights policy templates, see AD RMS Policy Template Considerations.
In Windows Server 2008 and later, rights policy templates can be archived instead of deleted. Archived templates can still be used to license content, but when you create or modify an Outlook protection rule, archived templates aren't included in the list of templates.

Outlook protection rules are similar to transport protection rules. Both are applied based on message conditions, and both protect messages by applying an AD RMS rights protection template. However, transport protection rules are applied in the Transport service on the Mailbox server by the Transport Rules agent. Outlook protection rules are applied in Outlook 2010, before the message leaves the user's computer. Messages protected by an Outlook protection rule enter the transport pipeline with IRM protection already applied. Additionally, messages protected with an Outlook protection rule are also saved in an encrypted format in the Sent Items folder of the sender's mailbox.

Note:

If transport decryption is enabled in your Exchange organization, messages that are IRM-protected by an Outlook protection rule using the AD RMS server in your organization can be

decrypted by the Decryption agent on Transport service. Message content can be inspected by the Transport Rules agent and other transport agents installed on the Transport service. For more details about transport decryption, see Transport decryption.

When you use transport protection rules, users have no indication of whether a message is going to be automatically protected on the Transport service. When an Outlook protection rule is applied to a message in Outlook 2010, users know if a message will be IRM-protected. If required, users can also select a different rights policy template.

Creating Outlook protection rules

To create Outlook protection rules, you must use the New-OutlookProtectionRule cmdlet in the Exchange Management Shell. For detailed instructions, see Create an Outlook Protection Rule.

When creating a rule, you can specify whether the user can override it, either by removing IRM-protection or by applying a different AD RMS rights policy template than the one specified in the rule. If a user overrides the IRM protection applied by an Outlook protection rule, Outlook 2010 inserts the `X-MS-Outlook-Client-Rule-Overridden` header in the message, which allows you to determine that the rule was overridden by the user.

Predicates in Outlook protection rules

Outlook protection rules allow you to use three predicates to automatically apply IRM protection in Outlook 2010:

- **FromDepartment** The *FromDepartment* predicate looks up the sender's department attribute in Active Directory and automatically IRM-protects the message if the sender's department matches the department specified in the rule. For example, you can create an Outlook protection rule to automatically protect all messages sent by the Research department.
- **SentTo** Your organization may need to protect messages sent to certain sensitive recipients, such as the All Company or Finance distribution groups. Using the *SentTo* predicate, you can create an Outlook protection rule to automatically IRM-protect messages sent to specified recipients.
- **SentToScope** The *SentToScope* predicate allows you to create an Outlook protection rule to automatically IRM-protect messages sent inside or outside the organization. For example, you can use the *SentToScope* predicate with the *FromDepartment* predicate to IRM-protect messages sent by a particular department to internal users.

Transport decryption

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-13*

In Microsoft Exchange Server 2013, Microsoft Outlook 2010 and later, and Microsoft Office Outlook Web App, users can use Information Rights Management (IRM) to protect their messages. You can create Outlook protection rules to automatically apply IRM protection to messages before they're sent from an Outlook 2010 client. You can also create transport protection rules to apply IRM protection to messages in transit that match the rule conditions. Transport decryption allows access to IRM-protected messaging content to enforce messaging policies.

For management tasks related to managing IRM, see Information Rights Management procedures.

Limitations of other encryption solutions

If it's critical that your organization protects sensitive information, including high business impact (HBI) information and personally identifiable information (PII), consider encrypting e-mail messages and attachments. E-mail encryption solutions such as S/MIME have been available for a long time. These encryption solutions have seen varying degrees of adoption in organizations of different types. However, such solutions present the following challenges:

- **Inability to apply messaging policies** Organizations also face compliance requirements that require inspection of messaging content to make sure it adheres to messaging policies. However, messages encrypted with most client-based encryption solutions, including S/MIME, prevent content inspection on the server. Without content inspection, an organization can't validate that all messages sent or received by its users comply with messaging policies. For example, to comply with a legal regulation, you've configured a transport rule to detect PII, such as a social security number, and automatically apply a disclaimer to the message. If the message is encrypted, the Transport Rules agent on the Transport service can't access message content, and therefore won't apply the disclaimer. This results in a violation of the policy.
- **Decreased security** Antivirus software is unable to scan encrypted message content, further exposing an organization to risk from malicious content such as viruses and worms. Encrypted messages are generally considered to be trusted by most users, thereby increasing the likelihood of a virus spreading throughout your organization. For example, you've configured an Outlook protection rule to automatically apply IRM protection to all messages sent to the All Employees distribution list with the Company Confidential rights management service (RMS) template. A user's workstation is infected with a virus that propagates by automatically using Reply All to reply to messages. If the message carrying the virus is encrypted, the antivirus scanner can't scan the message.
- **Impact to custom transport agents** Many organizations develop custom transport agents for different purposes, such as meeting additional processing requirements for compliance, security, or custom message routing. Custom transport agents developed by an organization to inspect or modify messages are unable to process encrypted messages. If the custom transport agents developed by your organization can't access message content, message encryption may prevent your organization from meeting the goals for which the custom transport agents are developed.

Using transport decryption for encrypted content

In Exchange 2013, IRM features address these challenges. If messages are IRM-protected, transport decryption allows you to decrypt them in transit. IRM-protected messages are decrypted by the Decryption agent, a compliance-focused transport agent.

Note:

In Exchange 2013, the Decryption agent is a built-in agent. Built-in agents aren't included in the list of agents returned by the **Get-TransportAgent** cmdlet. For more details, see Transport agents.

The Decryption agent decrypts the following types of IRM-protected messages:

- Messages IRM-protected by the user in Outlook Web App.
- Messages IRM-protected by the user in Outlook 2010.
- Messages IRM-protected automatically by Outlook protection rules in Exchange 2013 and Outlook 2010.

Important:

Only messages IRM-protected by the AD RMS server in your organization are decrypted by the Decryption agent.

Note:

Messages protected in-transit using transport protection rules aren't required to be decrypted by the Decryption agent. The Decryption agent fires on the **OnEndOfData** and **OnSubmit** transport events. Transport protection rules are applied by the Transport Rules agent, which fires on the **OnRoutedMessage** event, and IRM-protection is applied by the Encryption agent on the **OnRoutedMessage** event. For more information about transport agents and a list of SMTP events on which they can be registered, see Transport agents.

Transport decryption is performed on the first Exchange 2013 Transport service that handles a message in an Active Directory forest. If a message is transferred to a Transport service in another Active Directory forest, the message is decrypted again. After decryption, unencrypted content is available to other transport agents on that server. For example, the Transport Rules agent on a Transport service can inspect message content and apply transport rules. Any actions specified in the rule, such as applying a disclaimer or modifying the message in any other way, can be taken on the unencrypted message. Third-party transport agents, such as antivirus scanners, can scan the message for viruses and malware. After other transport agents have inspected the message and possibly made modifications to it, it's encrypted again with the same user rights that it had before being decrypted by the Decryption agent. The same message isn't decrypted again by other the Transport service on other Mailbox servers in the organization.

Messages decrypted by the Decryption agent don't leave the Transport service without being encrypted again. If a transient error is returned when decrypting or encrypting the message, the Transport service retries the operation twice. After the third failure, the error is treated as a permanent error. If any permanent errors occur, including when transient errors are treated as permanent errors after retries, the Transport service treats them as follows:

- If the permanent error occurs during decryption, a non-delivery report (NDR) is sent only if transport decryption is set to `Mandatory`, and the encrypted message is sent with the NDR. For more details about the configuration options available for transport decryption, see Configuring Transport Decryption later in this topic.
- If the permanent error occurs during re-encryption, an NDR is always sent without the decrypted message.

Important:

Any custom or third-party agents installed on a Transport service have access to the decrypted message. You must consider the behavior of such transport agents. We recommend that you test all custom and third-party transport agents thoroughly before you deploy them in a production environment.
After a message is decrypted by the Decryption agent, if a transport agent creates a new message and embeds (attaches) the original message to the new one, only the new message is protected. The original message, which becomes an attachment to the new message, doesn't get re-encrypted. A recipient receiving such a message can open the attached message and take actions such as forwarding or replying, which would bypass rights enforcement.

Configuring transport decryption

Transport decryption is configured by using the Set-IRMConfiguration cmdlet in the Exchange Management Shell. However, before you configure transport decryption, you must provide Exchange 2013 servers the right to decrypt content protected by your AD RMS server. This is done by adding the Federation mailbox to the super users group configured on the AD RMS cluster in your organization.

Important:

In cross-forest AD RMS deployments where you have an AD RMS cluster deployed in each forest, you must add the Federation mailbox to the super users group on the AD RMS cluster in each forest to allow the Transport service on an Exchange 2013 Mailbox server or an Exchange 2010 Hub Transport server to decrypt the messages protected against each AD RMS cluster.

For details, see Add the Federation Mailbox to the AD RMS Super Users Group.

Exchange 2013 allows two different settings when enabling transport decryption:

- **Mandatory** When transport decryption is set to `Mandatory`, the Decryption agent rejects the message and returns an NDR to the sender if a permanent error is returned when decrypting a message. If your organization doesn't want a message to be delivered if it can't be successfully decrypted and actions such as antivirus scanning and transport rules are applied, you must choose this setting.
- **Optional** When transport decryption is set to Optional, the Decryption agent uses a best-effort approach. Messages that can be decrypted are decrypted, but messages with a permanent error on decryption are also delivered. If your organization prioritizes message delivery over messaging policy, you must use this setting.

For more information about configuring transport decryption, see Enable or Disable Transport

Decryption.

Journal report decryption

Exchange Server 2013 > Messaging policy and compliance > Information Rights Management >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

In Microsoft Exchange Server 2013, Information Rights Management (IRM) allows Microsoft Outlook 2010 and later and Microsoft Office Outlook Web App users to protect their messages. You can create Outlook protection rules to automatically apply IRM-protection to messages before they're sent from an Outlook 2010 client. You can also create transport protection rules to apply IRM protection to messages in transit that match the rule conditions.

To learn about Outlook protection rules, see Outlook protection rules.

Limitations of standard encryption solutions

If your organization encrypts messages by using traditional solutions such as S/MIME, your records managers won't be able to inspect or search the encrypted content. Archiving encrypted messages that contain inaccessible and unsearchable content may not meet business, regulatory, or compliance requirements. When faced with an electronic discovery (eDiscovery) request, an inability to decrypt, search, and present content from encrypted messages can be a challenge, and failure to do so may expose your organization to legal and financial risks.

Also, your organization's messaging policies may require journaled messages to be decrypted so the content can be accessible to eDiscovery tools, automated processes, or records managers who access a journaling mailbox. Journal report decryption in Exchange 2010 can help you meet these requirements.

To learn more about journaling, see Journaling.

Journal report decryption

Journal report decryption allows you to save a clear-text copy of IRM-protected messages in journal reports, along with the original, IRM-protected message. If the IRM-protected message contains any supported attachments that were protected by the Active Directory Rights Management Services (AD RMS) cluster in your organization, the attachments are also decrypted.

Important:
To use journal report decryption, you must have an Exchange Enterprise client access license

(CAL). Journal report decryption only supports premium journaling.

Decryption is performed by the Journal Report Decryption agent, a compliance-focused transport agent. The Journal Report Decryption agent fires on the **OnCategorizedMessage** event. Messages protected in-transit using transport protection rules are already encrypted by the Encryption agent, which fires on the **OnRoutedMessage** event, before they get to the Journal Report Decryption agent. The Journal Report Decryption agent decrypts these messages.

Note:

In Exchange 2013, the Journal Report Decryption agent is a built-in agent. Built-in agents aren't included in the list of agents returned by the **Get-TransportAgent** cmdlet. For more details, see Transport agents.

The agent decrypts the following types of IRM-protected messages:

- Messages that were IRM-protected by the user in Outlook Web App.
- Messages that were IRM-protected by the user in Outlook 2010.
- Messages that were IRM-protected automatically in Outlook 2010 by using Outlook protection rules.
- Messages that were IRM-protected automatically in transit by using transport protection rules.

Important:

Only messages that were IRM-protected by the AD RMS server in your organization are decrypted by the Journal Report Decryption agent. The agent doesn't decrypt an attachment if it isn't protected at the same time as the message (and therefore doesn't have the same use license), or if an IRM-protected file is attached to an unprotected message.

Configuring journal report decryption

Journal report decryption is configuredb using the Set-IRMConfiguration cmdlet in the Exchange Management Shell. However, before you configure journal report decryption, you must assign Exchange 2013 servers the permissions to decrypt content that's IRM-protected by your AD RMS server. To do this, you add the Federation mailbox to the super users group configured on your organization's AD RMS cluster. For details, see Add the Federation Mailbox to the AD RMS Super Users Group.

Important:

In cross-forest AD RMS deployments where you have an AD RMS cluster deployed in each forest, you must add the Federation mailbox to the super users group on the AD RMS cluster in each forest to allow Exchange 2013 Transport service to decrypt the messages protected against each AD RMS cluster.

For details about how to configure journal report decryption, see Enable or Disable Journal Report Decryption.

After you enable journal report decryption, the journaling mailbox may contain journal reports with sensitive information in an unencrypted form. As a best practice, we recommend that access to the journaling mailbox be monitored closely and restricted only to authorized individuals. This is a

best-practice even if you're not using IRM protection for e-mail.

Information Rights Management in Outlook Web App

Applies to: *Exchange Server 2013*

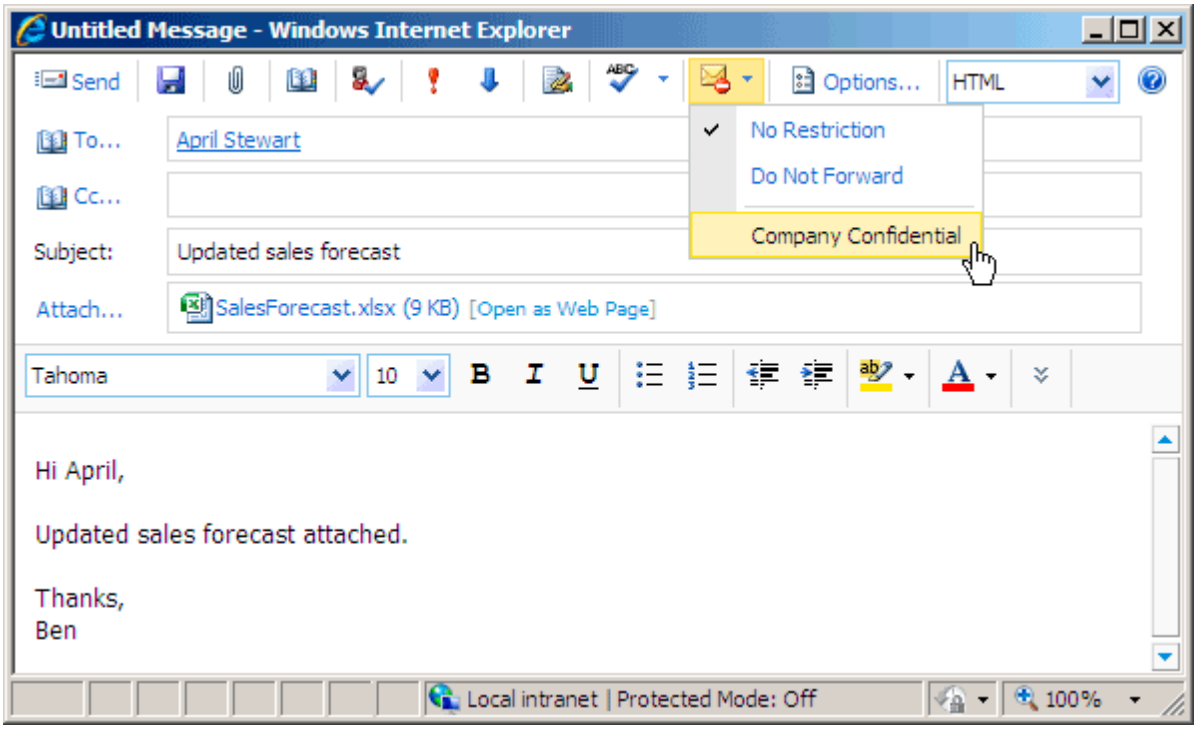
Topic Last Modified: *2012-10-16*

Information workers increasingly use e-mail to exchange sensitive information. To help secure this information, organizations can use Information Rights Management (IRM) to apply persistent protection to messaging content. Prior to Microsoft Exchange Server 2010, effective use of IRM protection was limited to Outlook clients. In Exchange Server 2007, Microsoft Outlook Web Access users were required to download the Rights Management add-in for Microsoft Internet Explorer so they could access IRM-protected content.

In Exchange 2013, IRM in Outlook Web App allows your users to access the rich IRM functionality offered by Exchange to apply persistent IRM-protection to messaging content.

The following IRM functionality is available in Outlook Web App:

- **Send IRM-protected messages** As shown in the following figure, Outlook Web App users can use the permissions drop-down list and select a rights policy template to apply to the message. This allows users to send IRM-protected messages from within Outlook Web App. Messages are IRM-protected by Client Access servers.



- **IRM-protected attachments** When users send an IRM-protected message from Outlook Web App, any files attached to the message also receive the same IRM protection and are protected by using the same rights policy template as the message. In Exchange 2013, IRM protection is applied to files associated with Microsoft Office Word, Excel, and PowerPoint, as well as .xps files and e-mail messages. IRM protection is applied to an attachment only if it's not already IRM-protected. To learn more about Active Directory Rights Management Services (AD RMS) rights policy templates, see Information Rights Management.

Note:
IRM in Outlook Web App protects only the supported file attachments mentioned in this section. Attachments that use unsupported file formats aren't protected. When Outlook Web App users protect a message and attach a file of an unsupported type, a notification is displayed informing the users that only supported file types are protected.

Important:
IRM protection can't be applied to a message that's already signed or encrypted by using S/MIME. To apply IRM protection, S/MIME signature and encryption must be removed from the message. The same applies for IRM-protected messages; users can't sign or encrypt them by using S/MIME.

- **Read IRM-protected messages** Messages protected by senders using your organization's AD RMS cluster are rendered in the preview pane in Outlook Web App. No add-ins need to be installed, and the computer doesn't need to be enrolled in the AD RMS deployment. When a user opens a message or views it in the preview pane, the message is decrypted by using the use license added by the Pre-licensing agent. After decryption, the message is displayed in the preview pane. If a pre-license isn't available, Outlook Web App requests one from the AD RMS server and then renders the message. When reading IRM-protected attachments in Outlook Web App, Web-Ready Document Viewing isn't available.

Note:
IRM in Outlook Web App can't prevent users from taking screen captures by using Print Screen functionality in the way Outlook and other Office applications do. This impacts the EXTRACT right, which prevents message content from being copied, if specified in the AD RMS rights policy template.

- **Cross-browser, multiple platform IRM support** IRM in Outlook Web App offers cross-browser, multiple platform IRM support. IRM in Outlook Web App is supported in all browsers supported by Exchange 2013, including on Apple Macintosh and Linux operating systems. To learn more about supported browsers and operating systems, see Outlook Web App Supported Browsers.
- **WebReady Document Viewing** In Exchange 2013, users can view supported IRM-protected attachments by using WebReady Document Viewing. This allows users to view supported attachments without having to download the attachment use the associated application.

Looking for management tasks related to managing IRM? See Information Rights Management procedures.

Enabling IRM in Outlook Web App

To enable IRM in Outlook Web App, you must add the Federation mailbox, a system mailbox created by Exchange 2013 Setup, to the super users group in AD RMS. For details, see Add the Federation Mailbox to the AD RMS Super Users Group. This allows Exchange 2013 servers to access IRM-protected messages.

You must also enable IRM in Outlook Web App by using the Set-IRMConfiguration cmdlet in the Exchange Management Shell. This enables IRM in Outlook Web App for your Exchange 2013 organization. You can disable or enable IRM in Outlook Web App for an Outlook Web App virtual directory. You can also control IRM in Outlook Web App at the following levels of granularity:

- **Per-Outlook Web App virtual directory** To enable or disable IRM in Outlook Web App for an Outlook Web App virtual directory, use the **Set-OWAVirtualDirectory** cmdlet and set the *IRMEnabled* parameter to `$false` or `$true` (default). This allows you to disable IRM in Outlook Web App for one virtual directory on an Exchange 2013 Client Access server, while keeping it enabled on another virtual directory on a different Client Access server.
- **Per-Outlook Web App mailbox policy** To enable or disable IRM in Outlook Web App for an Outlook Web App mailbox policy, use the **Set-OWAMailboxPolicy** cmdlet and set the *IRMEnabled* parameter to `$false` or `$true` (default). This allows you to enable IRM in Outlook Web App for one set of users and disable it for another set of users by assigning them a different Outlook Web App mailbox policy.

For more information, see Enable or Disable Information Rights Management on Client Access Servers.

Information Rights Management in Exchange ActiveSync

Exchange Server 2013 > Messaging policy and compliance > Information Rights Management >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-13*

Information workers often use e-mail to exchange sensitive information. To help secure this information, organizations can use Information Rights Management (IRM) to apply persistent protection to messaging content. Because mobile devices are increasingly being used to access e-mail, it's important that your mobile device users be able to create and consume IRM-protected content.

Contents

Security

Enabling IRM in Exchange ActiveSync

Looking for management tasks related to IRM? See Information Rights Management procedures.

Mobile IRM protection in Exchange 2013

In Exchange 2013, IRM in Microsoft Exchange ActiveSync allows your users to access rich IRM functionality on any supported Exchange ActiveSync device without having to configure AD RMS permissions or connect the device to a computer and activate it for IRM. Also, the mobile device doesn't need to be running Windows. Exchange ActiveSync is licensed by Microsoft to mobile device manufacturers, original equipment manufacturers (OEMs), and others. For a list of current Exchange ActiveSync licensees, see Exchange ActiveSync Protocol.

Using IRM in Exchange ActiveSync, mobile device users can:

- Create IRM-protected messages.
- Read IRM-protected messages.
- Reply to and forward IRM-protected messages.

Return to top

Requirements

The following requirements apply:

- The Client Access servers in your organization must be running Exchange 2010 SP1 or later.
- An AD RMS server must be deployed in your organization.
- IRM must be enabled for internal messages. This is a prerequisite for all IRM features in Exchange 2010. For details, see Enable or Disable IRM for Internal Messages.
- IRM must be enabled in the Exchange ActiveSync mailbox policy. You can enable or disable IRM for different sets of users using different Exchange ActiveSync mailbox policies.
- Devices that support Exchange ActiveSync protocol version 14.1, including Windows phones, can support IRM in Exchange ActiveSync. The device's mobile e-mail application must support the RightsManagementInformation tag defined in Exchange ActiveSync version 14.1.

Return to top

Security

When you enable IRM in Exchange ActiveSync, the Client Access server decrypts IRM-protected messages before providing the messages for access by the supported mobile device. Upon synchronization, IRM-protected messages reside on the mobile device in an unencrypted format. IRM protection is enforced by the IRM-capable e-mail client application on the mobile device.

IRM in Exchange ActiveSync doesn't decrypt IRM-protected attachments on the Client Access server.

Access to IRM-protected files is enforced by the application used to create or view the file. For example, on a Windows phone, IRM protection for Microsoft Office files is enforced by Microsoft Office Mobile. To access IRM-protected Office files, users must connect the device to a computer and activate Office Mobile with the RMS server.

When enabling IRM in Exchange ActiveSync, we recommend using the Exchange ActiveSync policy settings shown in the following table to help secure mobile devices.

Exchange ActiveSync policy settings

Setting	Configure using the New Exchange ActiveSync Mailbox Policy wizard	Configure using the New-ActiveSyncMailboxPolicy cmdlet
Require that the user enter a password to access information on their mobile device.	Select the **Require password** check box.	Set the *DevicePasswordEnabled* parameter to `$true`.
Enable encryption for the mobile device.	Select the **Require password** check box, and then select the **Require encryption on device** check box.	Set the *RequireDeviceEncryption* parameter to `$true`. **Important:** When you set the *RequireDeviceEncryption* parameter to `$true`, mobile devices that don't support device encryption will be unable to connect.
Don't allow non-provisionable mobile devices to synchronize with the Exchange server.	Clear the **Allow non-provisionable devices** check box.	Set the *AllowNonProvisionableDevices* parameter to `$false`.

To learn more, see Mobile device mailbox policies.

Return to top

Enabling IRM in Exchange ActiveSync

To enable IRM in Exchange ActiveSync, perform the following tasks:

1. Add the Federation mailbox (a system mailbox created by Exchange 2013 and Exchange 2010 Setup) to the super users group in AD RMS. This allows Exchange 2013 and Exchange 2010 servers to access IRM-protected messages. For details, see Add the Federation Mailbox to the AD

RMS Super Users Group.

2. Use the Set-IRMConfiguration cmdlet in the Exchange Management Shell to enable IRM on the Client Access server. This enables IRM in Exchange ActiveSync and IRM in Microsoft Office Outlook Web App for your organization. For details, see Enable or Disable Information Rights Management on Client Access Servers.

Return to top

Information Rights Management logging

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-13*

In Microsoft Exchange Server 2013, Information Rights Management (IRM) operations are logged in IRM logs. IRM logs help you monitor and troubleshoot interactions between the Rights Management Services (RMS) client on an Exchange 2013 server and the Active Directory Rights Management Services (AD RMS) cluster in your organization.

To learn about IRM, see Information Rights Management.

Contents

Looking for management tasks related to IRM? See Information Rights Management procedures.

Structure of IRM logs

By default, IRM logs are located in C:\Program Files\Microsoft\Exchange Server\V14\Logging\IRMLogs.

The naming convention for IRM log files is <*Process*>_<*Process identifier* or *IIS AppPool identifier*>_IRMLOG*yyyymmdd-nnnn*.log, where:

- <*Process*> = process that creates the log file. For example, on Transport service, this will be EdgeTransport.

- *<Process identifier* or *IIS AppPool identifier>* = numerical ID of the process.
- *yyyymmdd* = Coordinated Universal Time (UTC) date when the log file was created.
- *nnnn* = instance number, which starts at 1 for each day.

An example IRM log file name is EdgeTransport_1056_IRMLOG20101201-1.log.

The following table shows the logs generated on different server roles.

Logs on server roles

Server role	IRM log file name	Description
Transport service	EdgeTransport_<*Process identifier*>_IRMLOG*yyyymmdd*-*nnnn*.log	This log is used to record all RMS transactions made by the transport pipeline on Transport service (for example, transport protection rules and journal report decryption). The process identifier (PID) of the edgetransport.exe process is used to generate the log file name.
Mailbox	msftefd_<*Process identifier*>_IRMLOG*yyyymmdd*-*nnnn*.log	This log is used to record all RMS transactions that occur during search and index requests. Exchange 2013 Mailbox servers use the msftefd.exe process for content indexing. The PID of the msftefd.exe process is used to generate the log file name.
Client Access	w3wp_MSExchangeOWAAppOol_IRMLOG*yyyymmdd*-*nnnn*.log	This log is used to record all transactions for IRM in Microsoft Office Outlook Web App.
All Exchange 2013 server roles	w3wp_MSExchangePowerShellAppPool_IRMLOG*yyyymmdd*-	This log is used to record all IRM RMS transactions issued

	nnnn.log	from Windows PowerShell, for example, when issuing the **Test-IRMConfiguration** cmdlet.

Return to top

Logging process

Information is written to the log file until the file size reaches its maximum specified value. When the maximum size is reached, a log file that has an incremental instance number is created. This process is repeated throughout the day. Circular logging deletes the oldest log files when the IRM log directory reaches its maximum specified size or when a log file reaches the maximum age specified in the IRM logging configuration on each server.

Return to top

Information written to IRM logs

IRM log files are text files that contain data in comma-separated value (CSV) format. Each IRM log has a header that contains the following information:

- **#Software** Name of the software that created the IRM log file. Typically, the value is `Microsoft Exchange Server`.
- **#Version** Version number of the software that created the IRM log file.
- **#Log-type** Log type value, which is `Rms Client Manager Log`.
- **#Date** The UTC date and time when the log file was created. The UTC date and time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where:
  - yyyy = year
  - *mm* = month
  - *dd* = day
  - T = time designator used to show the start of the time component
  - *hh* = hour
  - *mm* = minute
  - *ss* = second
  - *fff* = fractions of a second
  - Z = Zulu, which is another way to denote UTC
- **#Fields** Comma-delimited field names used in IRM log files.

The IRM log stores each RMS transaction event on a single line, organized in comma-separated fields. The following table lists the fields in IRM logs for all server roles that have IRM features enabled.

Fields used in IRM logs

Field	Description
Date-time	Lists the UTC timestamp.
Feature	Lists the RMS client feature used. Valid values include: ○ `RacClc` ○ `Template` ○ `Prelicense` ○ `UseLicense` ○ Signature verification ○ `ServerInfo`
Event-Type	Lists the event type. Valid values include: ○ `Acquire` An RMS license or template is requested. ○ `Success` An RMS license or template is acquired successfully. ○ `Exception` An error has occurred. ○ `Queued` A request is pending.
Tenant-Id	Reserved for internal Microsoft use.
Server-url	Lists the RMS server URL accessed during the operation.
Context	Used by the calling process to tie multiple RMS transactions together. Valid values include: ○ `MessageID: <Actual message ID>` ○ `MailboxGuid: <Mailbox GUID>` ○ `AttachmentFileName: <File name>`
Transaction-id	Identifies a unique transaction. All events that occur during one transaction have the same transaction ID.

Return to top

Managing IRM logs

On each server role that has IRM features enabled, IRM logging is enabled by default. For each server role, you can modify the following IRM log configuration by using the server role's corresponding **Set** cmdlet. For example, to configure IRM logging on a Mailbox server, you use the **Set-MailboxServer** cmdlet.

Configuration parameters for IRM logs

Parameter	Description
IrmLogEnabled	Enables logging of IRM transactions. IRM logging is enabled by default. To disable IRM logging for a server role, set the parameter to `$false`.
IrmLogMaxAge	Specifies the maximum age for an IRM log file. Files older than the specified age are deleted. The default value is `30.00:00:00` (30 days).
IrmLogMaxDirectorySize	Specifies the maximum size of all IRM logs in the connectivity log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is `250 MB`.
IrmLogMaxFileSize	Specifies the maximum file size for a single log file. When a file reaches the specified size, a log file is created, and the instance number is incremented. The default value is `10 MB`.
IrmLogPath	Specifies the IRM log location. The default path is %ExchangeInstallPath%Logging\IRMLogs.

For detailed syntax and parameter information, see the following topics:

- Set-MailboxServer
- Set-ClientAccessServer
- Set-TransportService

Return to top

Information Rights Management procedures

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-11-01*

Enable or Disable IRM for Internal Messages

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-12*

In Microsoft Exchange Server 2013, Information Rights Management (IRM) is enabled by default for internal messages. This allows you to create transport protection rules and Microsoft Outlook protection rules to IRM-protect messages in transport and on Microsoft Outlook 2010 and later clients. Enabling IRM for internal messages is a prerequisite for all other IRM features in Exchange Server 2013, such as transport decryption, journal rule decryption, IRM in Microsoft Office Outlook Web App, and IRM in Microsoft Exchange ActiveSync.

Caution:

Disabling IRM for internal messages disables all IRM features in the Exchange organization. The client-side IRM features in Outlook (for example, the ability to read, reply to, forward, and create IRM-protected messages using an Active Directory Rights Management Services (AD RMS) server) aren't affected.

For additional management tasks related to IRM, see Information Rights Management procedures.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Rights protection" entry in the Messaging policy and compliance permissions topic.
- You can't use the Exchange Administration Center (EAC) to enable or disable IRM for internal messages. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to enable IRM for internal messages

This example enables IRM for internal messages for the Exchange organization.

```
Set-IRMConfiguration -InternalLicensingEnabled $true
```

For detailed syntax and parameter information, see Set-IRMConfiguration.

Use the Shell to disable IRM for internal messages

This example disables IRM for internal messages for the Exchange organization.

```
Set-IRMConfiguration -InternalLicensingEnabled $false
```

For detailed syntax and parameter information, see Set-IRMConfiguration.

How do you know this worked?

To verify that you've enabled or disabled IRM for internal messages, use the Get-IRMConfiguration cmdlet to check the configuration.

Create a Transport Protection Rule

Messaging policy and compliance > Information Rights Management > Information Rights Management procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-04*

You can use transport protection rules to apply persistent rights protection to messages based on properties such as sender, recipient, message subject, and content.

Caution:

Before you create transport rules in your production environment, we recommend creating them in a test environment and testing them thoroughly. The transport rules created in this topic are examples. You can create transport rules by using the appropriate transport rule predicates and values based on your requirements.

For additional management tasks related to Information Rights Management (IRM), see Information Rights Management procedures.

What do you need to know before you begin?

- Estimated time to completion: 2-5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.
- A server running Active Directory Rights Management Services (AD RMS) must be available in your organization and contain existing RMS templates.
- If you configure transport protection rules to protect messages using IRM, and you also use journaling, consider enabling journal report decryption to allow the Journaling agent to save an unencrypted copy of the message in the journal report. To learn more, see Journal report decryption.
- After you create a transport protection rule, if the rule can't be applied to messages because an AD RMS server is unavailable, messages will be queued by the Transport service on Mailbox servers. Depending on the volume of these messages, additional disk space may be consumed on Mailbox servers. Exchange will attempt to IRM-protect the message three times. After these attempts, if the AD RMS server is unreachable or the message can't be IRM-protected, a non-delivery report (NDR) is sent to the sender.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to create a transport protection rule

1. Navigate to **Mail flow** > **Rules**.
2. In the list view, click **New +**.
3. In **New Rule**, first click **More options**, and then complete the following fields:
   - **Name** Type a name for the transport rule.
   - **Apply this rule if** Select a condition and enter any required values for the condition. To add more conditions, click **Add condition**.

Important:

If you don't select any conditions when creating a transport protection rule, all messages handled by Exchange 2013 servers with the Transport service in your organization are IRM-protected. IRM-protecting all messages requires more resources. Therefore, we recommend that you plan your Mailbox server and AD RMS deployment accordingly.

   - **Do the following** Select **Apply rights protection to the message with** and then use the **Select RMS template** dialog box to select a template.
   - **Except if** (Optional) Click **Add exception** to specify an exception to the rule.
4. Click **Save** to create the transport rule.

Use the Shell to create a transport protection rule

- To create a transport protection rule, you must have existing RMS templates in your AD RMS deployment. This example retrieves the available templates from your AD RMS cluster.

```
Get-RMSTemplate | format-list
```

For detailed syntax and parameter information, see Get-RMSTemplate.

- This example creates the transport protection rule Protect-BusinessCriticalProject. The rule IRM-protects messages that contain the phrase "Business Critical" in the Subject field with the **Do Not Forward** template.

Note:

The `SubjectContainsWords` predicate is used in this example. You can use any combination of transport rule predicates to form the conditions and exceptions for the rule. For information about the available predicates, see Transport rule conditions (predicates).

```
New-TransportRule -Name "Protect-BusinessCriticalProject" -
SubjectContainsWords "Business Critical" -
ApplyRightsProtectionTemplate "Do Not Forward"
```

For detailed syntax and parameter information, see New-TransportRule.

How do you know this worked?

To verify that you have successfully created a transport protection rule, do one of the following:

- Use the EAC to verify that the rule has been created, and then click **Edit** to view the rule's properties.

- Use the Get-TransportRule cmdlet to retrieve the rule. For an example of how to retrieve a rule, see Examples in **Get-TransportRule**.
- Using Outlook, Outlook Web App, or a mobile device, send a test message that meets the rule conditions and check whether the message received by the recipient is IRM-protected.

Create an Outlook Protection Rule

Messaging policy and compliance > Information Rights Management > Information Rights Management procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-04*

Using Microsoft Outlook protection rules, you can protect messages with Information Rights Management (IRM) by applying an Active Directory Rights Management Services (AD RMS) template in Outlook 2010 before the messages are sent.

For additional management tasks related to IRM, see Information Rights Management procedures.

What do you need to know before you begin?

- Estimated time to completion: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Rights protection" entry in the Messaging policy and compliance permissions topic.
- You must have an AD RMS server deployed in the same Active Directory forest as your server running Microsoft Exchange Server 2013.
- If you configure Outlook protection rules to IRM-protect messages, consider enabling transport decryption to allow transport agents, including the Transport Rules agent, to decrypt and access the message. If you use journaling, you should also consider enabling journal report decryption to allow the Journaling agent to save an unencrypted copy of the message in the journal report. For more information, see Journal report decryption.
- You can't use the Exchange Administration Center (EAC) to create Outlook protection rules. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to create an Outlook protection rule

This example creates the Outlook protection rule Project Contoso. The rule protects messages sent to the ContosoPMs distribution group with the AD RMS template Business Critical.

```
New-OutlookProtectionRule -Name "Project Contoso" -SentTo
"DL-ContosoPMs@contoso.com" -ApplyRightsProtectionTemplate
"Business Critical"
```

Note:

When you use the `SentTo` predicate for an Outlook protection rule and specify a distribution group, only messages addressed to the distribution group in the To, Cc, or Bcc fields are IRM-protected. IRM protection isn't applied to messages addressed to individual members of the distribution group.

You can also use the `FromDepartment` and `SentToScope` predicates to apply IRM protection to messages sent from users in the specified department or messages sent to the specified scope (`InOrganization` for internal messages, `All` for all recipients).

For detailed syntax and parameter information, see New-OutlookProtectionRule.

How do you know this worked?

To verify that you have successfully created an Outlook protection rule, do the following:

- Run the Get-OutlookProtectionRule cmdlet to make sure that the rule has been created and to view the rule's properties. For an example of how to retrieve an Outlook protection rule, see Examples in **Get-OutlookProtectionRule**.
- Use Outlook 2010 to create a test message that meets the rule's condition and make sure the rule is triggered on the client.

Note:

It may take some time for an Outlook protection rule to be available in Outlook.

Remove an Outlook Protection Rule

Messaging policy and compliance > Information Rights Management > Information Rights Management procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-04*

Using Microsoft Outlook protection rules, you can protect messages with Information Rights Management (IRM) by applying an Active Directory Rights Management Services (AD RMS) template in Outlook 2010 before the messages are sent. To prevent an Outlook protection rule from being applied, you can disable the rule. Removing an Outlook protection rule removes the

rule definition from Active Directory.

For additional management tasks related to IRM, see Information Rights Management procedures.

What do you need to know before you begin?

- Estimated time to completion: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Rights protection" entry in the Messaging policy and compliance permissions topic.
- You can't use the Exchange Administration Center (EAC) to remove Outlook protection rules. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to remove an Outlook protection rule

This example removes the Outlook protection rule OPR-DG-Finance.

```
Remove-OutlookProtectionRule -Identity "OPR-DG-Finance"
```

For detailed syntax and parameter information, see Remove-OutlookProtectionRule.

Use the Shell to remove all Outlook protection rules

This example removes all Outlook protection rules in the Exchange organization.

```
Get-OutlookProtectionRule | Remove-OutlookProtectionRule
```

For detailed syntax and parameter information, see Get-OutlookProtectionRule and Remove-OutlookProtectionRule.

How do you know this worked?

To verify that you have successfully removed an Outlook protection rule, use the Get-OutlookProtectionRule cmdlet to retrieve Outlook protection rules. For an example of how to retrieve Outlook protection rules, see Examples in **Get-OutlookProtectionRule**.

Add the Federation Mailbox to the AD RMS Super Users Group

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-10*

For the following Microsoft Exchange Server 2013 Information Rights Management (IRM) features to be enabled, you must add the Federation mailbox (a system mailbox created by Exchange 2013 Setup) to the super users group on your organization's Active Directory Rights Management Services (AD RMS) cluster:

- IRM in Microsoft Office Outlook Web App
- IRM in Exchange ActiveSync
- Journal report decryption
- Transport decryption

You can configure a mail-enabled distribution group as a super users group in AD RMS. Members of the distribution group are granted an owner use license when they request a license from the AD RMS cluster. This allows them to decrypt all RMS-protected content published by that cluster. Whether you use an existing distribution group or create a distribution group and configure it as the super users group in AD RMS, we recommend that you dedicate the distribution group for this purpose and configure the appropriate settings to approve, audit, and monitor membership changes.

Caution:

Configuring a super users group in AD RMS allows group members to decrypt IRM-protected content. We recommend that you take adequate measures to control and monitor group membership and enable auditing to track membership changes. You can also limit unwanted changes to group membership by configuring the group as a restricted group using Group Policy. For details, see Restricted Groups Policy Settings.

For additional management tasks related to IRM, see Information Rights Management procedures.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.
- An AD RMS cluster must be deployed in the Active Directory forest.
- If a super users group is already configured on an AD RMS cluster, any modifications to the

distribution group membership can take up to 24 hours to be refreshed by the AD RMS cluster. This is a result of caching the group membership on the cluster.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Use the Shell to add the Federation mailbox to a distribution group

If a distribution group has been created and configured as a super users group in the AD RMS cluster, you can add the Exchange 2013 Federation mailbox as a member of that group. If a super users group isn't configured, you must create a distribution group and add the Federation mailbox as a member.

1. Create a distribution group dedicated for use as an AD RMS super users group. For details, see Manage Distribution Groups.
2. Add the user **FederatedEmail.4c1f4d8b-8179-4148-93bf-00a95fa1e042** to the new distribution group. The Federation mailbox is a system mailbox, and therefore not visible in the EAC. To add it to a distribution group, you must use the Add-DistributionGroupMember cmdlet from the Shell.

This example adds the Federation mailbox to the ADRMSSuperUsers distribution group.

```
Add-DistributionGroupMember ADRMSSuperUsers -Member
FederatedEmail.4c1f4d8b-8179-4148-93bf-00a95fa1e042
```

For detailed syntax and parameter information, see Add-DistributionGroupMember.

Step 2: Use AD RMS to set up a super users group

Perform the following procedure on an AD RMS cluster. The account used to perform this procedure must be a member of the AD RMS Enterprise Administrators local group on the AD RMS server.

1. Open the Active Directory Rights Management Services console and expand the AD RMS cluster.
2. In the console tree, expand **Security Policies**, and then click **Super Users**.
3. In the action pane, click **Enable Super Users**.
4. In the result pane, click **Change Super User Group** to open the **Super Users** property sheet.
5. In the **Super user group** box, type the email address of the distribution group you created in the

previous procedure, or click **Browse** to select a distribution group.

How do you know this worked?

After you have added the Federation mailbox to a new or existing distribution group, use the Get-DistributionGroupMember cmdlet to check the membership of the group.

For an example of how to check distribution group membership, see Example 1 in **Get-DistributionGroupMember**.

After you have used AD RMS to set up a super users group, you can use the following methods to verify that the super users group has been configured correctly. Additionally, you can use Test-IRMConfiguration cmdlet to verify IRM functionality.

- Use the AD RMS console to verify that the correct group has been configured as the super users group.
- Run the following PowerShell command on an AD RMS server to retrieve the super users group.

Important:
The ADRMSAdmin PowerShell module is available in Windows Server 2008 R2 and later.

```
Import-Module ADRMSAdmin
New-PSDrive -Name MyRmsAdmin -PsProvider AdRmsAdmin -Root https://localhost
Get-ItemProperty -Path MyRmsAdmin:\SecurityPolicy\SuperUser
```

Enable or Disable Transport Decryption

Messaging policy and compliance > Information Rights Management > Information Rights Management procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-22*

Enabling transport decryption allows the Transport Rules agent on Microsoft Exchange Server 2013 Mailbox servers to access content in messages protected by Information Rights Management (IRM). As a result, other transport agents can access message content and possibly make changes to it. For example, the Transport Rules agent may need to inspect message content and apply transport rules (such as rules that apply a disclaimer to the message). To successfully decrypt IRM-protected messages, you must add the Federated Delivery mailbox to the super users group configured on your Active Directory Rights Management Services (AD RMS) server.

Important:
Members of the super users group are granted an owner use license when they request a

license from the AD RMS cluster. This allows them to decrypt all RMS-protected content created by that AD RMS cluster.

When enabling transport decryption, you can specify the following settings:

- **Mandatory** Rejects messages that can't be decrypted and returns a non-delivery report (NDR) to the sender.
- **Optional** Uses a best-effort approach to decryption. If possible, messages are decrypted, but they're delivered even if decryption fails. This is the default setting.

To learn more about transport decryption, see Transport decryption.

For additional management tasks related to IRM, see Information Rights Management procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Rights protection" entry in the Messaging policy and compliance permissions topic.
- An AD RMS server exists in the Active Directory forest and is accessible.
- The Federated Delivery mailbox has been added to the AD RMS super users group. For details, see Add the Federation Mailbox to the AD RMS Super Users Group.
- You can't use the Exchange Administration Center (EAC) to enable transport decryption. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable transport decryption

This example enables transport decryption for the Exchange 2013 organization. Messages that can't be decrypted are rejected and an NDR is returned to the sender.

```
Set-IRMConfiguration -TransportDecryptionSetting Mandatory
```

For detailed syntax and parameter information, see Set-IRMConfiguration.

Use the Shell to disable transport decryption

This example disables transport decryption for the Exchange 2013 organization.

```
Set-IRMConfiguration -TransportDecryptionSetting Disabled
```

For detailed syntax and parameter information, see Set-IRMConfiguration.

How do I know this worked?

To verify that you have enabled or disabled transport decryption, use the **Get-IRMConfiguration** cmdlet and check the value of the *JournalDecryptionEnabled* property.

For an example of how to check the IRM configuration, see Examples in **Get-IRMConfiguration**.

Configure IRM for Exchange Search and In-Place eDiscovery

Messaging policy and compliance > Information Rights Management > Information Rights Management procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-16*

In Microsoft Exchange Server 2013, you can configure Information Rights Management (IRM) so that Exchange Search can index IRM-protected messages.

When members of the Discovery Management role group perform an In-Place eDiscovery search, IRM-protected messages are returned in the search results and copied to the Discovery mailbox specified in the search. Furthermore, members of the Discovery Management role group can use Outlook Web App to access the IRM-protected messages that were copied to the Discovery mailbox as a result of the discovery search.

Note:
Members of the Discovery Management role group can't access IRM-protected messages exported from a Discovery mailbox to another mailbox or to a .pst file. IRM-protected messages in a Discovery mailbox can be accessed only by using Outlook Web App.

For additional management tasks related to IRM, see Information Rights Management procedures.

What do you need to know before you begin?

- Estimated time to completion: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Rights protection" entry in the Messaging policy and compliance permissions topic.

- IRM must be configured in your Exchange 2013 organization. To learn more, see Enable or Disable IRM for Internal Messages.
- The Federation mailbox must be added to the Active Directory Rights Management Services (AD RMS) super users group. To learn more, see Add the Federation Mailbox to the AD RMS Super Users Group.
- You can't use the Exchange Administration Center (EAC) to configure IRM for Exchange Search and In-Place eDiscovery. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to configure IRM for Exchange Search

This example configures IRM to allow Exchange Search to index IRM-protected messages.

Note:

By default, the *SearchEnabled* parameter is set to `$true`. To disable indexing of IRM-protected messages, set it to `$false`. Disabling indexing of IRM-protected messages prevents them from being returned in search results when users search their mailbox or when discovery managers use In-Place eDiscovery.

```
Set-IRMConfiguration -SearchEnabled $true
```

For detailed syntax and parameter information, see Set-IRMConfiguration.

Use the Shell to configure IRM for In-Place eDiscovery

This example enables members of the Discovery Management role group to access IRM-protected messages that reside in the Discovery mailbox.

Note:

By default, the *EDiscoverySuperUserEnabled* parameter is set to `$true`. To disable access to IRM-protected messages for members of the Discovery Management role group, set it to `$false`.

```
Set-IRMConfiguration -EDiscoverySuperUserEnabled $true
```

For detailed syntax and parameter information, see Set-IRMConfiguration.

How do you know this worked?

To verify that you have successfully configured IRM for Exchange Search and In-Place eDiscovery, use the **Get-IRMConfigurtaion** cmdlet to retrieve the IRM configuration information. For an example of how to retrieve the IRM configuration, see Examples in **Get-IRMConfiguration**.

Enable or Disable Journal Report Decryption

Messaging policy and compliance > Information Rights Management > Information Rights Management procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-04*

Enabling journal report decryption allows the Journaling agent to attach a decrypted copy of a rights-protected message to the journal report. Before you enable journal report decryption, you must add the Federated Delivery mailbox to the super users group configured on your Active Directory Rights Management Services (AD RMS) server.

For additional management tasks related to Information Rights Management (IRM), see Information Rights Management procedures.

What do you need to know before you begin?

- Estimated time to complete: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Rights protection" entry in the Messaging policy and compliance permissions topic.
- Members of the super users group are granted an owner use license when they request a license from the AD RMS cluster. This allows them to decrypt all RMS-protected content created by that AD RMS cluster.
- An AD RMS cluster must be installed in the Active Directory forest.
- The Federated Delivery mailbox has been added to an AD RMS super users group. For details, see Add the Federation Mailbox to the AD RMS Super Users Group.
- You can't use the Exchange Administration Center (EAC) to enable journal report decryption. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable journal report decryption

This example enables journal report decryption for the Exchange organization.

```
Set-IRMConfiguration -JournalReportDecryptionEnabled $true
```

For detailed syntax and parameter information, see Set-IRMConfiguration.

Use the Shell to disable journal report decryption

This example disables journal report decryption for the Exchange organization.

```
Set-IRMConfiguration -JournalReportDecryptionEnabled $false
```

For detailed syntax and parameter information, see Set-IRMConfiguration.

How do you know this worked?

To verify that you have enabled or disabled journal report decryption, run the **Get-IRMConfiguration** cmdlet and check the value of the *JournalDecryptionEnabled* property.

For an example of how to check the IRM configuration, see Examples in **Get-IRMConfiguration**.

Enable or Disable Information Rights Management on Client Access Servers

Messaging policy and compliance > Information Rights Management > Information Rights Management procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-04*

Enabling Information Rights Management (IRM) on Client Access servers enables the following features:

- Microsoft Office Outlook Web App
- IRM in Microsoft Exchange ActiveSync

When IRM is enabled on Client Access servers, Outlook Web App users can IRM-protect messages by applying an Active Directory Rights Management Services (AD RMS) template created on your AD RMS cluster. Outlook Web App users can also view IRM-protected messages and supported attachments. Before you enable IRM on Client Access servers, you must add the Federation mailbox to the super users group on the AD RMS cluster.

Important:
Members of the super users group are granted an owner use license when they request a license from the AD RMS cluster. This allows them to decrypt all RMS-protected content by that cluster.

For additional management tasks related to IRM, see Information Rights Management procedures.

What do you need to know before you begin?

- Estimated time to completion: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.
- An AD RMS cluster must be installed in the Active Directory forest.
- The Federation mailbox has been added to the AD RMS super users group. For detailed instructions, see Add the Federation Mailbox to the AD RMS Super Users Group.
- IRM features must be enabled for the organization. For detailed instructions, see Enable or Disable IRM for Internal Messages.
- You can use the **Set-IRMConfiguration** cmdlet to enable or disable IRM in Outlook Web App and IRM in Exchange ActiveSync for the entire Exchange organization or at specific levels.

You can control IRM in Outlook Web App at the following levels:

  - **Per-Outlook Web App virtual directory** To enable or disable IRM in Outlook Web App for an Outlook Web App virtual directory, use the **Set-OWAVirtualDirectory** cmdlet and set the *IRMEnabled* parameter to `$false` or `$true` (default). This allows you to disable IRM in Outlook Web App for one virtual directory on an Exchange 2013 Client Access server, while keeping it enabled on another virtual directory on a different Client Access server.
  - **Per-Outlook Web App mailbox policy** To enable or disable IRM in Outlook Web App for an Outlook Web App mailbox policy, use the **Set-OWAMailboxPolicy** cmdlet and set the *IRMEnabled* parameter to `$false` or `$true` (default). This allows you to enable IRM in Outlook Web App for one set of users and disable it for another set of users by assigning them a different Outlook Web App mailbox policy.

You can control IRM in Exchange ActiveSync per Exchange ActiveSync mailbox policy. To disable or enable IRM in Exchange ActiveSync for an Exchange ActiveSync mailbox policy, use the **Set-ActiveSyncMailboxPolicy** cmdlet and set the *IRMEnabled* parameter to `$false` or `$true` (default). This allows you to enable IRM in Exchange ActiveSync for one set of users and disable it for another

set of users by assigning them a different Exchange ActiveSync mailbox policy.

- You can't use the Exchange Administration Center (EAC) to enable or disable IRM on Client Access servers. You must use the Shell.

What do you want to do?

Use the Shell to enable IRM on Client Access servers

This example enables IRM on a Client Access server for an Exchange organization.

```
Set-IRMConfiguration -ClientAccessServerEnabled $true
```

For detailed syntax and parameter information, see Set-IRMConfiguration.

Use the Shell to disable IRM on Client Access servers

This example disables IRM on a Client Access server for an Exchange organization.

```
Set-IRMConfiguration -ClientAccessServerEnabled $false
```

For detailed syntax and parameter information, see Set-IRMConfiguration.

How do you know this worked?

To verify that you have successfully enabled or disabled IRM on Client Access servers, do the following:

- Run the **Get-IRMConfiguration** cmdlet and check the value of the *ClientAccessServerEnabled* property.

For an example of how to retrieve the IRM configuration, see Examples in **Get-IRMConfiguration**.

- Use Outlook Web App to create or read an IRM-protected message.

Enable or Disable Information Rights Management Logging

Messaging policy and compliance > Information Rights Management > Information Rights Management procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-12*

In Exchange Server 2013, you can use Information Rights Management (IRM) logs to monitor and troubleshoot IRM operations. IRM logging is enabled by default.

IRM logs use the following common set of parameters:

- *IrmLogEnabled* Enables or disables IRM logging. Default: `$true`.
- *IrmLogMaxAge* Specifies the maximum age of IRM log files. Files older than the specified age are deleted. Default: 30 days.
- *IrmLogMaxDirectorySize* Specifies the maximum size of the directory that contains IRM logs. When a directory reaches its maximum file size, the server deletes the oldest log files first. Default: 250 MB.
- *IrmLogMaxFileSize* Specifies the maximum size of each IRM log file. When a log file reaches the specified size, a new log file is created. Default: 10 MB.
- *IrmLogPath* Specifies the location of the IRM log directory. Default: `%ExchangeInstallPath%Logging\IRMLogs`.

For additional management tasks related to IRM, see Information Rights Management procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 2-5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Configure IRM logging" entry in the Messaging policy and compliance permissions topic.
- You can't use the Exchange Administration Center (EAC) to enable or disable IRM logging on a server. You must use the Shell
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to enable IRM logging on a server

This example enables IRM log on a Mailbox server.

```
Set-TransportService -Identity EXCH01 -IRMLogEnabled $true
```

For detailed syntax and parameter information, see Set-TransportService.

Use the Shell to disable IRM logging on a server

This example disables IRM logging on a Mailbox server.

```
Set-TransportService -Identity EXCH01 -IRMLogEnabled $false
```

For detailed syntax and parameter information, see Set-TransportService.

How do you know this worked?

To verify that you have successfully enabled or disbled IRM logging on a server, run the Get-TransportService cmdlet to retrieve IRM settings.

This example retrieves all IRM logging properties on the server EXCH01.

```
Get-TransportService -Identity EXCH01 | Format-List IRMLog*
```

S/MIME for message signing and encryption

Exchange Server 2013 > Messaging policy and compliance >

Applies to: *Exchange Online*

Topic Last Modified: *2014-03-23*

S/MIME (Secure/Multipurpose Internet Mail Extensions) is a widely accepted method, or more precisely a protocol, for sending digitally signed and encrypted messages. S/MIME allows you to encrypt emails and digitally sign them. When you use S/MIME with an email message, it helps the people who receive that message to be certain that what they see in their inbox is the exact message that started with the sender. It will also help people who receive messages to be certain that the message came from the specific sender and not from someone pretending to be the sender. To do this, S/MIME provides for cryptographic security services such as authentication, message integrity, and non-repudiation of origin (using digital signatures). It also helps enhance privacy and data security (using encryption) for electronic messaging. For a more complete background about the history and architecture of S/MIME in the context of email, see Understanding S/MIME.

As an administrator, you can enable S/MIME-based security for your organization if you have mailboxes in either Exchange 2013 SP1 or Exchange Online, a part of Office 365. Use the guidance in the topics linked here along with the Exchange Management Shell to set up S/MIME. To use S/MIME in supported versions of Outlook or ActiveSync, with either Exchange 2013 SP1 or Exchange Online, the users in your organization must have certificates issued for signing and encryption purposes and data published to your on-premises Active Directory Domain Service (AD DS). Your AD DS must be located on computers at a physical location that you control and not at a remote facility or cloud-based service somewhere on the internet. For more information about AD DS, see

Active Directory Domain Services.

Supported scenarios and technical considerations

If your organization uses either Exchange 2013 SP1 or Exchange Online, you can set up S/MIME to work with any of the following end points:

- Outlook 2010
- Outlook 2013
- Outlook Web App
- Exchange ActiveSync (EAS)

The steps that you follow to set up S/MIME with each of these end points is slightly different depending on which you choose. Generally, you will need to accomplish the following:

- Set up your certificate authority and public key infrastructure to issue S/MIME certificates for users.
- Publish the user certificate in an on-premises AD DS account in the UserSMIMECertificate and/or UserCertificate attributes.
- For Exchange Online organizations, synchronize the user certificates from AD DS to Azure Active Directory by using an appropriate version of DirSync. These certificates will then get synchronized from Azure Active Directory to Exchange Online directory and will be used when encrypting a message to a recipient.
- Set up a virtual certificate collection in order to validate S/MIME. This information is used by OWA when validating the signature of an email and ensuring that it was signed by a trusted certificate.
- Set up the Outlook or EAS end point to use S/MIME.

Setup S/MIME with Outlook Web App

Setting up S/MIME for Exchange 2013 SP1 or Exchange Online with Outlook Web App involves the following key steps:

1. Configure S/MIME settings for Outlook Web App
2. Set up virtual certificate collection to validate S/MIME
3. **Sync user certificates to Office 365 for S/MIME** This step applies to Exchange Online only.

Related message encryption technologies

As message security becomes more important, administrators need to understand the principles and concepts of secure messaging. This understanding is especially important because of the growing variety of protection-related technologies, such as S/MIME, that have become available. To understand more about S/MIME and how it works in context of email, see Understanding S/MIME. A variety of encryption technologies work together to provide protection for messages at rest and in-transit. S/MIME can work simultaneously with the following technologies but is not dependent on them:

Transport Layer Security (TLS) encrypts the tunnel or the route between email servers in order to help prevent snooping and eavesdropping.

Secure Sockets Layer (SSL) encrypts the connection between email clients and Office 365 servers.

BitLocker encrypts the data on a hard drive in a datacenter so that if someone gets unauthorized access, they can't read it.

S/MIME compared with Office 365 Message Encryption

S/MIME requires a certificate and publishing infrastructure that is often used in business-to-business and business-to-consumer situations. The user controls the cryptographic keys in S/MIME and can choose whether to use them for each message they send. Email programs such as Outlook search a trusted root certificate authority location to perform digital signing and verification of the signature. Office 365 Message Encryption is a policy-based encryption service that can be configured by an administrator, and not an individual user, to encrypt mail sent to anyone inside or outside of the organization. It's an online service that's built on Windows Azure Rights Management and does not rely on a public key infrastructure. Office 365 Message Encryption also provides additional capabilities, such as the capability to customize the mail with organization's brand. For more information about Office 365 Message Encryption, see Office 365 Message Encryption.

More information

Outlook Web App

Secure Mail (2000)

Configure S/MIME settings for Outlook Web App

Exchange Server 2013 > Messaging policy and compliance > S/MIME for message signing and encryption >

Applies to: *Exchange Online*

Topic Last Modified: *2014-02-23*

As an organization administrator for both Exchange 2013 and Exchange Online, you can set up Outlook Web App to allow sending and receiving S/MIME-protected messages. Use the `SMIMEConfig` cmdlet to manage this feature through the Exchange Management Shell interface.

For more information such as a detailed description of parameters and examples for `get-SMIMEConfig` and `set-SMIMEConfig`, see the Get-SmimeConfig and Set-SmimeConfig documentation.

You can only use the Shell to perform this procedure. To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell. To learn how to use Windows PowerShell to connect to Exchange Online, see **Connect to Exchange Online using remote PowerShell**.

For more information

S/MIME for message signing and encryption

Set up virtual certificate collection to validate S/MIME

Exchange Server 2013 > Messaging policy and compliance > S/MIME for message signing and encryption >

Applies to: *Exchange Online*

Topic Last Modified: *2014-02-23*

As a tenant administrator you will need to configure a virtual certificate collection that will be used to validate S/MIME certificates. This virtual certificate collection is set up as a certificate store file type with an SST filename extension. The SST file contains all the root and intermediate certificates that are used when validating an S/MIME certificate.

Create and save an SST

You can only use the Shell to perform this procedure. To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell. To learn how to use Windows PowerShell to connect to Exchange Online, see **Connect to Exchange Online using remote PowerShell**.

As an administrator, you can create this SST file by exporting the certificates from a trusted machine using the `Export-Certificate` cmdlet and specifying the type as SST. For more information the `Export-Certificate` cmdlet, see the Export-Certificate reference topic.

Once the SST file is generated, use the `Set-Smimeconfig` cmdlet to save it in the virtual certificate store by using the *-SMIMECertificateIssuingCA* parameter. For example: `Set-SmimeConfig -SMIMECertificateIssuingCA (Get-Content filename.sst -Encoding Byte)`

Ensuring a certificate is valid

Exchange 2013 SP1 first checks for the SST file and validates the certificate. If the validation fails, it will look at the local machine certificate store to validate the certificate. This behavior is new for Exchange 2013 SP1 and different from prior versions of Exchange. In Exchange Online only the SST will be used for validation.

More Information

S/MIME for message signing and encryption

Get-SmimeConfig

Send and receive S/MIME signed and encrypted email

Exchange Server 2013 > Messaging policy and compliance > S/MIME for message signing and encryption >

Applies to: *Exchange Online*

Topic Last Modified: *2014-02-25*

Sending or replying to an S/MIME-encrypted message in Microsoft Outlook is very similar to the experience with a non-encrypted message. For more information about reading or sending S/MIME-encrypted messages from an email program such as Outlook Web App, see Use Outlook to send and reply to S/MIME encrypted messages.

For more information

S/MIME for message signing and encryption

Mailbox audit logging

Exchange Server 2013 > Messaging policy and compliance >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-12-12*

Because mailboxes can contain sensitive, high business impact (HBI) information and personally identifiable information (PII), it's important that you track who logs on to the mailboxes in your

organization and what actions are taken. It's especially important to track access to mailboxes by users other than the mailbox owner. These users are referred to as *delegate users*.

By using *mailbox audit logging*, you can log mailbox access by mailbox owners, delegates (including administrators with full access permissions to mailboxes), and administrators.

When you enable audit logging for a mailbox, you can specify which user actions (for example, accessing, moving, or deleting a message) will be logged for a logon type (administrator, delegate user, or owner). Audit log entries also include important information such as the client IP address, host name, and process or client used to access the mailbox. For items that are moved, the entry includes the name of the destination folder.

Mailbox audit logs

Mailbox audit logs are generated for each mailbox that has mailbox audit logging enabled. Log entries are stored in the Recoverable Items folder in the audited mailbox, in the Audits subfolder. This ensures that all audit log entries are available from a single location, regardless of which client access method was used to access the mailbox or which server or workstation an administrator used to access the mailbox audit log. If you move a mailbox to another Mailbox server, the mailbox audit logs for that mailbox are also moved because they're located in the mailbox.

By default, mailbox audit log entries are retained in the mailbox for 90 days and then deleted. You can modify this retention period by using the *AuditLogAgeLimit* parameter with the Set-Mailbox cmdlet. If a mailbox is on In-Place Hold or Litigation Hold, audit log entries are only retained until the audit log retention period for the mailbox is reached. To retain audit log entries longer, you have to increase the retention period by changing the value for the *AuditLogAgeLimit* parameter. You can also export audit log entries before the retention period is reached. For more information, see:

- Export mailbox audit logs
- Create a mailbox audit log search

Enabling mailbox audit logging

Mailbox audit logging is enabled per mailbox. Use the **Set-Mailbox** cmdlet to enable or disable mailbox audit logging. For details, see Enable or disable mailbox audit logging for a mailbox.

When you enable mailbox audit logging for a mailbox, access to the mailbox and certain administrator and delegate actions are logged by default. To log actions taken by the mailbox owner, you must specify which owner actions should be audited.

Mailbox actions logged by mailbox audit logging

The following table lists the actions logged by mailbox audit logging, including the logon types for which the action can be logged.

Action	Description	Administrator	Delegate	Owner
Copy	An item is copied to another folder.	Yes	Yes	No
Create	An item is created in the mailbox. (For example, a message is sent or received.) Note that folder creation isn't audited.	Yes*	Yes*	Yes
FolderBind	A mailbox folder is accessed.	Yes*	Yes**	No
HardDelete	An item is deleted permanently from the Recoverable Items folder.	Yes*	Yes*	Yes
MessageBind	An item is accessed in the reading pane or opened.	Yes	No	No
Move	An item is moved to another folder.	Yes*	Yes	Yes
MoveToDeletedItems	An item is moved to the Deleted Items folder.	Yes*	Yes	Yes
SendAs	A message is sent using Send As permissions.	Yes*	Yes*	Not applicable

SendOnBehalf	A message is sent using Send on Behalf permissions.	Yes*	Yes	Not applicable
SoftDelete	An item is deleted from the Deleted Items folder.	Yes*	Yes*	Yes
Update	An item's properties are updated.	Yes*	Yes*	Yes

* Audited by default if auditing is enabled for a mailbox.

** Entries for folder bind actions performed by delegates are consolidated. One log entry is generated for individual folder access within a time span of three hours.

If you no longer require certain types of mailbox actions to be audited, you should modify the mailbox's audit logging configuration to disable those actions. Existing log entries aren't purged until the age limit for audit log entries is reached.

Searching mailbox audit log entries

You can use the following methods to search mailbox audit log entries:

- **Synchronously search a single mailbox** You can use the Search-MailboxAuditLog cmdlet to synchronously search mailbox audit log entries for a single mailbox. The cmdlet displays search results in the Exchange Management Shell window. For details, see Search the mailbox audit log for a mailbox.
- **Asynchronously search one or more mailboxes** You can create a mailbox audit log search to asynchronously search mailbox audit logs for one or more mailboxes, and then have the search results sent to a specified email address. The search results are sent as an XML attachment. To create the search, use the New-MailboxAuditLogSearch cmdlet. For details, see Create a mailbox audit log search.
- **Use auditing reports in the Exchange Admin Center (EAC)** You can use the **Auditing** tab in the EAC to run a non-owner mailbox access report or export entries from the mailbox audit log. For details, see:
  - Run a non-owner mailbox access report
  - Export mailbox audit logs

Mailbox audit log entries

The following table describes the fields logged in a mailbox audit log entry.

Field	Populated with
Operation	One of the following actions: • Copy • Create • FolderBind • HardDelete • MessageBind • Move • MoveToDeletedItems • SendAs • SendOnBehalf • SoftDelete • Update
OperationResult	One of the following results: • Failed • PartiallySucceeded • Succeeded
LogonType	Logon type of the user who performed the operation. Logon types include: • Owner • Delegate • Admin
DestFolderId	Destination folder GUID for move operations.
DestFolderPathName	Destination folder path for move operations.
FolderId	Folder GUID.
FolderPathName	Folder path.
ClientInfoString	Details that identify which client or Exchange component performed the operation.
ClientIPAddress	Client computer IP address.

ClientMachineName	Client computer name.
ClientProcessName	Name of the client application process.
ClientVersion	Client application version.
InternalLogonType	Logon type of the user who performed the operation. Logon types include: • Owner • Delegate • Admin
MailboxOwnerUPN	Mailbox owner user principal name (UPN).
MailboxOwnerSid	Mailbox owner security identifier (SID).
DestMailboxOwnerUPN	Destination mailbox owner UPN, logged for cross-mailbox operations.
DestMailboxOwnerSid	Destination mailbox owner SID, logged for cross-mailbox operations.
DestMailboxOwnerGuid	Destination mailbox owner GUID.
CrossMailboxOperation	Information about whether the operation logged is a cross-mailbox operation (for example, copying or moving messages between mailboxes).
LogonUserDisplayName	Display name of user who is logged on.
DelegateUserDisplayName	Delegate user display name.
LogonUserSid	SID of user who is logged on.
SourceItems	ItemID of mailbox items on which the logged action is performed (for example, move or delete). For operations performed on a number of items, this field is returned as a collection of

	items.
SourceFolders	Source folder GUID.
ItemId	Item ID.
ItemSubject	Item subject.
MailboxGuid	Mailbox GUID.
MailboxResolvedOwnerName	Mailbox user resolved name in the format *DOMAIN\SamAccountName*.
LastAccessed	Time when the operation was performed.
Identity	Audit log entry ID.

More information

- **Administrator access to mailboxes** Mailboxes are considered to be accessed by an administrator only in the following scenarios:
  - In-Place eDiscovery is used to search a mailbox.
  - The New-MailboxExportRequest cmdlet is used to export a mailbox.
  - Microsoft Exchange Server MAPI Editor is used to access the mailbox.
- **Bypassing mailbox auditing logging** Mailbox access by authorized automated processes such as accounts used by third-party tools or accounts used for lawful monitoring can create a large number of mailbox audit log entries and may not be of interest to your organization. You can configure such accounts to bypass mailbox audit logging. For details, see Bypass a user account from mailbox audit logging.
- **Logging mailbox owner actions** For mailboxes such as the Discovery Search Mailbox, which may contain more sensitive information, consider enabling mailbox audit logging for mailbox owner actions such as message deletion.

Return to top

Mailbox audit logging procedures

Applies to: *Exchange Server 2013*

Topic Last Modified: *2010-07-16*

Enable or disable mailbox audit logging for a mailbox

Bypass a user account from mailbox audit logging

Search the mailbox audit log for a mailbox

Create a mailbox audit log search

Enable or disable mailbox audit logging for a mailbox

Messaging policy and compliance > Mailbox audit logging > Mailbox audit logging procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-14*

With mailbox audit logging, you can track logons to a mailbox as well as what actions are taken while the user is logged on. When you enable mailbox audit logging for a mailbox, some actions performed by administrators and delegates are logged by default. None of the actions performed by the mailbox owner are logged. To learn more about mailbox audit logging, see Mailbox audit logging.

Caution:

Auditing of mailbox owner actions can generate a large number of mailbox audit log entries and is therefore disabled by default. We recommend that you only enable auditing of specific owner actions needed to meet business or security requirements.

For additional tasks related to mailbox audit logging, see Mailbox audit logging procedures.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox audit logging" entry in the Messaging policy and compliance permissions topic.
- You can't use the Exchange Administration Center (EAC) to enable or disable mailbox audit logging. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable mailbox audit logging

You can use the Shell to enable or disable mailbox audit logging for a mailbox. This enables or disables logging of all operations specified for administrator, delegates, and the mailbox owner.

This example enables mailbox audit logging for Ben Smith's mailbox.

```
Set-Mailbox -Identity "Ben Smith" -AuditEnabled $true
```

This example disables mailbox audit logging for Ben Smith's mailbox.

```
Set-Mailbox -Identity "Ben Smith" -AuditEnabled $false
```

For detailed syntax and parameter information, see Set-Mailbox.

Use the Shell to configure mailbox audit logging settings for administrator, delegate, and owner access

When mailbox audit logging is enabled for a mailbox, only the administrator, delegate, and owner actions specified in the audit logging configuration for the mailbox are logged.

This example specifies that the `SendAs` or `SendOnBehalf` actions performed by delegate users will be logged for Ben Smith's mailbox.

```
Set-Mailbox -Identity "Ben Smith" -AuditDelegate
SendAs,SendOnBehalf -AuditEnabled $true
```

This example specifies that the `MessageBind` and `FolderBind` actions performed by administrators will be logged for Ben Smith's mailbox.

```
Set-Mailbox -Identity "Ben Smith" -AuditAdmin
MessageBind,FolderBind -AuditEnabled $true
```

This example specifies that the `HardDelete` action performed by the mailbox owner will be logged for Ben Smith's mailbox.

```
Set-Mailbox -Identity "Ben Smith" -AuditOwner HardDelete -
AuditEnabled $true
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To verify that you have successfully enabled mailbox audit logging for a mailbox and specified the correct logging settings for administrator, delegate, or owner access, use the Get-Mailbox cmdlet to retrieve the mailbox audit logging settings for that mailbox.

This example retrieves Ben Smith's mailbox settings and pipes the specified audit settings, including the audit log age limit, to the **Format-List** cmdlet.

```
Get-Mailbox "Ben Smith" | Format-List *audit*
```

Bypass a user account from mailbox audit logging

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-05-21*

When you enable mailbox audit logging for a mailbox, specified mailbox access events (for example, accessing a folder or a message, or permanently deleting a message) are logged. However, access by some authorized accounts, such as accounts used by third-party tools or accounts used for lawful monitoring, can create a large number of mailbox audit log entries and may not be of interest to your organization.

You can configure a user or computer account to bypass mailbox audit logging, so actions taken by that user or account for any mailbox aren't logged. By bypassing trusted user or computer accounts that need frequent access to mailboxes, you can reduce the noise in mailbox audit logs.

Caution:

If you use mailbox audit logging to audit mailbox access and actions, you must monitor mailbox audit bypass associations at regular intervals. If a mailbox audit bypass association is added for an account, the account can access any mailbox in the organization to which it has been assigned permissions, without any mailbox audit logging entries being generated for such access or any actions taken (such as message deletions).

For additional management tasks related to mailbox audit logging, see Mailbox audit logging procedures.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox audit logging" entry in the Messaging policy and compliance permissions topic.
- When an account is configured to bypass mailbox audit logging, access to any mailbox by that account won't be logged. You can't configure an account to bypass the logging of access to a specific mailbox.
- You can't use the Exchange Administration Center (EAC) to enable or disable mailbox audit logging bypass for an account. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to enable or disable mailbox audit logging bypass for an account

For an example of how to enable mailbox audit logging bypass for an account, see Example 1 in Set-MailboxAuditBypassAssociation.

For an example of how to disable mailbox audit logging bypass for an account, see Example 2 in Set-MailboxAuditBypassAssociation.

How do you know this worked?

After you have bypassed a user account from mailbox audit logging, you can check the bypass settings by running the Get-MailboxAuditBypassAssociation cmdlet.

For examples of how to check mailbox audit bypass associations, see Examples in Get-MailboxAuditBypassAssociation.

Search the mailbox audit log for a mailbox

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

You can synchronously search mailbox audit log entries for a mailbox and have the search results displayed in the Shell.

If you want to search mailbox audit logs for multiple mailboxes and have the results emailed to a specified address, you must create a mailbox audit log search instead. For details, see Create a mailbox audit log search.

For additional management tasks related to mailbox audit logging, see Mailbox audit logging procedures.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox audit logging" entry in the Messaging policy and compliance permissions topic.
- By default, mailbox audit logging is disabled for all mailboxes. For each mailbox you want to audit, you must enable audit logging and specify the mailbox owner, delegate, or administrator actions you want to audit. For details, see Enable or disable mailbox audit logging for a mailbox.
- You can't use the EAC to search the mailbox audit log for a mailbox. However, you can use the EAC to run or search for and export a non-owner mailbox access report.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to search the mailbox audit log for a mailbox

For examples of how to use the Shell to search the mailbox audit log for a mailbox, see Examples in Search-MailboxAuditLog.

Create a mailbox audit log search

Messaging policy and compliance > Mailbox audit logging > Mailbox audit logging procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-11*

You can create a mailbox audit log search to asynchronously search one or more mailboxes and have the search results sent by email as an XML file to specified addresses.

To search mailbox audit logs for a single mailbox and have the results displayed in the Shell, see Search the mailbox audit log for a mailbox.

For additional management tasks related to mailbox audit logging, see Mailbox audit logging procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox audit logging" entry in the Messaging policy and compliance permissions topic.
- By default, mailbox audit logging is disabled for all mailboxes. For each mailbox you want to audit, you must enable audit logging and specify the mailbox owner, delegate, or administrator actions you want to audit. For more details, see Enable or disable mailbox audit logging for a mailbox.
- You can't use the EAC to search mailbox audit logs for owner access. The auditing section in EAC includes reports for non-owner mailbox access and also allows you to search for and export non-owner access events.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to create a mailbox audit log search

1. Navigate to **Compliance management** > **Auditing**.
2. In the list view, select **Export mailbox audit logs**.
3. In **Export Mailbox Audit Logs**, complete the following fields, and then click **Export**:
   - **Start date**
   - **End date**
   - **Search these mailboxes or leave blank to find all mailboxes accessed by non-owners**

Caution:
Depending on the number of mailboxes in your organization and the amount of mailbox audit

log data in each mailbox, searching all mailboxes may take a long time.

- **Search for access by**

Select from the following types of access events you want to search:

  - **All non-owners**
  - **External users**
  - **Administrators and delegated users**
  - **Administrators**

- **Send the auditing report to**

Use the Shell to create a mailbox audit log search

For an example of how to use the Shell to create a mailbox audit log search, see Example 1 in **New-MailboxAuditLogSearch**.

Administrator audit logging

Exchange Server 2013 > Messaging policy and compliance >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-08*

You can use administrator audit logging in Microsoft Exchange Server 2013 to log when a user or administrator makes a change in your organization. By keeping a log of the changes, you can trace changes to the person who made the change, augment your change logs with detailed records of the change as it was implemented, comply with regulatory requirements and requests for discovery, and more.

By default, audit logging is enabled in new installations of Exchange 2013.

Contents

What gets audited

Cmdlets that are run directly in the Exchange Management Shell are audited. In addition, operations performed using the Exchange admin center (EAC) are also logged because those operations run cmdlets in the background.

Cmdlets, regardless of where they're run, are audited if a cmdlet is on the cmdlet auditing list and one or more parameters on that cmdlet are on the parameter auditing list. **Get-** and **Search-** cmdlets aren't logged. Audit logging is intended to show what actions have been taken to modify objects in an Exchange organization rather than what objects have been viewed.

Important:
A cmdlet might not be logged if an error occurs before the cmdlet calls the Admin Audit Log cmdlet extension agent. If an error occurs after the Admin Audit Log agent is called, the cmdlet is logged along with the associated error. For more information, see the Admin Audit Log Agent section later in this topic. Changes made using Microsoft Exchange Server 2010 management tools are logged; however, changes using Microsoft Exchange Server 2007 management tools aren't logged. Changes to the audit log configuration are refreshed every 60 minutes on computers that have the Shell open at the time a configuration change is made. If you want to apply the changes immediately, close and then open the Shell again on each computer. A command may take up to 15 minutes after it's run to appear in audit log search results. This is because audit log entries must be indexed before they can be searched. If a command doesn't appear in the administrator audit log, wait a few minutes and run the search again.

Audit logging configuration

By default, if audit logging is enabled, a log entry is created every time any cmdlet, other than a **Get-** or **Search-** cmdlet, is run. If you don't want to audit every cmdlet that's run, you can configure audit logging to audit only the cmdlets and parameters you're interested in. You configure audit logging with the **Set-AdminAuditLogConfig** cmdlet. The parameters referenced in the following sections are used with this cmdlet.

Important:
Changes to the administrator audit log configuration are always logged, regardless of whether the **Set-AdministratorAuditLog** cmdlet is included in the list of cmdlets being audited and whether audit logging is enabled or disabled.

When a command is run, Exchange inspects the cmdlet that was used. If the cmdlet that was run matches any of the cmdlets provided with the *AdminAuditLogConfigCmdlets* parameter, Exchange then checks the parameters specified in the *AdminAuditLogConfigParameters* parameter. If at least one or more parameters from the parameters list are matched, Exchange logs the cmdlet that was run in the mailbox specified using the *AdminAuditLogMailbox* parameter. The following sections contain more information about each aspect of the audit logging configuration.

For more information about managing audit logging configuration, see Manage administrator audit logging.

Return to top

Cmdlets

You can control which cmdlets are audited by providing a list of cmdlets, and their parameters, that you want to log. When you configure audit logging, you can specify to audit every cmdlet, or you can specify the cmdlets you want to audit by using the *AdminAuditLogConfigCmdlets* parameter. You can specify full cmdlet names, such as **New-Mailbox**, or you can specify partial cmdlet names and enclose those names in wildcard characters, such as an asterisk (*). For example, if you want to log when any cmdlet that contains the string `Transport` runs, you can specify a value of `*Transport*`. You can use a mix of full cmdlet names and partial cmdlet names at the same time to tailor the audit logging configuration to your needs.

Parameters

In addition to specifying which cmdlets you want to log, you can also indicate that cmdlets should only be logged if certain parameters on those cmdlets are used. Use the *AdminAuditLogConfigParameters* parameter to specify which parameters should be logged. As with cmdlets, you can specify full parameter names, such as `Database`, or partial parameter names enclosed in wildcard characters (*), such as `*Address*`, or a combination of both.

Audit log age limit

By default, audit logging is configured to store audit log entries for 90 days. After 90 days, the audit log entry is deleted. You can change the audit log age limit using the *AdminAuditLogAgeLimit* parameter. You can specify the number of days, hours, minutes, and seconds that audit log entries should be kept. To specify a value, use the format `dd.hh:mm:ss` where the following applies:

- **dd** The number of days to keep the audit log entry.
- **hh** The number of hours to keep the audit log entry.
- **mm** The number of minutes to keep the audit log entry.
- **ss** The number of seconds to keep the audit log entry.

You must specify multiple years using the `dd` field. For example, 365 days equals one year; 730 days equals two years; 913 days equals two years and six months. For example, to set the audit log age limit to two years and six months, use the syntax `913.00:00:00`.

Caution:

You can set the audit log age limit to a value that's less than the current age limit. If you do this, any audit log entry whose age exceeds the new age limit is deleted.
If you set the age limit to 0, Exchange deletes all the entries in the audit log.
We recommend that you grant permissions to configure the audit log age limit only to highly trusted users.

Verbose logging

By default, the administrator audit log records only the cmdlet name, cmdlet parameters (and values specified), the object that was modified, who ran the cmdlet, when the cmdlet was run, and on what server the cmdlet was run. The administrator audit log doesn't log what properties were modified on the object. If you want the audit log to also include the properties of the object that were modified, you can enable verbose logging by setting the *LogLevel* parameter to `verbose`. When you enable verbose logging, in addition to the information logged by default, the properties modified on an object, including their old and new values, are included in the audit log.

Test cmdlets

Cmdlets that begin with the verb **Test** aren't logged by default. You can indicate that **Test** cmdlets should be logged by setting the *TestCmdletLoggingEnabled* parameter to `$true`. Although you can enable logging of test cmdlets, we recommend that you do this only for short periods of time because test cmdlets can produce a large amount of information.

Audit logs

Each time a cmdlet is logged, an audit log entry is created. Audit logs are stored in a hidden, dedicated arbitration mailbox that can only be accessed using the EAC or the **Search-AdminAuditLog** or **New-AdminAuditLogSearch** cmdlet. It can't be opened using Microsoft Outlook Web App or Microsoft Outlook. The following sections provide information about the following:

- What's included in the logs
- Reports available on the EAC **auditing** page
- Audit log search cmdlets

Audit log contents

Each audit log entry contains the information described in the following table. The audit log contains one or more audit log entries. The number of audit log entries is controlled by the audit log age limit specified using the **Set-AdminAuditLogConfig** cmdlet. Any audit log entry that exceeds the age limit is deleted.

Audit log entry fields

Field	Description
`RunspaceId`	This field is used internally by Exchange.
`ObjectModified`	This field contains the object that was modified by the cmdlet specified in the `CmdletName` field.
`CmdletName`	This field contains the name of the cmdlet that

	was run by the user in the `Caller` field.
`CmdletParameters`	This field contains the parameters that were specified when the cmdlet in the `CmdletName` field was run. Also stored in this field, but not visible in the default output, is the value specified with the parameter, if any. For more information about how to access the additional information in this field, see Search the role group changes or administrator audit logs.
`ModifiedProperties`	This field contains the properties that were modified on the object in the `ObjectModified` field. Also stored in this field, but not visible in the default output, are the old value of the property and the new value that was stored. For more information about how to access the additional information in this field, see Search the role group changes or administrator audit logs. **Important:** This field is only populated if the *LogLevel* parameter on the **Set-AdminAuditLogConfig** cmdlet is set to `Verbose`.
`Caller`	This field contains the user account of the user who ran the cmdlet in the `CmdletName` field.
`Succeeded`	This field specifies whether the cmdlet in the `CmdletName` field ran successfully. The value is either `True` or `False`.
`Error`	This field contains the error message generated if the cmdlet in the `CmdletName` field failed to complete successfully.
`RunDate`	This field contains the date and time when the

	cmdlet in the `CmdletName` field was run. The date and time are stored in Coordinated Universal Time (UTC) format.
`OriginatingServer`	This field indicates the server on which the cmdlet specified in the `CmdletName` field was run.
`Identity`	This field is used internally by Exchange.
`IsValid`	This field is used internally by Exchange.
`ObjectState`	This field is used internally by Exchange.

Return to top

EAC auditing reports

The **auditing** page in the EAC has several reports that provide information about various types of compliance and administrative configuration changes. The following reports provide information about configuration changes in your organization:

- **Administrator role group report** This report enables you to search for changes to management role groups that you specify within a specified timeframe. The results that are returned include the role groups that have been changed, who changed them and when, and what changes were made. A maximum of 3,000 entries can be returned. If your search might return more than 3,000 entries, use the **Administrator audit log** report or the **Search-AdminAuditLog** cmdlet.
- **Administrator audit log** This report enables you to export the audit log entries recorded within a specified timeframe to a XML file and then send the file via email to a recipient you specify. For more information about the contents of the XML file, see Administrator audit log structure.

For information about how to use these reports, see Search the role group changes or administrator audit logs.

For information about the other reports included on the **auditing** page see Exchange auditing reports.

Search-AdminAuditLog cmdlet

When you run the **Search-AdminAuditLog** cmdlet, all the audit log entries that match the search criteria you specify are returned. You can specify the following search criteria:

- **Cmdlets** Specifies the cmdlets you want to search for in the administrator audit log.
- **Parameters** Specifies the parameters, separated by commas, you want to search for in the

administrator audit log. You can only search for parameters if you specify a cmdlet to search for.

- **End date** Scopes the administrator audit log results to log entries that occurred on or before the specified date.
- **Start date** Scopes the administrator audit log results to log entries that occurred on or after the specified date.
- **Object IDs** Specifies that only administrator audit log entries that contain the specified changed objects should be returned
- **User IDs** Specifies that only the administrator audit log entries that contain the specified IDs of the user who ran the cmdlet should be returned.
- **Successful completion** Specifies whether only administrator audit log entries that indicated a success or failure should be returned.

Each audit log entry returned contains the information described in the table in Audit Log Contents. By default, only the first 1,000 log entries that match the criteria you specify are returned. However, you can override this default and return more or fewer entries using the *ResultSize* parameter. You can specify a value of `Unlimited` with the *ResultSize* parameter to return all log entries that match the specified criteria.

For information about how to use the **Search-AdminAuditLog** cmdlet, see Search the role group changes or administrator audit logs.

New-AdminAuditLogSearch cmdlet

The **New-AdminAuditLogSearch** cmdlet searches the audit log just like the **Search-AdminAuditLog** cmdlet. However, instead of displaying the results of the audit log search in the Shell, the **New-AdminAuditLogSearch** cmdlet performs the search and then sends the results of the search to a recipient you specify via an email message. The results are included as an XML attachment to the email message.

You can use the same search criteria with the **New-AdminAuditLogSearch** cmdlet that's used on the **Search-AdminAuditLog** cmdlet. For a list of the search criteria, see Search-AdminAuditLog Cmdlet.

After you run the **New-AdminAuditLogSearch** cmdlet, Exchange may take up to 15 minutes to deliver the report to the specified recipient. The XML file attached report can be a maximum of 10 megabytes (MB). The XML file contains the same information described in the table in Audit Log Contents. For more information about the structure of the XML file, see Administrator audit log structure.

Note:

Outlook Web App doesn't allow you to open XML attachments by default. You can either configure Exchange to allow XML attachments to be viewed using Outlook Web App, or you can use another email client, such as Microsoft Outlook, to view the attachment. For information about how to configure Outlook Web App to allow you to view an XML attachment, see View or configure Outlook Web App virtual directories.

For information about how to use the **New-AdminAuditLogSearch** cmdlet, see Search the role group changes or administrator audit logs.

Return to top

Manual audit log entries

In addition to logging Exchange cmdlets when they're run, Exchange 2013 enables you to manually write log entries to the audit log. Exchange 2013 supports this using the **Write-AdminAuditLog** cmdlet. Situations where you might want to add a manual log entry include the following:

- Custom script entry and exit
- Change control information
- Maintenance start and end times

With the **Write-AdminAuditLog** cmdlet, you specify a string of text to include in the audit log using the *Comment* parameter. The *Comment* parameter accepts an alphanumeric string up to 500 characters. Included in the manual audit log entry along with the comment string is all of the same information captured when an Exchange cmdlet is logged. For a description of each field included in the audit log, see the table in Audit Log Contents.

You can retrieve manual audit log entries the same way as any other log entry, using the EAC **auditing** page or using the **Search-AdminAuditLog** or **New-AdminAuditLogSearch** cmdlets.

To view the contents of the *Comment* parameter on the **Write-AdminAuditLog** cmdlet in a manual audit log entry, see Search the role group changes or administrator audit logs.

Active Directory replication

Administrator audit logging relies on Active Directory replication to replicate the configuration settings you specify to the domain controllers in your organization. Depending on your replication settings, the changes you make may not be immediately applied to all servers running Exchange 2013 or Exchange 2010 in your organization.

Admin Audit Log agent

The Admin Audit Log built-in cmdlet extension agent performs administrator audit logging of cmdlet operations in Exchange 2013. This agent reads the audit log configuration and then performs an evaluation of each cmdlet run in your organization. If the criteria you've specified in the audit log configuration matches the cmdlet that's being run, the agent generates an audit log.

The Admin Audit Log agent is enabled by default, which is required for audit logging to function. It can't be disabled, and its priority can't be changed. For more information about cmdlet extension agents, see Cmdlet extension agents.

Administrator audit log structure

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-19*

Administrator audit logs contain a record of all the cmdlets and parameters that have been run in the Exchange Management Shell and by the Exchange Administration Center (EAC). They're created on-demand when you run the Administrator audit log report in the EAC, or when you run the **New-AdminAuditLogSearch** cmdlet in the Shell. For more information about audit logs, see Administrator audit logging.

The audit logs are XML files and can contain multiple audit log entries. The following table describes each XML tag and its associated attributes.

Looking for management tasks related to Administrator audit logs? See Manage administrator audit logging.

Audit log XML tags and attributes

Element	Attribute	Description
`<?xml version="1.0" encoding="utf-8"?>`	`N/A`	This is the XML document declaration tag. It's included in every audit log XML file and contains the XML version number and the character encoding value.
`SearchResults`	`N/A`	This tag contains all the audit log entries in the XML file. The `Event` tag is a child of this tag. There is only one `SearchResults` tag per XML file.
`Event`		This tag contains the audit log entry for an individual cmdlet. This tag contains the `Caller`, `Cmdlet`, `ObjectModified`, `RunDate`, `Succeeded`, `Error`, and

		`OriginatingServer` attributes. The `CmdletParameters` and `ModifiedProperties` tags are children of this tag. There is one `Event` tag per audit log entry.
	`Caller`	This attribute contains the user account of the user who ran the cmdlet in the `Cmdlet` attribute.
	`Cmdlet`	This attribute contains the name of the cmdlet that was run by the user in the `Caller` attribute.
	`ObjectModified`	This attribute contains the object that was modified by the cmdlet specified in the `Cmdlet` attribute. The `ModifiedProperties` tag shows which properties were modified on this object.
	`RunDate`	This attribute contains the date and time when the cmdlet in the `Cmdlet` attribute was run.
	`Succeeded`	This attribute specifies whether the cmdlet in the `Cmdlet` attribute ran successfully. The value is either `True` or `False`.
	`Error`	This attribute contains the error message generated if the cmdlet in the `Cmdlet` attribute failed to complete successfully.

		If no error was encountered, the value is set to `None`.
	`OriginatingServer`	This attribute contains the server on which the cmdlet specified in the `Cmdlet` attribute was run.
`CmdletParameters`	`N/A`	This tag contains all of the parameters specified when the cmdlet was run. The `Parameter` tag is a child of this tag. There is one `CmdletParameters` tag per `Event` tag.
`Parameter`		This tag contains an individual parameter that was specified when the cmdlet was run. This tag contains the `Name` and `Value` attributes. There can be multiple `Parameter` tags per `CmdletParameters` tag.
	`Name`	This attribute contains the name of the parameter that was specified on the cmdlet that was run.
	`Value`	This attribute contains the value that was provided on the parameter specified in the `Name` attribute.
`ModifiedProperties`	`N/A`	This tag contains all of the properties that were modified

		by the cmdlet that was run. The `Property` tag is a child of this tag. There is one `ModifiedProperties` tag per `Event` tag. **Important:** This tag is only populated if the *LogLevel* parameter on the **Set-AdminAuditLogConfig** cmdlet is set to `Verbose`.
`Property`		This tag contains an individual property that was specified when the cmdlet was run. This tag contains the `Name`, `OldValue`, and `NewValue` attributes. There can be multiple `Property` tags per `ModifiedProperties` tag.
	`Name`	This attribute contains the name of the property that was modified when the cmdlet was run.
	`OldValue`	This attribute contains the value that was contained in the property specified in the `Name` attribute before it was changed.
	`NewValue`	This attribute contains the value that the property in the `Name` attribute was changed to.

Example audit log entry

The following is an example of a typical audit log entry. Based on the information in log entry, we know the following occurred:

- On 10/18/2012 at 3:48 P.M. Pacific Daylight Time (UTC-7), the user `Administrator` ran the cmdlet **Set-Mailbox**.
- The two following parameters were provided when the **Set-Mailbox** cmdlet was run:
  - *Identity* with a value of `david`
  - *ProhibitSendReceiveQuota* with a value of `10GB`
- The two following properties on the object `david` were modified:

Note:

The modified properties are saved to the audit log because the *LogLevel* parameter on the `Set-AdminAuditLogConfig` cmdlet was set to `Verbose` in this example.

  - *ProhibitSendReceiveQuota* with a new value of `10GB`, which replaced the old value of `35GB`
- The operation completed successfully without any errors.

```
<?xml version="1.0" encoding="utf-8"?>
<SearchResults>
  <Event Caller="corp.e15a.contoso.com/Users/Administrator"
Cmdlet="Set-Mailbox" ObjectModified="corp.e15a.contoso.com/
Users/david" RunDate="2012-10-18T15:48:15-07:00"
Succeeded="true" Error="None" OriginatingServer="WIN8MBX
(15.00.0516.032)">
    <CmdletParameters>
      <Parameter Name="Identity" Value="david" />
      <Parameter Name="ProhibitSendReceiveQuota" Value="10
GB (10,737,418,240 bytes)" />
    </CmdletParameters>
    <ModifiedProperties>
      <Property Name="ProhibitSendReceiveQuota"
OldValue="35 GB (37,580,963,840 bytes)" NewValue="10 GB
(10,737,418,240 bytes)" />
    </ModifiedProperties>
  </Event>
</SearchResults>
```

Manage administrator audit logging

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-05-17*

Administrator audit logging in Microsoft Exchange Server 2013 enables you to create a log entry each time a specified cmdlet is run. Log entries provide you with information about what cmdlet was run, which parameters were used, who ran the cmdlet, and what objects were affected. For more information about administrator audit logging, see Administrator audit logging.

What do you need to know before you begin?

- Estimated time to complete each procedure: less than 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Administrator audit logging" entry in the Exchange and Shell infrastructure permissions topic.
- Administrator audit logging relies on Active Directory replication to replicate the configuration settings you specify to the domain controllers in your organization. Depending on your replication settings, the changes you make may not be immediately applied to all Exchange 2013 servers in your organization.
- Changes to the audit log configuration are refreshed every 60 minutes on computers that have the Shell open at the time a configuration change is made. If you want to apply the changes immediately, close and then open the Shell again on each computer.
- A command may take up to 15 minutes after it's run to appear in audit log search results. This is because audit log entries must be indexed before they can be searched. If a command doesn't appear in the administrator audit log, wait a few minutes and run the search again.
- You must use the Shell to perform these procedures.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Specify the cmdlets to be audited

By default, audit logging creates a log entry for every cmdlet that's run. If you're enabling audit logging for the first time and want this behavior, you don't have to change the cmdlet audit list. If you've previously specified cmdlets to audit and now want to audit all cmdlets, you can audit all cmdlets by specifying the asterisk (*) wildcard character with the *AdminAuditLogCmdlets* parameter on the **Set-AdminAuditLogConfig** cmdlet, as shown in the following command.

```
Set-AdminAuditLogConfig -AdminAuditLogCmdlets *
```

You can specify which cmdlets to audit by providing a list of cmdlets using the *AdminAuditLogCmdlets* parameter. When you provide the list of cmdlets to audit, you can provide single cmdlets, cmdlets with the asterisk (*) wildcard characters, or a mix of both. Each entry in the list is separated by commas. The following values are all valid:

- `New-Mailbox`
- `*TransportRule`
- `*Management*`
- `Set-Transport*`

This example audits the cmdlets specified in the preceding list.

```
Set-AdminAuditLogConfig -AdminAuditLogCmdlets New-Mailbox,
*TransportRule, *Management*, Set-Transport*
```

For detailed syntax and parameter information, see Set-AdminAuditLogConfig.

Specify the parameters to be audited

By default, audit logging creates a log entry for every cmdlet that's run, regardless of the parameters specified. If you're enabling audit logging for the first time and want this behavior, you don't have to change the parameter audit list. If you've previously specified parameters to audit and now want to audit all parameters, you can do so by specifying the asterisk (*) wildcard character with the *AdminAuditLogParameters* parameter on the **Set-AdminAuditLogConfig** cmdlet, as shown in the following command.

```
Set-AdminAuditLogConfig -AdminAuditLogParameters *
```

You can specify which parameters you want to audit by using the *AdminAuditLogParameters* parameter. When you provide the list of parameters to audit, you can provide single parameters, parameters with the asterisk (*) wildcard characters, or a mix of both. Each entry in the list is separated by commas. The following values are all valid:

- `Database`
- `*Address*`
- `Custom*`
- `*Region`

Note:

For an audit log entry to be created when a command is run, the command must include at least one or more parameters that exist on at least one or more cmdlets specified with the *AdminAuditLogCmdlets* parameter.

This example audits the parameters specified in the preceding list.

```
Set-AdminAuditLogConfig -AdminAuditLogParameters Database,
*Address*, Custom*, *Region
```

For detailed syntax and parameter information, see Set-AdminAuditLogConfig.

Specify the audit log age limit

The audit log age limit determines how long audit log entries will be retained. When a log entry exceeds the age limit, it's deleted. The default is 90 days.

You can specify the number of days, hours, minutes, and seconds that audit log entries should be kept. To specify a value, use the format dd.hh.mm:ss where the following applies:

- **dd** Number of days to keep the audit log entry
- **hh** Number of hours to keep the audit log entry
- **mm** Number of minutes to keep the audit log entry
- **ss** Number of seconds to keep the audit log entry

Caution:

You can set the audit log age limit to a value that's less than the current age limit. If you do this, any audit log entry whose age exceeds the new age limit will be deleted.
If you set the age limit to 0, Exchange deletes all the entries in the audit log.
We recommend that you grant permissions to configure the audit log age limit only to highly trusted users.

This example specifies an age limit of two years and six months.

```
Set-AdminAuditLogConfig -AdminAuditLogAgeLimit 913.00:00:00
```

For detailed syntax and parameter information, see Set-AdminAuditLogConfig.

Enable or disable logging of Test cmdlets

Cmdlets that start with the verb **Test** aren't logged by default. This is because **Test** cmdlets can generate a significant amount of data in a short time. Only enable the logging of **Test** cmdlets for short periods of time.

This command enables the logging of **Test** cmdlets.

```
Set-AdminAuditLogConfig -TestCmdletLoggingEnabled $True
```

This command disables the logging of **Test** cmdlets.

```
Set-AdminAuditLogConfig -TestCmdletLoggingEnabled $False
```

For detailed syntax and parameter information, see Set-AdminAuditLogConfig.

Disable administrator audit logging

To disable administrator audit logging, use the following command.

```
Set-AdminAuditLogConfig -AdminAuditLogEnabled $False
```

Enable administrator audit logging

To enable administrator audit logging, use the following command.

```
Set-AdminAuditLogConfig -AdminAuditLogEnabled $True
```

View administrator audit logging settings

To view the administrator audit logging settings that you've configured for your organization, use the following command.

```
Get-AdminAuditLogConfig
```

Exchange auditing reports

Exchange Server 2013 > Messaging policy and compliance >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-26*

Use audit logging to troubleshoot configuration issues by tracking specific changes made by administrators and to help you meet regulatory, compliance, and litigation requirements. Microsoft Exchange provides two types of audit logging:

- *Administrator audit logging* records any action, based on an Exchange Management Shell cmdlet, performed by an administrator. This can help you troubleshoot configuration issues or identify the cause of security-related or compliance-related problems. In Exchange Online, actions performed by Microsoft administrators and delegated administrators, are also recorded.
- *Mailbox audit logging* records whenever a mailbox is accessed by someone other than the person who owns the mailbox. This can help you determine who has accessed a mailbox and what they've done.

This topic explains the following:

- Export audit logs
- Run auditing reports
- Configure audit logging

- Enable mailbox audit logging
- Give users access to auditing reports
- Configure Outlook Web App to allow XML attachments

Export audit logs

On the **Compliance Management** > **Auditing** page in the Exchange admin center (EAC), you can search for and export entries from the administrator audit log and the mailbox audit log.

- **Export the administrator audit log** Any action performed by an administrator that's based on a Shell cmdlet and doesn't begin with the verbs **Get**, **Search**, or **Test** is logged in the administrator audit log. Audit log entries include the cmdlet that was run, the parameter and values used with the cmdlet, and when the operation was successful. You can search for and export entries from the administrator audit log. When you export your search results, Microsoft Exchange saves them in an XML file and attaches it to an email message. For more information, see:
  - Search the role group changes or administrator audit logs
  - **View and export the external admin audit log**
- **Export mailbox audit logs** When mailbox audit logging is enabled for a mailbox, Microsoft Exchange stores a record of actions performed on mailbox data by non-owners in the mailbox audit log, which is stored in a hidden folder in the mailbox being audited. Entries in this log indicate if the mailbox was accessed by someone other than the owner, who accessed the mailbox and when, the actions performed by the non-owner, and whether the action was successful. When you search for entries in the mailbox audit log and export them, Microsoft Exchange saves the search results in an XML file and attaches it to an email message. For more information, see Export mailbox audit logs.

Note:

By default, audit log entries are kept for 90 days. When an entry is older than 90 days, it's deleted. This setting can't be changed in a cloud-based organization. However, it can be changed in an on-premises Exchange organization by using the **Set-AdminAuditLog** cmdlet.

Run auditing reports

When you run any of the following reports on the **Auditing** page in the EAC, the results are displayed in the details pane of the report.

- **Run a non-owner mailbox access report** Use this report to find mailboxes that have been accessed by someone other than the person who owns the mailbox. For more information, see Run a non-owner mailbox access report.
- **Run an administrator role group report** Use this report to search for changes made to administrator role groups. For more information, see Search the role group changes or administrator audit logs.
- **Run an in-place discovery and hold report** Use this report to find mailboxes that have been put on, or removed from, In-Place Hold. For more information, see:
  - In-Place Hold

    - In-Place eDiscovery
- **Run a per-mailbox litigation hold report** Use this report to find mailboxes that were put on, or removed from, litigation hold. For more information, see.
    - Run a per-mailbox litigation hold report
    - Place a mailbox on Litigation Hold
- **Run the admin audit log report** Use this report to view entries from the administrator audit log. Instead of exporting the administrator audit log, which can take up to 24 hours to receive in an email message, you can run this report in the EAC. This report records configuration changes made by administrators in your organization. Up to 5000 entries will be displayed on multiple pages. To narrow the search, you can specify a date range. For more information, see:
    - View the administrator audit log
    - Administrator audit logging
- **Run the external admin audit log report** This report is only available in Exchange Online and Exchange Online Protection. Actions performed by Microsoft administrators or delegated administrators are logged in the administrator audit log. Use the external admin audit log report to search for and view the actions that administrators outside your organization performed on the configuration of your Exchange Online organization. For more information, see **View and export the external admin audit log**.

Configure audit logging

Before you can run auditing reports and export audit logs, you have to configure audit logging for your organization.

Enable mailbox audit logging

You have to enable mailbox audit logging for each mailbox that you want to run a non-owner mailbox access report for. If mailbox audit logging isn't enabled for a mailbox, you won't get any results when you run a report for it or export the mailbox audit log.

To enable mailbox audit logging for a single mailbox, run the following command in the Shell.

```
Set-Mailbox <Identity> -AuditEnabled $true
```

To enable mailbox auditing for all user mailboxes in your organization, run the following commands.

```
$UserMailboxes = Get-mailbox -Filter {(RecipientTypeDetails
-eq 'UserMailbox')}
$UserMailboxes | ForEach {Set-Mailbox $_.Identity -
AuditEnabled $true}
```

Give users access to Auditing reports

By default, administrators can access and run any of the reports on the Auditing page in the EAC. However, other users, such as a records manager or legal staff, have to be assigned the necessary permissions.

The easiest way to give users access is to add them to the Records Management role group. You can also use the Shell to give a user access to the **Auditing** page in the EAC by assigning the Audit Logs role to the user.

Add a user to the Records Management role group

1. Go to **Permissions** > **Admin Roles**.
2. In the list of role groups, click **Records Management**, and then click **Edit** .
3. Under **Members**, click **Add +**.
4. In the **Select Members** dialog box, select the user. You can search for a user by typing all or part of a display name, and then clicking **Search** . You can also sort the list by clicking the **Name** or **Display Name** column headings.
5. Click **Add +** and then click **OK** to return to the role group page.
6. Click **Save** to save the change to the role group.

In the details pane, the user is listed under **Members** and can access the Auditing page in the EAC, run auditing reports, and export audit logs.

Assign the Audit Logs role to a user

Run the following command to assign the Audit Logs role to a user.

```
New-ManagementRoleAssignment -Role "Audit Logs" -User
<Identity>
```

This enables the user to select **Compliance Management** > **Auditing** in the EAC to run any of the reports. The user can also export the mailbox audit log, and export and view the administrator audit log.

Note:

To allow a user to run auditing reports but not to export audit logs, use the preceding command to assign the View-Only Audit Logs role.

Configure Outlook Web App to allow XML attachments

When you export the mailbox audit log or administrator audit log, Microsoft Exchange attaches the audit log, which is an XML file, to an email message. However, Outlook Web App blocks XML attachments by default. If you want to use Outlook Web App to access these audit logs, you have to configure Outlook Web App to allow XML attachments.

Run the following command to allow XML attachments in Outlook Web App.

```
Set-OwaMailboxPolicy -Identity Default -AllowedFileTypes
'.rpmsg','.xlsx','.xlsm','.xlsb','.tiff','.pptx','.pptm','.
ppsx','.ppsm','.docx','.docm','.zip','.xls','.wmv','.wma','
.wav','.vsd','.txt','.tif','.rtf','.pub','.ppt','.png','.pd
f','.one','.mp3','.jpg','.gif','.doc','.bmp','.avi','.xml'
```

Export mailbox audit logs

Exchange Server 2013 > Messaging policy and compliance > Exchange auditing reports >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-11-13*

When mailbox auditing is enabled for a mailbox, Microsoft Exchange logs information in the *mailbox audit log* whenever a user other than the owner accesses the mailbox. Each log entry includes information about who accessed the mailbox and when, the actions performed by the non-owner, and whether the action was successful. Entries in the mailbox audit log are retained for 90 days by default. You can use the mailbox audit log to determine if a user other than the owner has accessed a mailbox.

When you export entries from mailbox audit logs, Microsoft Exchange saves the entries in an XML file and attaches it to an email message sent to the specified recipients.

What do you need to know before you begin?

- Estimated time to complete each procedure: Times are variable. In Exchange Online, the mailbox audit log is sent within a few days after you export it.
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Configure mailbox audit logging

You have to enable mailbox audit logging on each mailbox that you want to audit before you can

export and view mailbox audit logs. You also have to configure Microsoft Outlook Web App to allow XML attachments to use Outlook Web App to access the audit log.

Step 1: Enable mailbox audit logging

You have to enable mailbox audit logging for each mailbox that you want to run a non-owner mailbox access report for. If mailbox audit logging isn't enabled for a mailbox, you won't get any results for that mailbox when you export the mailbox audit log.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox audit logging" entry in the Messaging policy and compliance permissions topic.

To enable mailbox audit logging for a single mailbox, run the command in the Shell.

```
Set-Mailbox <Identity> -AuditEnabled $true
```

To enable mailbox audit logging for all user mailboxes in your organization, run the following commands.

```
$UserMailboxes = Get-mailbox -Filter {(RecipientTypeDetails
-eq 'UserMailbox')}

$UserMailboxes | ForEach {Set-Mailbox $_.Identity -
AuditEnabled $true}
```

Step 2: Configure Outlook Web App to allow XML attachments

When you export the mailbox audit log, Microsoft Exchange attaches the audit log, which is an XML file, to an email message. However, Outlook Web App blocks XML attachments by default. To access the exported audit log, you have to use Microsoft Outlook or configure Outlook Web App to allow XML attachments.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Outlook Web App mailbox policies" entry in the Clients and mobile devices permissions topic.

Run the following command to allow XML attachments in Outlook Web App.

```
Set-OwaMailboxPolicy -Identity Default -AllowedFileTypes
'.rpmsg','.xlsx','.xlsm','.xlsb','.tiff','.pptx','.pptm','.
ppsx','.ppsm','.docx','.docm','.zip','.xls','.wmv','.wma','
.wav','.vsd','.txt','.tif','.rtf','.pub','.ppt','.png','.pd
f','.one','.mp3','.jpg','.gif','.doc','.bmp','.avi','.xml'
```

Note:

In Exchange Online, use the value `OwaMailboxPolicy-Default` for the *Identity* parameter.

How do you know this worked?

To verify that you've successfully configured mailbox audit logging, do the following:

1. Run the following command to verify that audit logging is configured for mailboxes.

```
Get-Mailbox | FL Name,AuditEnabled
```

A value of `True` for the *AuditEnabled* property verifies that audit logging is enabled.

2. Run the following command to verify that XML attachments are allowed in Outlook Web App for your organization.

```
Get-OwaMailboxPolicy | Select-Object -ExpandProperty AllowedFileTypes
```

Verify that `.xml` is included in the list of allowed file types.

Export the mailbox audit log

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "View-only administrator audit logging" entry in the Exchange and Shell infrastructure permissions topic.

1. In the Exchange Administration Center (EAC), navigate to **Compliance Management** > **Auditing**.
2. Click **Export mailbox audit logs**.
3. Configure the following search criteria for exporting the entries from the mailbox audit log:
   - **Start and end dates** Select the date range for the entries to include in the exported file.
   - **Mailboxes to search audit log for** Select the mailboxes to retrieve audit log entries for.
   - **Type of non-owner access** Select one of the following options to define the type of non-owner access to retrieve entries for:
     - **All non-owners** Search for access by administrators and delegated users inside your organization, and by Microsoft datacenter administrators in Exchange Online.
     - **External users** Search for access by Microsoft datacenter administrators.
     - **Administrators and delegated users** Search for access by administrators and delegated users inside your organization.
     - **Administrators** Search for access by administrators in your organization.
   - **Recipients** Select the users to send the mailbox audit log to.
4. Click **Export**.

Microsoft Exchange retrieves entries in the mailbox audit log that meet your search criteria, saves them to a file named SearchResult.xml, and then attaches the XML file to an email message sent to the recipients that you specified.

How do you know this worked?

Sign in to the mailbox where the mailbox audit log was sent. If you've successfully exported the audit log, you'll receive a message sent from Exchange. The audit log will be attached to this

message. As previously stated, in Exchange Online, it may take a few days to receive this message.

View the mailbox audit log

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "View-only administrator audit logging" entry in the Exchange and Shell infrastructure permissions topic.

To open or save the SearchResult.xml file:

1. Sign in to the mailbox where the mailbox audit log was sent.
2. In the Inbox, open the message with the XML file attachment sent by Microsoft Exchange. Notice that the body of the email message contains the search criteria.
3. Click the attachment and select to open or save the XML file.

Entries in the mailbox audit log

The following example shows an entry from the mailbox audit log contained in the SearchResult.xml file. Each entry is preceded by the **<Event>** XML tag and ends with the **</Event>** XML tag. This entry shows that the administrator purged the message with the subject, "**Notification of litigation hold**" from the Recoverable Items folder in David's mailbox on April 30, 2010.

```
<Event MailboxGuid="6d4fbdae-e3ae-4530-8d0b-f62a14687939"
  Owner="PPLNSL-dom\david50001-1363917750"
  LastAccessed="2010-04-30T11:01:55.140625-07:00"
  Operation="HardDelete"
  OperationResult="Succeeded"
  LogonType="Admin"
 FolderId="0000000073098C3277988F4CB882F5B82EBF64610100A7C3
17F68C24304BBD18ABE1F185E79B00000026BD4F0000"
  FolderPathName="\Recoverable Items\Deletions"
  ClientInfoString="Client=OWA;Action=ViaProxy"
  ClientIPAddress="10.196.241.168"
  InternalLogonType="Owner"
  MailboxOwnerUPN=david@contoso.com
  MailboxOwnerSid="S-1-5-21-290112810-296651436-1966561949-
1151"
  CrossMailboxOperation="false"
  LogonUserDN="Administrator"
  LogonUserSid="S-1-5-21-290112810-296651436-1966561949-
1149">
  <SourceItems>
```

```
   <ItemId="0000000073098C3277988F4CB882F5B82EBF64610700A7C
317F68C24304BBD18ABE1F185E79B00000026BD4F0000A7C317F68C2430
4BBD18ABE1F185E79B00000026BD540"
    Subject="Notification of litigation hold"
    FolderPathName="\Recoverable Items\Deletions" />
  </SourceItems>
</Event>
```

Useful fields in the mailbox audit log

Watch for these fields. They can help you identify specific information about each instance of non-owner access of a mailbox.

Field	Description
Owner	The owner of the mailbox that was accessed by a non-owner.
LastAccessed	The date and time when the mailbox was accessed.
Operation	The action that was performed by the non-owner. For more information, see the "What gets logged in the mailbox audit log?" section in Learn more about running a non-owner mailbox access report.
OperationResult	Whether the action performed by the non-owner succeeded or failed.
LogonType	The type of non-owner access. These include administrator, delegate, and external.
FolderPathName	The name of the folder that contained the message that was affected by the non-owner.
ClientInfoString	Information about the mail client used by the non-owner to access the mailbox.
ClientIPAddress	The IP address of the computer used by the non-owner to access the mailbox.

InternalLogonType	The logon type of the account used by the non-owner to access this mailbox.
MailboxOwnerUPN	The email address of the mailbox owner.
LogonUserDN	The display name of the non-owner.
Subject	The subject line of the email message that was affected by the non-owner.

How do you know this worked?

If you can view the XML file that's attached to the message sent by Exchange and the file contains audit log entries, then you've successfully configured, exported, and viewed a mailbox audit log.

Run a non-owner mailbox access report

Exchange Server 2013 > Messaging policy and compliance > Exchange auditing reports >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-25*

The Non-Owner Mailbox Access Report in the Exchange Administration Center (EAC) lists the mailboxes that have been accessed by someone other than the person who owns the mailbox. When a mailbox is accessed by a non-owner, Microsoft Exchange logs information about this action in a mailbox audit log that's stored as an email message in a hidden folder in the mailbox being audited. Entries from this log are displayed as search results and include a list of mailboxes accessed by a non-owner, who accessed the mailbox and when, the actions performed by the non-owner, and whether the action was successful. By default, entries in the mailbox audit log are retained for 90 days.

When you enable mailbox audit logging for a mailbox, Microsoft Exchange logs specific actions by non-owners, including both administrators and users, called *delegated users*, who have been assigned permissions to a mailbox. You can also narrow the search to users inside or outside your organization.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox audit logging" entry in the Messaging policy

and compliance permissions topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Enable mailbox audit logging

You have to enable mailbox audit logging for each mailbox that you want to run a non-owner mailbox access report for. If mailbox audit logging isn't enabled, you won't get any results when you run a report.

To enable mailbox audit logging for a single mailbox, run the following Shell command.

```
Set-Mailbox <Identity> -AuditEnabled $true
```

For example, to enable mailbox auditing for a user named Florence Flipo, run the following command.

```
Set-Mailbox "Florence Flipo" -AuditEnabled $true
```

To enable mailbox auditing for all user mailboxes in your organization, run the following commands.

```
$UserMailboxes = Get-mailbox -Filter {(RecipientTypeDetails
-eq 'UserMailbox')}

$UserMailboxes | ForEach {Set-Mailbox $_.Identity -
AuditEnabled $true}
```

How do you know this worked?

Run the following command to verify that you've successfully configured mailbox audit logging.

```
Get-Mailbox | FL Name,AuditEnabled
```

A value of `True` for the *AuditEnabled* property verifies that audit logging is enabled.

Run a non-owner mailbox access report

1. In the EAC, navigate to **Compliance Management** > **Auditing**.

2. Click **Run a non-owner mailbox access report**.

By default, Microsoft Exchange runs the report for non-owner access to any mailboxes in the organization over the past two weeks. The mailboxes listed in the search results have been enabled for mailbox audit logging.

3. To view non-owner access for a specific mailbox, select the mailbox from the list of mailboxes. View the search results in the details pane.

Tip:
Want to narrow the search results? Select the start date, end date, or both, and select specific mailboxes to search. Click **Search** to re-run the report.

Search for specific types of non-owner access

You can also specify the type of non-owner access, also called the logon type, to search for. Here are your options:

- **All non-owners** Search for access by administrators and delegated users inside your organization. Also includes access user outside of your organization.
- **External users** Search for access by users outside of your organization.
- **Administrators and delegated users** Search for access by administrators and delegated users inside your organization.
- **Administrators** Search for access by administrators in your organization.

What gets logged in the mailbox audit log?

When you run a non-owner mailbox access report, entries from the mailbox audit log are displayed in the search results in the EAC. Each report entry contains this information:

- Who accessed the mailbox and when
- The actions performed by the non-owner
- The affected message and its folder location
- Whether the action was successful

The following table describes the types of actions logged, and whether these actions are logged by default for access by administrators and for access by delegated users. If you want to track actions that aren't logged by default, you have to use the Shell to enable logging of those actions.

Action	Description	Administrators	Delegated users
Update	A message was changed.	Yes	Yes
Copy	A message was copied to another folder.	No	No
Move	A message was moved to another folder.	Yes	No

Move To Deleted Items	A message was moved to the Deleted Items folder.	Yes	No
Soft-delete	A message was deleted from the Deleted Items folder.	Yes	Yes
Hard-delete	A message was purged from the Recoverable Items folder.	Yes	Yes
FolderBind	A mailbox folder was accessed.	Yes	No
Send as	A message was sent using SendAs permission. This means another user sent the message as though it came from the mailbox owner.	Yes	Yes
Send on behalf of	A message was sent using SendOnBehalf permission. This means another user sent the message on behalf of the mailbox owner. The message will indicate to the recipient who the message was sent on behalf of and who actually sent the message.	Yes	No

MessageBind	A message was viewed in the preview pane or opened.	No	No

How do you know this worked?

To verify that you've successfully run a non-owner mailbox access report, check the search results pane. Mailboxes that you ran the report for are displayed in this pane. If there are no results for a specific mailbox, it's possible there hasn't been access by a non-owner or that non-owner access hasn't taken place within the specified date range. As previously described, be sure to verify that audit logging has been enabled for the mailboxes you want to search for access by non-owners.

Run a per-mailbox litigation hold report

Exchange Server 2013 > Messaging policy and compliance > Exchange auditing reports >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-13*

If your organization is involved in a legal action, you may have to take steps to preserve relevant data, such as email messages, that may be used as evidence. In situations like this, you can use litigation hold to retain all email sent and received by specific people or retain all email sent and received in your organization for a specific time period. For more information about what happens when a mailbox is on litigation hold and how to enable and disable it, see the "Mailbox Features" section in Manage user mailboxes.

Use the litigation hold report to keep track of the following types of changes made to a mailbox in a given time period:

- Litigation hold was enabled.
- Litigation hold was disabled.

For each of these change types, the report includes the user who made the change and the time and date the change was made.

What do you need to know before you begin?

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "View-only administrator audit logging" entry in the Exchange and Shell infrastructure permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the EAC to run a litigation hold report

1. In the EAC, navigate to **Compliance Management** > **Auditing**.
2. Click **Run a per-mailbox litigation hold report**.

Microsoft Exchange runs the report for litigation hold changes made to any mailbox in the past two weeks.

3. To view the changes for a specific mailbox, in the search results pane, select the mailbox. View the search results in the details pane.

Tip:

Want to narrow the search results? Select the start date, end date, or both, and select specific mailboxes to search. Click **Search** to re-run the report.

How do you know this worked?

To verify that you've successfully run a litigation hold report, mailboxes that had litigation hold changes within the date range are displayed in the search results pane. If there are no results, then no changes to litigation hold have taken place within the date range or recent changes haven't taken effect yet.

Note:

When a mailbox is put on litigation hold, it can take up to 60 minutes for the hold to take effect.

Search the role group changes or administrator audit logs

Exchange Server 2013 > Messaging policy and compliance > Exchange auditing reports >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-12-02*

You can search the administrator audit logs to discover who made changes to organization, server, and recipient configuration. This can be helpful when you're trying to track the cause of unexpected behavior, to identify a malicious administrator, or to verify that compliance requirements are being met. For more information about administrator audit logging, see Administrator audit logging.

If you want to search the mailbox audit log, see Mailbox audit logging.

Tip:
In Exchange Online, you can use the EAC to view entries in the administrator audit log. For more information, see View the administrator audit log.

What do you need to know before you begin?

- Estimated time to complete each procedure: less than 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "View-only administrator audit logging" entry in the Exchange and Shell infrastructure permissions topic.
- Administrator audit logging is enabled by default. To verify that it's enabled, run the following command:

```
Get-AdminAuditLogConfig | FL AdminAuditLogEnabled
```

A value of `True` indicates that administrator audit logging is enabled. A value of `False` indicates that it's disabled. If you need to enable administrator audit logging for an on-premises Exchange organization, run the following command:

```
Set-AdminAuditLogConfig -AdminAuditLogEnabled $true
```

Note:
The **Set-AdminAuditLogConfig** cmdlet isn't available in Exchange Online.

For more information, see Manage administrator audit logging.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to run the management role group changes report

If you want to know what changes to management role group membership have been made to role groups in your organization, you can use the Administrator Role Group report in the Exchange Administration Center (EAC). Using the Administrator Role Group report, you can view a list of role groups that have changed during a specified date range. You can also select the specific role groups you want to view changes for.

1. In the EAC, select **Compliance management** > **Auditing**, and then click **Run an administrator**

role group report.

2. Select a date range using the **Start date** and **End date** fields.
3. Click **Select role groups**, and then select the role groups you want to show changes for or leave this field blank to search for changes in all role groups.
4. Click **Search**.

If any changes are found using the criteria you specified, a list of changes will be displayed in the results pane. Clicking a role group displays the changes to the role group in the details pane.

Use the EAC to export the administrator audit log

If you want to create an XML file that contains changes made to your organization, you can use the Export Administrator Audit Log report in the EAC. Using the Export Administrator Audit Log report, you can specify a date range to search for audit log entries that contain changes made by users you specify. The XML file is then sent to a recipient as an email attachment. The maximum size of the XML file is 10 megabytes (MB).

Note:
Outlook Web App doesn't allow you to open XML attachments by default. You can either configure Exchange to allow XML attachments to be viewed using Outlook Web App, or you can use another email client, such as Microsoft Outlook, to view the attachment. For information about how to configure Outlook Web App to allow you to view an XML attachment, see View or configure Outlook Web App virtual directories.

1. In the EAC, select **Compliance management** > **Auditing**, and then click **Export the administrator audit log**.
2. Select a date range using the **Start date** and **End date** fields.
3. In the **Send the auditing report to** field, click **Select users** and then select the recipient you want to send the report to.
4. Click **Export**.

If any log entries are found using the criteria you specified, an XML file will be created and sent as an email attachment to the recipient you specified.

Use the Shell to search for audit log entries

You can use the Shell to search for audit log entries that meet the criteria you specify. For a list of search criteria, see Administrator audit logging. This procedure uses the **Search-AdminAuditLog** cmdlet and displays search results in the Shell. You can use this cmdlet when you need to return a set of results that exceeds the limits defined on the **New-AdminAuditLogSearch** cmdlet or in the EAC Audit Reporting reports.

If you want to send audit log search results in an email attachment to a recipient, see Use the Shell to search for audit log entries and send results to a recipient later in this topic.

To search the audit log for criteria you specify, use the following syntax.

```
Search-AdminAuditLog - Cmdlets <cmdlet 1, cmdlet 2, ...> -
Parameters <parameter 1, parameter 2, ...> -StartDate
<start date> -EndDate <end date> -UserIds <user IDs> -
ObjectIds <object IDs> -IsSuccess <$True | $False >
```

Note:

The **Search-AdminAuditLog** cmdlet returns a maximum of 1,000 log entries by default. Use the *ResultSize* parameter to specify up to 250,000 log entries. Or, use the value `Unlimited` to return all entries.

This example performs a search for all audit log entries with the following criteria:

- **Start date** 08/04/2012
- **End date** 10/03/2012
- **User IDs** davids, chrisd, kima
- **Cmdlets** **Set-Mailbox**
- **Parameters** *ProhibitSendQuota*, *ProhibitSendReceiveQuota*, *IssueWarningQuota*, *MaxSendSize*, *MaxReceiveSize*

```
Search-AdminAuditLog -Cmdlets Set-Mailbox -Parameters
ProhibitSendQuota, ProhibitSendReceiveQuota,
IssueWarningQuota, MaxSendSize, MaxReceiveSize -StartDate
08/04/2012 -EndDate 10/03/2012 -UserIds davids, chrisd,
kima
```

This example searches for changes made to a specific mailbox. This is useful if you're troubleshooting or you need to provide information for an investigation. The following criteria are used:

- **Start date** 05/01/2012
- **End date** 10/03/2012
- **Object ID** contoso.com/Users/DavidS

```
Search-AdminAuditLog -StartDate 05/01/2012 -EndDate
10/03/2012 -ObjectID contoso.com/Users/DavidS
```

If your searches return many log entries, we recommend that you use the procedure provided in Use the Shell to search for audit log entries and send results to a recipient later in this topic. The procedure in that section sends an XML file as an email attachment to the recipients you specify, enabling you to more easily extract the data you're interested in.

For detailed syntax and parameter information, see Search-AdminAuditLog.

View details of audit log entries

The **Search-AdminAuditLog** cmdlet returns the fields described in the "Audit log contents section of Administrator audit logging. Of the fields returned by the cmdlet, two fields, **CmdletParameters**

and **ModifiedProperties**, contain additional information that isn't viewable by default.

To view the contents of the **CmdletParameters** and **ModifiedProperties** fields, use the following steps. Or, you can use the procedure in Use the Shell to search for audit log entries and send results to a recipient later in this topic to create an XML file.

This procedure uses the following concepts:

- Arrays
- User-defined variables

1. Decide the criteria you want to search for, run the **Search-AdminAuditLog** cmdlet, and store the results in a variable using the following command.

```
$Results = Search-AdminAuditLog <search criteria>
```

2. Each audit log entry is stored as an array element in the variable `$Results`. You can select an array element by specifying its array element index. Array element indexes start at zero (0) for the first array element. For example, to retrieve the 5th array element, which has an index of 4, use the following command.

```
$Results[4]
```

3. The previous command returns the log entry stored in array element 4. To see the contents of the **CmdletParameters** and **ModifiedProperties** fields for this log entry, use the following commands.

```
$Results[4].CmdletParameters
$Results[4].ModifiedProperties
```

4. To view the contents of the **CmdletParameters** or **ModifiedParameters** fields in another log entry, change the array element index.

Use the Shell to search for audit log entries and send results to a recipient

You can use the Shell to search for audit log entries that meet the criteria you specify, and then send those results to a recipient you specify as an XML file attachment. The results are sent to the recipient within 15 minutes. For a list of search criteria, see Administrator audit logging.

Note:
Outlook Web App doesn't allow you to open XML attachments by default. You can either configure Exchange to allow XML attachments to be viewed using Outlook Web App, or you can use another email client, such as Microsoft Outlook, to view the attachment. For information about how to configure Outlook Web App to allow you to view an XML attachment, see View or configure Outlook Web App virtual directories.

To search the audit log for criteria you specify, use the following syntax.

```
New-AdminAuditLogSearch -Cmdlets <cmdlet 1, cmdlet 2, ...>
-Parameters <parameter 1, parameter 2, ...> -StartDate
<start date> -EndDate <end date> -UserIds <user IDs> -
ObjectIds <object IDs> -IsSuccess <$True | $False > -
StatusMailRecipients <recipient 1, recipient 2, ...> -Name
<string to include in subject>
```

This example performs a search for all audit log entries with the following criteria:

- **Start date** 08/04/2012
- **End date** 10/03/2012
- **User IDs** davids, chrisd, kima
- **Cmdlets** **Set-Mailbox**
- **Parameters** *ProhibitSendQuota*, *ProhibitSendReceiveQuota*, *IssueWarningQuota*, *MaxSendSize*, *MaxReceiveSize*

The command sends the results to the davids@contoso.com SMTP address with "Mailbox limit changes" included in the subject line of the message.

```
New-AdminAuditLogSearch -Cmdlets Set-Mailbox -Parameters
ProhibitSendQuota, ProhibitSendReceiveQuota,
IssueWarningQuota, MaxSendSize, MaxReceiveSize -StartDate
08/04/2012 -EndDate 10/03/2012 -UserIds davids, chrisd,
kima -StatusMailRecipients davids@contoso.com -Name
"Mailbox limit changes"
```

Note:

The report that the **New-AdminAuditLogSearch** cmdlet generates can be a maximum of 10 MB in size. If the search you perform returns a report larger than 10 MB, change the search criteria you specified. For example, reduce the size of the date range and run multiple reports, each with a portion of the original date range.

For more information about the format of the XML file, see Administrator audit log structure.

For detailed syntax and parameter information, see New-AdminAuditLogSearch.

View the administrator audit log

Exchange Server 2013 > Messaging policy and compliance > Exchange auditing reports >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-08-07*

In Microsoft Exchange Online Protection (EOP), Microsoft Exchange Online, and Microsoft Exchange 2013, you can use the Exchange admin center (EAC) to search for and view entries in the *administrator audit log*. The administrator audit log records specific actions, based on Exchange Management Shell cmdlet, performed by administrators and users who have been assigned administrative privileges. Entries in the administrator audit log provide you with information about what cmdlet was run, which parameters were used, who ran the cmdlet, and what objects were affected.

Note:

- The administrator audit log doesn't record any action that is based on an Exchange Management Shell cmdlet that begins with the verbs **Get**, **Search**, or **Test**.
- Audit log entries are kept for 90 days. When an entry is older than 90 days, it's deleted.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "View reports" entry in the **Feature permissions in EOP** topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to view the administrator audit log

1. In the EAC, navigate to **Compliance management** > **Auditing**, and choose **View the administrator audit log**.
2. Choose a **Start date** and **End date**, and then choose **Search**. All configuration changes made during the specified time period are displayed, and can be sorted, using the following information:
   - **Date** The date and time that the configuration change was made. The date and time are stored in Coordinated Universal Time (UTC) format.
   - **Cmdlet** The name of the cmdlet that was used to make the configuration change.
   - **User** The name of the user account of the user who made the configuration change.

Up to 5000 entries will be displayed on multiple pages. Specify a smaller date range if you need to narrow your results. If you select an individual search result, the following information is displayed in the details pane:

- **Object modified** The object that was modified by the cmdlet.
- **Parameters (Parameter:Value)** The cmdlet parameters that were used, and any value specified with the parameter.

3. If you want to print a specific audit log entry, choose the **Print** button in the details pane.

How do you know this worked?

If you've successfully run an administrator audit log report, configuration changes made within the date range you specify are displayed in the search results pane. If there are no results, change the date range and then run the report again.

Note:

When a change is made in your organization, it may take up to 15 minutes to appear in audit log search results. If a change doesn't appear in the administrator audit log, wait a few minutes and run the search again.

Anti-spam and anti-malware protection

Exchange Server 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-13*

Microsoft Exchange Server 2013 customers are automatically provided with anti-spam and anti-malware protection. The following topics (and their associated subtopics) provide overview information and configuration steps for customizing the built-in spam filtering and malware filtering settings so that they best meet the needs of your organization:

Anti-spam protection Describes the basic built-in anti-spam protection features as well as other anti-spam protection options such as using a cloud-hosted anti-spam solution and managing quarantined messages.

Anti-malware protection Describes the basic built-in anti-malware protection features as well as other anti-malware protection options. Among the information included are an Anti-Malware FAQ and details about how to configure anti-malware settings using the Exchange Administration Center or the Exchange Management Shell.

You can also use anti-malware programs in the Windows operating system on Exchange servers to help further enhance the security and health of your Exchange organization. However, there are important considerations for how to best implement file-level scanning with Exchange 2013. For more information, see Anti-Virus Software in the Operating System on Exchange Servers.

Tip:

You can also create transport rules to enforce company specific regulations and policies; for more information see Transport rules. Exchange 2013 customers who have purchased the data loss prevention (DLP) feature can also create DLP policies to help protect sensitive data; for more information, see Data loss prevention.

Anti-spam protection

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-23*

Spammers, or malicious senders, use a variety of techniques to send spam into your organization. No single tool or process can eliminate all spam. However, Microsoft Exchange Server 2013 provides a layered, multipronged, and multifaceted approach to reducing spam. Exchange uses transport agents to provide anti-spam filtering, and the built-in anti-spam agents that are available in Exchange 2013 are relatively unchanged from Microsoft Exchange Server 2010.

For more anti-spam features and easier management, you can elect to purchase the Forefront Online Protection for Exchange (FOPE) hosted email filtering service or the next version of this service, Microsoft Exchange Online Protection (EOP). For a comparison of EOP and Exchange 2013 features, see Benefits of anti-spam features in Exchange Online Protection over Exchange Server 2013.

For information about the built-in anti-malware capabilities in Exchange 2013, see Anti-malware protection.

Contents

Anti-spam agents on Mailbox servers

Typically, you would enable the anti-spam agents on a mailbox server if your organization doesn't have an Edge Transport server, or doesn't do any prior anti-spam filtering before accepting incoming messages. For more information, see Enable anti-spam functionality on Mailbox servers.

Like all transport agents, each anti-spam agent is assigned a priority value. A lower value indicates a higher priority, so typically, an anti-spam agent with priority 1 will act on a message before an anti-spam agent with priority 9. However, the SMTP event where the anti-spam agent is registered is also very important in determining the order that anti-spam agents act on messages. A low priority anti-spam agent that's registered on an SMTP event early in the transport pipeline will act on a message before a high priority anti-spam agent that's registered on an SMTP event later in the transport pipeline.

Based on the default priority value of the anti-spam agent, and the SMTP event in the transport pipeline where the anti-spam agent is registered, the following list describes the agents and the default order in which they are applied to messages on a Mailbox server:

1. **Sender Filter agent** Sender filtering compares the sender on the MAIL FROM: SMTP command to an administrator-defined list of senders or sender domains who are prohibited from sending messages to the organization to determine what action, if any, to take on an inbound message. For more information, see Sender filtering.
2. **Sender ID agent** Sender ID relies on the IP address of the sending server and the Purported Responsible Address (PRA) of the sender to determine whether the sender is spoofed or not. For more information, see Sender ID.
3. **Content Filter agent** Content filtering assesses the contents of a message. For more information, see Content filtering.

Spam quarantine is a feature of the Content Filter agent that reduces the risk of losing legitimate messages that are incorrectly classified as spam. Spam quarantine provides a temporary storage location for messages that are identified as spam and that shouldn't be delivered to a user mailbox inside the organization. For more information, see Spam quarantine.

Content filtering also acts on the safelist aggregation feature. Safelist aggregation collects data from the anti-spam safe lists that Microsoft Outlook and Outlook Web App users configure and makes this data available to the Content Filter agent. For more information, see Safelist Aggregation.

4. **Protocol Analysis agent** The Protocol Analysis agent is the underlying agent that implements the sender reputation functionality. Sender reputation relies on persisted data about the IP address of the sending server to determine what action, if any, to take on an inbound message. A sender reputation level (SRL) is calculated from several sender characteristics that are derived from message analysis and external tests. For more information, see Sender reputation and the Protocol Analysis agent.

Return to top

Anti-spam agents on Edge Transport servers

If your organization has an Edge Transport server installed in the perimeter network, all of the anti-spam agents that are available on a Mailbox server are installed and enabled by default on the Edge Transport server. However, the following anti-spam agents are only available on an Edge Transport server:

- **Connection Filtering agent** Connection filtering inspects the IP address of the remote server that's trying to send messages to determine what action, if any, to take on an inbound message. Connection filtering uses an IP Block list, IP Allow list, IP Block List provider services and IP Allow List provider services to determine whether the connection IP should be blocked or allowed. For more information, see Connection Filtering on Edge Transport Servers.
- **Recipient Filter agent** Recipient filtering compares the message recipients on the RCPT TO: SMTP command to an administrator-defined Recipient Block list. If a match is found, the message

isn't permitted to enter the organization. The recipient filter also compares recipients on inbound messages to the local recipient directory to determine whether the message is addressed to valid recipients. When a message isn't addressed to valid recipients, the message is rejected. For more information, see Recipient filtering on Edge Transport servers.

Note:

Although the Recipient Filter agent is available on Mailbox servers, you shouldn't configure it. When recipient filtering on a Mailbox server detects one invalid or blocked recipient in a message that contains other valid recipients, the message is rejected. If you install the anti-spam agents on a Mailbox server, the Recipient Filter agent is enabled by default. However, it isn't configured to block any recipients.

- **Attachment Filtering agent** Attachment filtering blocks messages based on attachment file name, file name extension, or file MIME content type. You can configure attachment filtering to block a message and its attachment, to strip the attachment and allow the message to pass through, or to silently delete the message and its attachment. For more information, see Attachment filtering on Edge Transport servers.

Based on the default priority value of the anti-spam agent, and the SMTP event in the transport pipeline where the anti-spam agent is registered, this is the default order in which the anti-spam agents are applied on an Edge Transport server:

1. Connection Filtering agent
2. Sender Filter agent
3. Recipient Filter agent
4. Sender ID agent
5. Content Filter agent
6. Protocol Analysis agent for sender reputation
7. Attachment Filtering agent

Return to top

Anti-spam stamps

Anti-spam stamps help you diagnose spam-related problems by applying diagnostic metadata, or stamps, such as sender-specific information, puzzle validation results, and content filtering results, to messages as they pass through the anti-spam features that filter inbound messages from the Internet. For more information, see Anti-spam stamps.

Return to top

Strategy for anti-spam approach

Your strategy for how to configure the anti-spam features and establish the aggressiveness of your anti-spam agent settings requires that you plan and calculate carefully. If you set all anti-spam filters to their most aggressive levels and configure all anti-spam features to reject all suspicious messages, you're more likely to reject messages that aren't spam. On the other hand, if you don't set

the anti-spam filters at a sufficiently aggressive level and don't set the spam confidence level (SCL) threshold low enough, you probably won't see a reduction in the spam that enters your organization.

It's a best practice to reject a message when Exchange detects a bad message through the Connection Filtering agent, Recipient Filter agent, or Sender Filter agent. This approach is better than quarantining such messages or assigning metadata, such as anti-spam stamps, to such messages. The Connection Filtering agent and Recipient Filter agent automatically block messages that are identified by the respective filters. The Sender Filter agent is configurable.

This best practice is recommended because the SCL that underlies connection filtering, recipient filtering, or sender filtering is relatively high. For example, with sender filtering, where the administrator has configured specific senders to block, there's no reason to assign the sender filtering data to such messages and to continue to process them. In most organizations, blocked messages should be rejected. (If you didn't want the messages rejected, you wouldn't have put them on the Blocked Senders List.)

The same logic applies to real-time block list services and recipient filtering, although the underlying confidence isn't as high as the IP Block list. You should be aware that the further along the mail flow path a message travels, the greater the probability of false positives, because the anti-spam features are evaluating more variables. Therefore, you may find that if you configure the first several anti-spam features in the anti-spam chain more aggressively, you can reduce the bulk of your spam. As a result, you'll save processing, bandwidth, and disk resources so that you can process more ambiguous messages.

Ultimately, you must plan to monitor the overall effectiveness of the anti-spam features. If you monitor carefully, you can continue to adjust the anti-spam features to work well together for your environment. With this approach, you should plan on a fairly non-aggressive configuration of the anti-spam features when you start. This approach lets you minimize the number of false positives. As you monitor and adjust the anti-spam features, you can become more aggressive about the type of spam and spam attacks that your organization experiences.

Return to top

Benefits of anti-spam features in Exchange Online Protection over Exchange Server 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

Note:
Exchange Online Protection (EOP) is the next version of Forefront Online Protection for Exchange (FOPE). EOP anti-spam protection features are included in Exchange Online.

The following are benefits of using Exchange anti-spam protection in the cloud (Microsoft Exchange Online or Microsoft Exchange Online Protection) as opposed to Microsoft Exchange Server 2013, which has most of the same built-in anti-spam capabilities as Microsoft Exchange Server 2010:

- **More control and easier configuration** Administrators can use the Exchange Administration Center (EAC) web-based management console in order to customize spam filtering settings so that they best meet the needs of your organization. There is no anti-spam user interface in Exchange Server 2013.
- **Stronger connection filtering** In Exchange 2013, connection filtering IP Allow lists and IP Block lists are available only if you install an Edge Transport server in your perimeter network. For more information, see Edge Transport servers. In the cloud, you can choose to skip spam filtering on email messages sent from trusted senders (gathered from various third-party sources), ensuring that these messages are not mistakenly marked as spam. Also, the hosted filtering service uses Microsoft's own block lists and lists aggregated from vendors to provide greater IP-level filtering.
- **Stronger content filtering** You can easily configure your content filter policy to:
  - Filter messages written in specific languages.
  - Filter messages sent from specific countries or regions.
  - Mark bulk email messages (such as advertisements and marketing emails) as spam.
  - Search for attributes in a message and act upon the message if it matches a specific advanced spam option attribute. If you are concerned about phishing, some of these options offer a combination of Sender ID and SPF technologies to authenticate and verify that messages are not spoofed.

In addition to the above content filter options that you can configure in the EAC, the hosted filtering service uses additional URL lists to block suspicious messages that contain specific URLs within their message body.

- **Quicker updates** Spam updates are propagated more quickly across the network. In Exchange Server 2013 updates occur two times per month, whereas the service is updated multiple times per hour.
- **Outbound filtering** Outbound spam filtering is always enabled if you use the hosted service for sending outbound email, thereby protecting organizations using the service and their intended recipients.

Enable anti-spam functionality on Mailbox servers

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-23*

In Microsoft Exchange Server 2013, the following anti-spam agents are available in the Transport service on Mailbox servers, but they are not installed by default:

- Content Filter agent
- Sender ID agent
- Sender Filter agent
- Protocol Analysis agent for sender reputation

However, you can install these anti-spam agents on a Mailbox server using a script in the Exchange Management Shell. Typically, you would install the anti-spam agents on a Mailbox server only when your organization accepts all incoming mail without any prior anti-spam filtering.

Note:

Although the Recipient Filter agent is available on Mailbox servers, you shouldn't configure it. When recipient filtering on a Mailbox server detects one invalid or blocked recipient in a message that contains other valid recipients, the message is rejected. Although the Recipient Filter agent is enabled by default, it isn't configured to block any recipients. For more information, see Manage recipient filtering on Edge Transport servers.

What happens if you install the available anti-spam agents in the Transport service on a Mailbox server, but you also have other Exchange anti-spam agents operating on the messages before they reach the Mailbox server? For example, what if you have an Edge Transport server in the perimeter network? The anti-spam agents on the Mailbox server recognize the anti-spam X-header values that are added to messages by other Exchange anti-spam agents, and messages that contain these X-headers pass through without being scanned again. However, recipient look-ups performed by the Recipient Filter agent will occur again on the Mailbox server.

What do you need to know before you begin?

- Estimated time to complete this task: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport configuration" entry in the Mail flow permissions topic.
- The Connection Filter agent and the Attachment Filter agent aren't available on Mailbox servers. They're only available on an Edge Transport server. However, the Malware agent is installed and enabled by default on a Mailbox server. For more information, see Anti-malware protection.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Use the Shell to run the Install-AntispamAgents.ps1 script

Run the following command:

```
& $env:ExchangeInstallPath\Scripts\Install-AntiSpamAgents.ps1
```

How do you know this step worked?

You know this step worked if the script runs without errors, and asks you to restart the Microsoft Exchange Transport service.

Step 2: Use the Shell to restart the Microsoft Exchange Transport service

Run the following command:

```
Restart-Service MSExchangeTransport
```

How do you know this step worked?

You know this step worked if the Microsoft Exchange Transport service restarts without errors.

Step 3: Use the Shell to specify the internal SMTP servers in your organization

You need to specify the IP addresses of any internal SMTP servers that should be ignored by the Sender ID agent. In fact, you need to specify the IP address of at least one internal SMTP server. If the Mailbox server where you're running the anti-spam agents is the only SMTP server in your organization, specify the IP address of that computer.

To add the IP addresses of internal SMTP servers without affecting any existing values, run the following command:

```
Set-TransportConfig -InternalSMTPServers @{Add="<ip address1>","<ip address2>"...}
```

This example adds the internal SMTP server addresses 10.0.1.10 and 10.0.1.11 to the transport configuration of your organization.

```
Set-TransportConfig -InternalSMTPServers
@{Add="10.0.1.10","10.0.1.11"}
```

How do you know this step worked?

To verify that you have successfully specified the IP address of at least one internal SMTP server, do the following:
1. Run the following command:

```
Get-TransportConfig | Format-List InternalSMTPServers
```

2. Verify the IP address of at least one valid internal SMTP server is displayed.

Sender filtering

Exchange Server 2013 > Anti-spam and anti-malware protection > Anti-spam protection >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-12*

Sender filtering relies on the MAIL FROM: SMTP header to determine what action, if any, to take on an inbound email message. Sender filtering is provided by the Sender Filter agent.

The Sender Filter agent acts on messages from specific senders outside the organization. Administrators maintain a list of senders who are blocked from sending messages to the organization. As an administrator, you can block single senders (such as kim@contoso.com), whole domains (contoso.com), or domains and all subdomains (*.contoso.com). You can also configure what action the Sender Filter agent should take when a message that has a blocked sender is found. You can configure the following actions:

- The Sender Filter agent rejects the SMTP request with a `554 5.1.0 Sender Denied` SMTP session error and closes the connection.
- The Sender Filter agent accepts the message and updates the message to indicate that the message came from a blocked sender. Because the message came from a blocked sender and it's marked as such, the Content Filter agent will use this information when it calculates the spam confidence level (SCL).

You can designate blocked senders and define how the Sender Filter agent should act on messages from blocked senders. For more information about how to configure the Sender Filter agent, see Manage sender filtering.

Important:

The MAIL FROM: SMTP headers can be spoofed. Therefore, you shouldn't rely on the Sender Filter agent only. Use the Sender Filter agent and the Sender ID agent together. The Sender ID agent uses the originating IP address of the sending server to verify that the domain in the MAIL FROM: SMTP header matches the domain that's registered. For more information about the Sender ID agent, see Sender ID.

Using the Sender Filter agent to block messages

When the Sender Filter agent is enabled on an Exchange server, sender filtering blocks inbound messages that come from the Internet, but aren't authenticated. These messages are handled as external messages. You can disable the Sender Filter agent in individual computer configurations. For more information, see Manage sender filtering.

When you enable the Sender Filter agent on an Exchange server, the Sender Filter agent filters all messages that come through all Receive connectors on that computer. As noted earlier in this topic, only messages that come from external sources are filtered. *External sources* are defined as non-authenticated sources. These are considered anonymous Internet sources.

As a best practice, you shouldn't filter email messages from trusted partners or from inside your organization. When you run anti-spam filters, there's always a chance that the filters will detect false positives. You should configure anti-spam agents to run only on messages from potentially untrusted and unknown sources. This will reduce the chance that anti-spam filters will mishandle legitimate messages. You can enable and disable the Sender Filter agent to run on messages from any source. For more information, see Manage sender filtering.

You can configure the Sender Filter agent to block inbound messages that don't specify a sender and domain in the MAIL FROM: SMTP header. You can use this feature to prevent non-delivery report (NDR) attacks on the Exchange server. Most legitimate SMTP messages come from SMTP servers that provide a sender and domain in the MAIL FROM: SMTP command.

Specifying the block action

After you've specified blocked senders and domains, you must specify how you want the Sender Filter agent to act on messages from blocked senders and domains. We recommend that you reject the messages. When you use the Sender Filter agent to block email addresses and domains that are specified by an Exchange administrator, the chance of false positives is relatively less than when you use other anti-spam agents. For example, the Content Filter agent is an anti-spam agent that relies on many different variables to determine whether a message is spam.

There are only two scenarios in which legitimate messages may be rejected by the Sender Filter agent:

- If you mistype an email address or domain name, the wrong sender may be blocked.
- If a domain name is reregistered to a legitimate company after you add the domain to your Blocked Senders list, you will unintentionally block legitimate messages.

In either of these cases, it may still make sense to reject the messages.

Manage sender filtering

Anti-spam and anti-malware protection > Anti-spam protection > Sender filtering >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Sender filtering is provided by the Sender Filter agent. The Sender Filter agent relies on the **MAIL FROM:** SMTP header to determine what action, if any, to take on an inbound email message.

When sender filtering functionality is enabled on an Exchange server, sender filtering functionality filters all messages that come through all Receive connectors on that computer.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.
- You can only use the Shell to perform this procedure.
- By default, anti-spam features aren't enabled in the Transport service on a Mailbox server. Typically, you only enable the anti-spam features on a Mailbox server if your Exchange organization doesn't do any prior anti-spam filtering before accepting incoming messages. For more information, see Enable anti-spam functionality on Mailbox servers.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable sender filtering

To disable sender filtering, run the following command:

```
Set-SenderFilterConfig -Enabled $false
```

To enable sender filtering, run the following command:

```
Set-SenderFilterConfig -Enabled $true
```

Note:
When you disable sender filtering, the underlying Sender Filter agent is still enabled. To disable the Sender Filter agent, run the command: `Disable-TransportAgent "Sender Filter Agent"`.

How do you know this worked?

To verify that you have successfully enabled or disabled sender filtering, do the following:
1. Run the following command:

```
Get-SenderFilterConfig | Format-List Enabled
```

2. Verify the value displayed is the value you configured.

Use the Shell to configure blocked senders and domains

To replace the existing values, run the following command:

```
Set-SenderFilterConfig -BlockedSenders <sender1,sender2...>
-BlockedDomains <domain1,domain2...> -
BlockedDomainsAndSubdomains <domain1,domain2...>
```

This example configures the Sender Filter agent to block messages from kim@contoso.com and john@contoso.com, messages from the fabrikam.com domain, and messages from northwindtraders.com and all its subdomains.

```
Set-SenderFilterConfig -BlockedSenders
kim@contoso.com,john@contoso.com -BlockedDomains
fabrikam.com -BlockedDomainsAndSubdomains
northwindtraders.com
```

To add or remove entries without modifying any existing values, run the following command:

```
Set-SenderFilterConfig -BlockedSenders
@{Add="<sender1>","<sender2>"...;
Remove="<sender1>","<sender2>"...} -BlockedDomains
@{Add="<domain1>","<domain2>"...;
Remove="<domain1>","<domain2>"...} -
BlockedDomainsAndSubdomains
@{Add="<domain1>","<domain2>"...;
Remove="<domain1>","<domain2>"...}
```

This example configures the Sender Filter agent with the following information:

- Add chris@contoso.com and michelle@contoso.com to the list of existing senders who are blocked.
- Remove tailspintoys.com from the list of existing sender domains that are blocked.
- Add blueyonderairlines.com to the list of existing sender domains and subdomains that are blocked.

```
Set-SenderFilterConfig -BlockedSenders
@{Add="chris@contoso.com","michelle@contoso.com"} -
BlockedDomains @{Remove="tailspintoys.com"} -
BlockedDomainsAndSubdomains @{Add="blueyonderairlines.com"}
```

How do you know this worked?

To verify that you have successfully configured blocked senders, do the following:
1. Run the following command:

```
Get-SenderFilterConfig | Format-List
BlockedSenders,BlockedDomains,BlockedDomainsAndSubdomains
```

2. Verify the values displayed are the values you configured.

Use the Shell to enable or disable blocking messages with blank senders

To enable or disable blocking message with blank senders, run the following command:

```
Set-SenderFilterConfig -BlankSenderBlockingenabled <$true |
$false>
```

This example configures the Sender Filter agent to block messages that don't specify a sender in the MAIL FROM: SMTP command:

```
Set-SenderFilterConfig -BlankSenderBlockingEnabled $true
```

How do you know this worked?

To verify that you have successfully enabled or disabled blocking messages with blank senders, do the following:
1. Run the following command:

```
Get-SenderFilterConfig | Format-List
BlankSenderBlockingEnabled
```

2. Verify the value displayed is the value you configured.

Sender ID

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

The Sender ID agent is an anti-spam agent that's available in Microsoft Exchange Server 2013. The Sender ID agent relies on the RECEIVED SMTP header and a query to the sending system's DNS service to determine what action, if any, to take on an inbound message.

Sender ID is intended to combat the impersonation of a sender and a domain, a practice that's frequently called *spoofing*. A *spoofed mail* is an email message that has a sending address that was modified to appear as if it originates from a sender other than the actual sender of the message.

Spoofed mails typically contain a From: address that purports to be from a certain organization. In the past, it was relatively easy to spoof the From: address, in both the SMTP session, such as the MAIL FROM: header, and in the RFC 2822 message data, such as From: "Masato Kawai" masato@contoso.com, because the headers weren't validated.

Contents

Using Sender ID to combat spoofing

Sender ID makes spoofing more difficult. When you enable Sender ID, each message contains a Sender ID status in the metadata of the message. When an email message is received, the Exchange server queries the sender's DNS server to verify that the IP address from which the message was received is authorized to send messages for the domain that's specified in the message headers. The IP address of the authorized sending server is referred to as the purported responsible address (PRA).

Domain administrators publish sender policy framework (SPF) records on their DNS servers. SPF records identify authorized outbound email servers. If an SPF record is configured on the sender's DNS server, the Exchange server parses the SPF record and determines whether the IP address from which the message was received is authorized to send email on behalf of the domain that's specified in the message. For more information about what an SPF record contains and how to create an SPF record, see Sender ID.

The Exchange server updates the message metadata with the Sender ID status based on the SPF record. After the Exchange server updates the message metadata, message delivery proceeds as it

ordinarily would.

Sender ID status values

The Sender ID evaluation process generates a Sender ID status for the message. The Sender ID status is used to evaluate the spam confidence level (SCL) rating for the message. This status can be set to one of the following values:

- **Pass** Both the IP address and Purported Responsible Address (PRA) passed the Sender ID verification check.
- **Neutral** Published Sender ID data is explicitly inconclusive.
- **Soft fail** The IP address for the PRA may be in the not permitted set.
- **Fail** The IP Address is not permitted; no PRA is found in the incoming mail or the sending domain does not exist.
- **None** No published SPF data exists in the sender's DNS.
- **TempError** A temporary DNS failure occurred, such as an unavailable DNS server.
- **PermError** The DNS record is invalid, such as an error in the record format.

The Sender ID status is added to the message metadata and is later converted to a MAPI property. The junk email filter in Microsoft Outlook uses the MAPI property during the generation of the SCL value.

Outlook neither displays the Sender ID status nor necessarily flags a message as junk at certain Sender ID values. Outlook uses the Sender ID status value only during the calculation of the SCL value.

Besides the seven scenarios that generate the Sender ID statuses, the Sender ID evaluation process may reveal instances where the From: IP address is missing. If the From: IP address is missing, the Sender ID status can't be set. If the Sender ID status can't be set, Exchange continues to process the message without including a Sender ID status on the message. The message isn't discarded or rejected. In this scenario, Sender ID status isn't set, and an application event is logged.

For more information about how the Sender ID status is displayed in messages, see Anti-spam stamps.

Sender ID options for handling spoofed mail and unreachable DNS servers

You can also define how the Exchange server handles messages that are identified as spoofed mail and how the Exchange server handles messages when a DNS server can't be reached. The options for how the Exchange server handles spoofed mail and unreachable DNS servers include the following actions:

- **Stamp the status** This option is the default action. All inbound messages to your organization have the Sender ID status included in the metadata of the message.
- **Reject** This option rejects the message and sends an SMTP error response to the sending server.

The SMTP error response is a 5*xx* level protocol response with text that corresponds to the Sender ID status.

- **Delete** This option deletes the message without informing the sending system of the deletion. In fact, the Exchange server sends a fake OK SMTP command to the sending server and then deletes the message. Because the sending server assumes the message was sent, it doesn't retry sending the message in the same session.

For more information about how to configure the Sender ID agent, see Manage Sender ID.

Return to top

Updating your organization's Internet-facing DNS to support Sender ID

The effectiveness of Sender ID depends on specific DNS data. The more organizations that update their Internet-facing DNS servers by using an SPF record, the more effectively Sender ID identifies spoofed email messages.

To support the Sender ID infrastructure, you must update your Internet-facing DNS data by creating an SPF record and hosting the SPF record on your public DNS servers. For more information about how to create and deploy SPF records, see Sender ID.

Return to top

Specifying recipients and sender domains to exclude from Sender ID filtering

You may want to exclude specific recipients and sender domains from Sender ID filtering. To do this, you specify the recipients and sender domains using the **Set-SenderIdConfig** cmdlet in the Exchange Management Shell. For more information, see Set-SenderIdConfig.

Return to top

Manage Sender ID

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Sender ID functionality is provided by the Sender ID agent. Sender ID validates the origin of email

messages by verifying the IP address of the sender against the purported owner of the sender domain. Sender ID filtering is performed on inbound messages that come from the Internet but aren't authenticated. These messages are handled as external messages.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.
- You can only use the Shell to perform this procedure.
- By default, anti-spam features aren't enabled in the Transport service on a Mailbox server. Typically, you only enable the anti-spam features on a Mailbox server if your Exchange organization doesn't do any prior anti-spam filtering before accepting incoming messages. For more information, see Enable anti-spam functionality on Mailbox servers.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable Sender ID

To disable Sender ID, run the following command:

```
Set-SenderIDConfig -Enabled $false
```

To enable Sender ID, run the following command:

```
Set-SenderIDConfig -Enabled $true
```

Note:

When you disable Sender ID, the underlying Sender ID agent is still enabled. To disable the Sender ID agent, run the command: `Disable-TransportAgent "Sender ID Agent"`.

How do you know this worked?

To verify that you have successfully enabled or disabled Sender ID, do the following:
1. Run the following command:

```
Get-SenderIDConfig | Format-List Enabled
```

2. Verify the value displayed is the value you configured.

Use the Shell to configure the Sender ID action for spoofed messages

To configure the Sender ID action for spoofed messages, run the following command:

```
Set-SenderIDConfig -SpoofedDomainAction <StampStatus | Reject | Delete>
```

This example configures the Sender ID agent to reject any messages where the IP address of the sending server isn't listed as an authoritative SMTP sending server in the DNS Sender Policy Framework (SPF) record for the sending domain.

```
Set-SenderIDConfig -SpoofedDomainAction Reject
```

How do you know this worked?

To verify that you have successfully configured the Sender ID action for spoofed messages, do the following:
1. Run the following command:

```
Get-SenderIDConfig | Format-List SpoofedDomainAction
```

2. Verify the value displayed is the value you configured.

Use the Shell to configure the Sender ID action for transient errors

To configure the Sender ID action for transient errors, run the following command:

```
Set-SenderIDConfig -TempErrorAction <StampStatus | Reject | Delete>
```

This example configures the Sender ID agent to stamp the messages when the Sender ID status can't be determined due to a temporary DNS server error. The message will be processed by other anti-spam agents and the Content Filter agent will use the mark when determining the SCL value for the message.

```
Set-SenderIDConfig -TempErrorAction StampStatus
```

Note that `stampstatus` is the default value for the *TempErrorAction* parameter.

How do you know this worked?

To verify that you have successfully configured the Sender ID action for transient errors, do the following:

1. Run the following command:

```
Get-SenderIDConfig | Format-List TempErrorAction
```

2. Verify the value displayed is the value you configured.

Use the Shell to configure recipient and sender domain exceptions

To replace the existing values, run the following command:

```
Set-SenderIDConfig -BypassedRecipients
<recipient1,recipient2...> -BypassedSenderDomains
<domain1,domain2...>
```

This example configures the Sender ID agent to bypass the Sender ID check for messages sent to kim@contoso.com and john@contoso.com, and to bypass the Sender ID check for messages sent from the fabrikam.com domain.

```
Set-SenderIDConfig -BypassedRecipients
kim@contoso.com,john@contoso.com -BypassedSenderDomains
fabrikam.com
```

To add or remove entries without modifying any existing values, run the following command:

```
Set-SenderIDConfig -BypassedRecipients
@{Add="<recipient1>","<recipient2>"...;
Remove="<recipient1>","<recipient2>"...} -
BypassedSenderDomains @{Add="<domain1>","<domain2>"...;
Remove="<domain1>","<domain2>"...}
```

This example configures the Sender ID agent with the following information:

- Add chris@contoso.com and michelle@contoso.com to the list of existing recipients who bypass the Sender ID check.
- Remove tailspintoys.com from the list of existing domains that bypass the Sender ID check.

```
Set-SenderIDConfig -BypassedRecipients
@{Add="chris@contoso.com","michelle@contoso.com"} -
BypassedSenderDomains @{Remove="tailspintoys.com"}
```

How do you know this worked?

To verify that you have successfully configured recipient and sender domain exceptions, do the following:

1. Run the following command:

```
Get-SenderIDConfig | Format-List
BypassedRecipients,BypassedSenderDomains
```

2. Verify the values displayed are the values you configured.

Content filtering

Exchange Server 2013 > Anti-spam and anti-malware protection > Anti-spam protection >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

The Content Filter agent evaluates inbound email messages and assesses the probability that an inbound message is legitimate or spam. Unlike many other filtering technologies, the Content Filter agent uses characteristics from a statistically significant sample of email messages. The inclusion of legitimate messages in this sample reduces the chance of mistakes. Because the Content Filter agent recognizes characteristics of legitimate messages and spam, its accuracy is increased. Updates to the Content Filter agent are available periodically through Microsoft Update.

Contents

Using the Content Filter agent

The Content Filter agent is one of several anti-spam agents in Exchange. When you configure anti-spam agents on an Exchange server, the agents act on messages cumulatively to reduce the amount of spam that enters the organization. For more information about how to plan and deploy anti-spam agents, see Anti-spam protection.

The Content Filter agent assigns a spam confidence level (SCL) rating to each message. The SCL rating is a number between 0 and 9. A higher SCL rating indicates that a message is more likely to be spam.

You can configure the Content Filter agent to take the following actions on messages according to their SCL rating:

- Delete message
- Reject message
- Quarantine message

For example, you may determine that messages that have an SCL rating of 7 or higher must be deleted, messages that have an SCL rating of 6 must be rejected, and messages that have an SCL rating of 5 must be quarantined.

You can adjust the SCL threshold behavior by assigning different SCL ratings to each of these actions. For more information about how to adjust the SCL threshold to suit your organization's requirements and about per-recipient SCL thresholds, see Spam Confidence Level Threshold.

Note:

Messages that are over 11 MB aren't scanned by the Intelligent Message Filter. Instead, they pass through the Content Filter without being scanned.

Allow phrases and Block phrases

You can customize how the Content Filter agent assigns SCL values by configuring custom words. Custom words are individual words or phrases that the Content Filter agent uses to apply appropriate filter processing. You configure approved words or phrases with Allow phrases and unapproved words or phrases with Block phrases. When the Content Filter agent detects a preconfigured Allow phrase in an inbound message, the Content Filter agent automatically assigns an SCL value of 0 to the message. Alternatively, when the Content Filter agent detects a configured Block phrase in an inbound message, the Content Filter agent assigns an SCL rating of 9.

You can enter custom words or phrases in any combination of uppercase and lowercase letters. However, when the Content Filter agent evaluates message content, it ignores case. The maximum number of custom words or phrases that can be created is 800.

Outlook Email Postmark validation

The Content Filter agent also includes Microsoft Office Outlook Email Postmark validation, a computational proof that Outlook applies to outgoing messages to help recipient messaging systems distinguish legitimate email from junk email. This feature helps reduce the chance of false positives. In the context of spam filtering, a *false positive* exists when a spam filter incorrectly identifies a message from a legitimate sender as spam. When Outlook Email Postmark validation is enabled, the Content Filter agent parses the inbound message for a computational postmark header. The presence of a valid, solved computational postmark header in the message indicates that the client computer that generated the message solved the computational postmark.

Computers don't require significant processing time to solve individual computational postmarks. However, processing postmarks for many messages may be prohibitive to a malicious sender. Anyone who sends millions of spam messages is unlikely to invest the processing power that is required to solve computational postmarks for all outbound spam. If a sender's email contains a valid, solved computational postmark, it's unlikely that the sender is a malicious sender. In this case, the Content Filter agent would lower the SCL rating. If the postmark validation feature is enabled and an inbound message either doesn't contain a computational postmark header or the

computational postmark header isn't valid, the Content Filter agent would not change the SCL rating.

Bypassing the recipient, sender, and sender domain

In some organizations, all email to certain aliases must be accepted. This scenario can introduce problems if your organization is in an industry that manages significant volumes of spam.

For example, a company named Woodgrove Bank has an alias named customerloans@woodgrovebank.com that provides email-based support to external loan customers. The Exchange administrators configure the Content Filter agent to set Block phrases that filter out words or phrases that are typically used in spam that is sent by unscrupulous loan agencies. To prevent potentially legitimate messages from being rejected, the administrators set exceptions to content filtering by entering a list of SMTP email recipient addresses in the Content Filter agent configuration.

You can also specify senders and sender domains that you do not want the Content Filter agent to block.

Safelist aggregation

In Exchange 2013, the Content Filter agent uses the Outlook Safe Senders Lists, Blocked Sender List, Safe Recipients Lists, and trusted contacts from Outlook to optimize spam filtering. *Safelist aggregation* is a set of anti-spam functionality that is shared across Outlook and Exchange. As its name suggests, this functionality collects data from the anti-spam safe lists that Outlook users configure and makes this data available to the anti-spam agents on the Exchange server. Email messages that Outlook users receive from contacts that those users have added to their Outlook Safe Recipients List, Safe Senders List, or trusted contacts list are identified by the Content Filter agent as safe. The Sender Filter agent also performs per-recipient sender filtering using the Blocked Senders list that users configure. For more information, see Safelist Aggregation.

Configuring the Content Filter agent

You configure the Content Filter agent by using the Exchange Management Shell.

Important:
Configuration changes that you make to the Content Filter agent in the Exchange Management Shell are only made on the local computer. If you have the Content Filter agent running on multiple Exchange servers in your organization, you must make Content Filter configuration changes to each computer.

The Content Filter agent depends on updates to determine whether a message can be delivered with confidence that it isn't spam. These updates contain data about phishing Web sites, Microsoft SmartScreen spam heuristics, and other Intelligent Message Filter updates. Content filter updates

generally contain about 6 MB of data that's useful for longer periods of time than other anti-spam update data.

Content filter updates are available from Microsoft Update. The content filter update data is updated and available for download every two weeks.

For more information about how to configure content filtering, see Manage content filtering.

Manage content filtering

Anti-spam and anti-malware protection > Anti-spam protection > Content filtering >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-04*

Content filtering is provided by the Content Filter agent. The Content Filter agent filters all messages that come through all Receive connectors on the Exchange server. Only messages that come from non-authenticated sources are filtered.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-spam feature" entry in the Anti-spam and anti-malware permissions topic.
- You can only use the Shell to perform this procedure.
- By default, anti-spam features aren't enabled in the Transport service on a Mailbox server. Typically, you only enable the anti-spam features on a Mailbox server if your Exchange organization doesn't do any prior anti-spam filtering before accepting incoming messages. For more information, see Enable anti-spam functionality on Mailbox servers.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable content filtering

To disable content filtering, run the following command:

```
Set-ContentFilterConfig -Enabled $false
```

To enable content filtering, run the following command:

```
Set-ContentFilterConfig -Enabled $true
```

Note:
When you disable content filtering, the underlying Content Filter agent is still enabled. To disable the Content Filter agent, run the command: `Disable-TransportAgent "Content Filter Agent"`.

How do you know this worked?

To verify that you have successfully enabled or disabled content filtering, do the following:
1. Run the following command:

```
Get-ContentFilterConfig | Format-List Enabled
```

2. Verify the value of the *Enabled* property that's displayed.

Use the Shell to enable or disable content filtering for external messages

By default, content filtering functionality is enabled for external messages.

To disable content filtering for external messages, run the following command:

```
Set-ContentFilterConfig -ExternalMailEnabled $false
```

To enable content filtering for external messages, run the following command:

```
Set-ContentFilterConfig -ExternalMailEnabled $true
```

How do you know this worked?

To verify that you have successfully enabled or disabled content filtering for external messages, do the following:
1. Run the following command:

```
Get-ContentFilterConfig | Format-List ExternalMailEnabled
```

2. Verify the value of the *ExternalMailEnabled* property that's displayed.

Use the Shell to enable or disable content filtering for internal messages

As a best practice, you should not filter messages from trusted partners or from inside your organization. When you run anti-spam filters, there's always a chance that the filters will detect false positives. To reduce the chance that filters will mishandle legitimate email messages, you should enable anti-spam agents to run only on messages from potentially untrusted and unknown sources.

To enable content filtering for internal messages, run the following command:

```
Set-ContentFilterConfig -InternalMailEnabled $true
```

To disable content filtering for internal messages, run the following command:

```
Set-ContentFilterConfig -InternalMailEnabled $false
```

How do you know this worked?

To verify that you have successfully enabled or disabled content filtering for internal messages, do the following:

1. Run the following command:

```
Get-ContentFilterConfig | Format-List InternalMailEnabled
```

2. Verify the value of the *InternalMailEnabled* property that's displayed.

Use the Shell to configure recipient and sender exceptions

To replace the existing values, run the following command:

```
Set-ContentFilterConfig -BypassedRecipients
<recipient1,recipient2...> -BypassedSenders
<sender1,sender2...> -BypassedSenderDomains
<domain1,domain2...>
```

This example configures the following exceptions in in content filtering:

- The recipients laura@contoso.com and julia@contoso.com aren't checked by content filtering.
- The senders steve@fabrikam.com and cindy@fabrikam.com aren't checked by content filtering.
- All senders in the domain nwtraders.com and all subdomains aren't checked by content filtering.

```
Set-ContentFilterConfig -BypassedRecipients
laura@contoso.com,julia@contoso.com -BypassedSenders
steve@fabrikam.com,cindy@fabrikam.com -
BypassedSenderDomains *.nwtraders.com
```

To add or remove entries without modifying any existing values, run the following command:

```
Set-ContentFilterConfig -BypassedRecipients
```

```
@{Add="<recipient1>","<recipient2>"...;
Remove="<recipient1>","<recipient2>"...} -BypassedSenders
@{Add="<sender1>","<sender2>"...;
Remove="<sender1>","<sender2>"...} -BypassedSenderDomains
@{Add="<domain1>","<domain2>"...;
Remove="<domain1>","<domain2>"...}
```

This example configures the following exceptions in content filtering:

- Add tiffany@contoso.com and chris@contoso.com to the list of existing recipients who aren't checked by content filtering.
- Add joe@fabrikam.com and michelle@fabrikam.com to the list of existing senders who aren't checked by content filtering.
- Add blueyonderairlines.com to the list of existing domains whose senders aren't checked by content filtering.
- Remove the domain woodgrovebank.com and all subdomains from the list of existing domains whose senders aren't checked by content filtering.

```
Set-ContentFilterConfig -BypassedRecipients
@{Add="tiffany@contoso.com","chris@contoso.com"} -
BypassedSenders
@{Add="joe@fabrikam.com","michelle@fabrikam.com"} -
BypassedSenderDomains @{Add="blueyonderairlines.com";
Remove="*.woodgrovebank.com"}
```

How do you know this worked?

To verify that you have successfully configured the recipient and sender exceptions, do the following:

1. Run the following command:

```
Get-ContentFilterConfig | Format-List Bypassed*
```

2. Verify the values displayed match the settings you specified.

Use the Shell to configure allowed and blocked phrases

To add allowed and blocked words and phrases, run the following command:

```
Add-ContentFilterPhrase -Influence GoodWord -Phrase
<Phrase> -Influence BadWord -Phrase <Phrase>
```

This example allows all messages that contain the phrase "customer feedback".

```
Add-ContentFilterPhrase -Influence GoodWord -Phrase
```

```
"customer feedback"
```

This example blocks all messages that contain the phrase "stock tip".

```
Add-ContentFilterPhrase -Influence BadWord -Phrase "stock
tip"
```

To remove allowed or blocked phrases, run the following command:

```
Remove-ContentFilterPhrase -Phrase <Phrase>
```

This example removes the phrase "stock tip":

```
Remove-ContentFilterPhrase -Phrase "stock tip"
```

How do you know this worked?

To verify that you have successfully configured the allowed and block phrases, do the following:
1. Run the following command:

```
Get-ContentFilterPhrase | Format-List Influence,Phrase
```

2. Verify the values displayed match the settings you specified.

Use the Shell to configure SCL thresholds

To configure the spam confidence level (SCL) thresholds and actions, run the following command:

```
Set-ContentFilterConfig -SCLDeleteEnabled <$true | $false>
-SCLDeleteThreshold <Value> -SCLRejectEnabled <$true |
$false> -SCLRejectThreshold <Value> -SCLQuarantineEnabled
<$true | $false> -SCLQuarantineThreshold <Value>
```

Note:
The Delete action takes precedence over the Reject action, and the Reject action takes precedence over the Quarantine action. Therefore, the SCL threshold for the Delete action should be greater than the SCL threshold for the Reject action, which in turn should be greater than the SCL threshold for the Quarantine action. Only the Reject action is enabled by default, and it has the SCL threshold value 7.

This example configures the following values for the SCL thresholds:

- The Delete action is enabled and the corresponding SCL threshold is set to 9.
- The Reject action is enabled and the corresponding SCL threshold is set to 8.
- The Quarantine action is enabled and the corresponding SCL threshold is set to 7.

```
Set-ContentFilterConfig -SCLDeleteEnabled $true -
```

```
SCLDeleteThreshold 9 -SCLRejectEnabled $true -
SCLRejectThreshold 8 -SCLQuarantineEnabled $true -
SCLQuarantineThreshold 7
```

How do you know this worked?

To verify that you have successfully configured the SCL thresholds, do the following:
1. Run the following command:

```
Get-ContentFilterConfig | Format-List SCL*
```

2. Verify the values displayed match the settings you specified.

Use the Shell to configure the rejection response

When the Reject action is enabled, you can customize the rejection response that's sent to the message sender. The rejection response can't exceed 240 characters.

To configure a custom rejection response, run the following command:

```
Set-ContentFilterConfig -RejectionResponse "<Custom Text>"
```

This example configures the Content Filter agent to send a customized rejection response.

```
Set-ContentFilterConfig -RejectionResponse "Your message
was rejected because it appears to be SPAM."
```

How do you know this worked?

To verify that you have successfully configured the rejection response, do the following:
1. Run the following command:

```
Get-ContentFilterConfig | Format-List *Reject*
```

2. Verify the values displayed match the settings you specified.

Use the Shell to enable or disable Outlook Email Postmarking

Outlook Email Postmarking validation is a computational proof that Microsoft Outlook applies to outgoing messages to help recipient messaging systems distinguish legitimate email from junk email. Postmarking is available in Outlook 2007 or newer. Postmarking helps reduce false positives. Outlook Email Postmarking is enabled by default.

To disable Outlook Email Postmarking, run the following command:

```
Set-ContentFilterConfig -
OutlookEmailPostmarkValidationEnabled $false
```

To enable Outlook Email Postmarking, run the following command:

```
Set-ContentFilterConfig -
OutlookEmailPostmarkValidationEnabled $true
```

How do you know this worked?

To verify that you have successfully configured Outlook Email Postmarking, do the following:
1. Run the following command:

```
Get-ContentFilterConfig | Format-List
OutlookEmailPostmarkValidationEnabled
```

2. Verify the value displayed matches the setting you specified.

Safelist Aggregation

Anti-spam and anti-malware protection > Anti-spam protection > Content filtering >

Topic Last Modified: *2013-09-30*

In Microsoft Exchange Server 2013, *safelist aggregation* refers to anti-spam functionality shared across Microsoft Outlook and Exchange. This functionality collects data from the anti-spam Safe Recipients Lists, Safe Senders Lists, Blocked Senders Lists, and contact data that Outlook users configure, and makes this data available to the Exchange anti-spam agents.

When you enable and correctly configure safelist aggregation, the Content Filter agent passes safe email messages to the enterprise mailbox without additional processing. Email messages that Outlook users receive from contacts that those users have added to their Outlook Safe Recipients List or Safe Senders List or have trusted are identified by the Content Filter agent as safe. An Outlook *contact* is a person, inside or outside the user's organization, about whom the user can save several types of information, such as email and street addresses, telephone and fax numbers, and Web page URLs.

In Exchange 2013, the safelist aggregation process also allows the Sender Filtering agent to block incoming messages from the per-recipient Block Senders List.

Safelist aggregation can help reduce the instances of false-positives in anti-spam filtering performed by Exchange. In the context of spam filtering, a *false-positive* occurs when a spam filter incorrectly identifies a legitimate message as spam.

For organizations that filter hundreds of thousands of messages from the Internet every day, even a

small percentage of false-positives means that users might not receive many messages that were incorrectly identified as spam if those messages were quarantined or deleted.

Safelist aggregation is likely the most effective way to reduce false-positives. In Outlook 2010 or newer versions, users can create *Safe Senders Lists*. Safe Senders Lists specify a list of domain names and email addresses from which the Outlook user wants to receive messages. By default, email addresses in Outlook Contacts and in the Exchange global address list are included in this list. By default, Outlook adds all external contacts to which the user sends mail to the Safe Senders List.

Contents

Information stored in the Outlook user's safelist collection

A *safelist collection* is the combined data from the user's Safe Senders List, Safe Recipients List, Blocked Senders List, and external contacts. This data is stored in Outlook and in the Exchange mailbox.

The following types of information are stored in an Outlook user's safelist collection:

- **Safe senders and safe recipients** The From message header indicates a sender. The To field of the email message indicates a recipient. Safe senders and safe recipients are represented by full SMTP addresses, such as masato@contoso.com. Outlook users can add senders and recipients to their safe lists.
- **Blocked senders** Just like safe senders, users can block unwanted senders by adding them to their Blocked Senders Lists.
- **Safe domain** The domain is the part of an SMTP address that follows the @ symbol. For example, contoso.com is the domain in the masato@contoso.com address. Outlook users can add sending domains to their safe lists.

Important:
By default, Exchange doesn't include the domains during safelist aggregation. However, you can configure safelist aggregation to include the safe domain data for the anti-spam agents. For more information, see Configure Content Filtering to Use Safe Domain Data.

- **External contacts** Two types of external contacts can be included in the safelist aggregation. The first type of external contact includes contacts to whom Outlook users have sent mail. This class of contact is added to the Safe Senders List only if an Outlook user selects the corresponding option in the Junk Email settings in Outlook 2007.

The second type of external contact includes the users' Outlook contacts. Users can add or import these contacts into Outlook. This class of contact is added to the Safe Senders List only if an Outlook user selects the corresponding option in the Junk Email Filter settings in Outlook 2010 or

Outlook 2007.

Return to top

How Exchange uses the safelist collection

The safelist collection is stored on the user's Mailbox server. A user can have up to 1,024 unique entries in a safelist collection. Exchange 2013 has a mailbox assistant, called the Junk Email Options mailbox assistant, which monitors changes to the safelist collection for your mailboxes. It then replicates these changes to Active Directory, where the safelist collection is stored on each user object. When the safelist collection is stored on the user object in Active Directory, the safelist collection is aggregated with the anti-spam functionality of Exchange 2013 and is optimized for minimized storage and replication. Exchange uses the safelist collection data during content filtering. If you have a subscribed Edge Transport server in your perimeter network, the Microsoft Exchange EdgeSync service replicates the safelist collection to the Active Directory Lightweight Directory Services (AD LDS) instance on the Edge Transport server.

Important:
Although the safe recipient data is stored in Outlook and can be aggregated into the safelist collection, the content filtering functionality doesn't act on safe recipient data.

Return to top

Hashing of safelist collection entries

Safelist collection entries are hashed (SHA-256) one way before they are stored as array sets across three user object attributes, **msExchSafeSenderHash**, **msExchSafeRecipientHash**, and **msExchBlockedSendersHash**, as a binary large object. When data is hashed, an output of fixed length is produced, and the output is likely to be unique. For hashing of safelist collection entries, a 4-byte hash is produced. When a message is received from the Internet, Exchange hashes the sender address and compares it to the hashes stored on behalf of the Outlook user to whom the message was sent. If the sender matches the safe senders hash, the message bypasses content filtering. If the sender matches the blocked senders hash, the message is blocked.

One-way hashing of safelist collection entries performs the following important functions:

- **Minimizes storage and replication space** Most of the time, hashing reduces the size of the data hashed. Therefore, saving and transmitting a hashed version of a safelist collection entry conserves storage space and replication time. For example, a user who has 200 entries in his or her safelist collection would create about 800 bytes of hashed data stored and replicated in Active Directory.
- **Renders user safelist collections unusable by malicious users** Because one-way hash values are impossible to reverse-engineer into the original SMTP address or domain, the safelist collections don't yield usable email addresses for malicious users who might compromise an Exchange server.

Return to top

Enabling safelist aggregation

Safelist aggregation is enabled by default in Exchange 2013. Unlike in Exchange Server 2007, you don't need to manually run the **Update-SafeList** cmdlet to hash and write the safelist collection data to Active Directory. In Exchange 2013, the safelist collection data is written to Active Directory by the Junk Email Options mailbox assistant.

To make the safelist aggregation data in Active Directory available to Edge Transport servers in the perimeter network, you need to install and configure the Microsoft Exchange EdgeSync service so that the safelist aggregation data is replicated to AD LDS.

You can still manually run safelist aggregation by using the **UpdateSafelist** cmdlet. However, you need to be mindful of the network and replication traffic that may be generated when you run this command. Running **Update-Safelist** on multiple mailboxes where safelists are heavily used may generate a significant amount of traffic. We recommend that if you run the command on multiple mailboxes, you should run the command during off-peak, non-business hours.

The **Update-SafeList** cmdlet reads the safelist collection from the user's mailbox, hashes each entry, sorts the entries for easy search, and then converts the hash to a binary attribute. Finally, the **Update-SafeList** cmdlet compares the binary attribute that was created to any value stored on the attribute. If the two values are identical, the **Update-SafeList** cmdlet doesn't update the user attribute value with the safelist aggregation data. If the two attribute values are different, the **Update-SafeList** cmdlet updates the safelist aggregation value.

Return to top

Manage safelist aggregation

Anti-spam protection > Content filtering > Safelist Aggregation >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-10*

Safelist aggregation refers to anti-spam functionality that's shared from Microsoft Outlook to Microsoft Exchange Server 2013. This functionality collects data from the Safe Recipients Lists, Safe Senders Lists, Blocked Senders Lists, and contact data that Outlook users configure, and makes this data available to the Exchange anti-spam agents. Safelist aggregation can help reduce the instances of false-positives in anti-spam filtering performed by the Exchange servers where the anti-spam agents are running.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic, and the "Anti-spam features" section in the Anti-spam and anti-malware permissions topic.
- You can only use the Shell to perform this procedure.
- By default, anti-spam features aren't enabled in the Transport service on a Mailbox server. Typically, you only enable the anti-spam features on a Mailbox server if your Exchange organization doesn't do any prior anti-spam filtering before accepting incoming messages. For more information, see Enable anti-spam functionality on Mailbox servers.
- Be mindful of the network and replication traffic that may be generated when you run the **Update-SafeList** cmdlet. Running the command on multiple mailboxes where safelists are heavily used may generate a significant amount of traffic. We recommend that if you run the command on multiple mailboxes, you should run the command during off-peak, non-business hours.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to configure the mailbox safelist collection limits

You can configure the maximum number of safe senders and blocked senders a user can configure. By default, users can configure up to 5,000 safe senders and 500 blocked senders.

To configure the maximum number of safe senders and blocked senders, run the following command:

```
Set-Mailbox <MailboxIdentity> -MaxSafeSenders <Integer> -
MaxBlockedSenders <Integer>
```

This example configures the mailbox john@contoso.com to have a maximum of 2,000 safe senders and 200 blocked senders.

```
Set-Mailbox john@contoso.com -MaxSafeSenders 2000 -
```

```
MaxBlockedSenders 200
```

How do you know this worked?

To verify that you have successfully configured the mailbox safelist collection limits, do the following:

1. Run the following command:

```
Get-Mailbox <Identity> | Format-List Name,Max*Senders
```

2. Verify the values displayed match the values you configured.

Use the Shell to run the Update-Safelist command

In Exchange 2013, safelist aggregation is done automatically, so you don't need to schedule or manually run the **Update-Safelist** cmdlet. However, you may want to occasionally run this cmdlet to test safelist aggregation.

This example writes the safe senders list for the mailbox john@contoso.com to Active Directory.

```
Update-Safelist john@contoso.com -Type SafeSenders
```

For detailed syntax and parameter information, see Update-SafeList.

How do you know this worked?

To verify that you have successfully configured safelist aggregation, perform the following steps:

Step 1: Use the Shell to verify the Content Filter agent is enabled on the Exchange server

1. Run the following command:

```
Get-ContentFilterConfig | Format-List Enabled
```

2. If the output shows the *Enabled* parameter to be `True`, content filtering is enabled. If it isn't, run the following command to enable content filtering and the Content Filter agent on the Exchange server:

```
Set-ContentFilterConfig -Enabled $true
```

Step 2: (Optional) Use ADSI Edit to verify replication of the safelist aggregation data to Edge Transport servers

This step is only required if you run the Content Filter agent on an Edge Transport server in your perimeter network.

You can view the user objects in the Active Lightweight Directory Services (AD LDS) instance on the Edge Transport server to verify that the safelist collection data is updated for the user objects and that the Microsoft Exchange EdgeSync service has replicated the data to the AD LDS instance.

There are three safelist collection attributes for each user object:

- **msExchSafeRecipientsHash** This attribute stores the hash of the Safe Recipients List collection for the user.
- **msExchSafeSendersHash** This attribute stores the hash of the Safe Senders List collection for the user.
- **msExchBlockedSendersHash** This attribute stores the hash of the Blocked Senders List collection for the user.

If a hexadecimal string, such as `0xac 0xbd 0x03 0xca`, is present on the attribute, the user object was updated. If the attribute has a value of `<Not Set>`, the attribute wasn't updated.

You can search for and view the attributes by using the AD LDS Active Directory Service Interfaces (ADSI) Edit snap-in.

Step 3: Send a test message to verify safelist aggregation is working

To test whether safelist aggregation is functioning, you need to send yourself a message from a safe sender that would otherwise be blocked by content filtering. If safelist aggregation is functioning, the message should arrive in your Inbox.

1. Find an existing external email account to use, or create an email account at a free web-based email provider like Microsoft Hotmail.
2. Add that account to your Safe Senders List in Microsoft Outlook.
3. Use the **Update-SafeList** cmdlet to have the safelist collection from that mailbox copied to Active Directory.
4. Optional: if you are running the Content Filter agent on an Edge Transport server in the perimeter network, run the **Start-EdgeSynchronization** cmdlet to force EdgeSync replication.
5. Add a specific word as a blocked phrase to your content filtering configuration. For detailed steps, see Manage content filtering.
6. From the external email account in step 1, send a message to your Exchange mailbox that includes the blocked phrase you configured in step 5.

If the message is successfully delivered to your Inbox, safelist aggregation is working correctly.

Configure Content Filtering to Use Safe

Domain Data

Topic Last Modified: *2013-09-30*

Safe domain data is an entire domain (for example, @contoso.com) that's stored in a user's Safe Senders List. By default, the Content Filter agent doesn't use safe domain data to identify senders that are allowed to bypass content filtering.

This default setting helps reduce the amount of spam that's delivered into your organization. For example, a user might add the domain of a large email provider to their Safe Senders List. If this domain is frequently used or spoofed by spammers, and if content filtering is configured to use safe domain data to mark the messages as safe, messages from any sender in that domain would be delivered to recipients in your organization.

We recommend that you don't modify the default setting in most cases. However, you can configure users' safe domain data to be stored in Active Directory and used by content filtering to mark messages as safe. To do this, you need to modify the MSExchangeMailboxAssistants.exe.config XML application configuration file that's associated with the Microsoft Exchange Mailbox Assistants service, as detailed later in this topic. When you make this configuration change, the safe domain data is hashed and stored in each user's **msExchSafeSenderHash** user object attribute in Active Directory as part of safelist aggregation. The Content Filter agent can then use the safe domain data to mark messages from senders in those domains as safe. For more information about safelist aggregation, see Safelist Aggregation.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes
- Exchange permissions don't apply to the procedures in this topic. These procedures are performed in the operating system of the Exchange Server.
- Changes you save to the MSExchangeMailboxAssistants.exe.config file are applied after you restart the Microsoft Exchange Mailbox Assistants service.
- Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use a command prompt to configure content filtering to use safe domain data

1. In the Command Prompt window, open the MSExchangeMailboxAssistants.exe.config file in Notepad by running the following command:

```
Notepad %ExchangeInstallPath%Bin
\MSExchangeMailboxAssistants.exe.config
```

2. Locate the *</appsettings>* key at the end of the file, and paste the following key before the *</appsettings>* key:

```
<add key="IncludeSafeDomains" value="true" />
```

3. When you are finished, save and close the MSExchangeMailboxAssistants.exe.config file.
4. Restart the Microsoft Exchange Mailbox Assistants service by running the following command:

```
net stop MSExchangeMailboxAssistants && net start
MSExchangeMailboxAssistants
```

How do you know this worked?

To verify that you have successfully configured content filtering to use safe domain data, do the following:

1. Verify that adding a domain to a user's Safe Senders List in Outlook updates the user's **msExchSafeSenderHash** attribute in Active Directory. To do this, view the attribute in ADSIEdit.exe or LDP.exe, open the user's mailbox in Outlook, add a domain to the Safe Senders List, run the command `Update-Safelist <username>`, and verify that the original and current values of **msExchSafeSenderHash** are different.
2. After you've verified the safe domain data is stored in Active Directory, send a test message from an external sender in that domain to a user in your organization. Verify the message is marked as safe by examining the anti-spam header fields in the message header.

Spam Confidence Level Threshold

Anti-spam and anti-malware protection > Anti-spam protection > Content filtering >

Topic Last Modified: *2013-10-10*

In Microsoft Exchange Server 2013, you can define specific actions according to spam confidence level (SCL) thresholds. For example, you can define different thresholds for rejecting, deleting, or

quarantining messages on an Exchange server that's running the Content Filter agent.

The combination of this SCL threshold configuration in the Content Filter agent and the SCL Junk Email folder configuration on the user's mailbox helps you implement a more comprehensive and precise anti-spam strategy. This more precise and detailed SCL threshold adjustment functionality in Exchange 2013 can help you reduce the overall cost of deploying and maintaining an anti-spam solution across your Exchange organization.

The Content Filter agent assigns an SCL rating to messages late in the anti-spam cycle, after other anti-spam agents have processed any inbound messages. Many of the other anti-spam agents that process inbound messages before they're processed by the Content Filter agent are deterministic in how they act on a message. For example, on an Edge Transport server the Connection Filter agent rejects any message sent from an IP address on a real-time block list. The Sender Filter agent and Recipient Filter agent process messages in a similarly deterministic manner.

In Exchange 2013, these deterministic anti-spam agents process messages first and therefore greatly reduce the number of messages that must be processed by the Content Filter agent. For more information about the order in which anti-spam agents process messages, see Anti-spam protection.

Because content filtering isn't an exact, deterministic process, the ability to adjust the action that the Content Filter agent performs on different SCL values is important. By carefully adjusting the SCL threshold configuration, you can minimize the following:

- Size of the spam quarantine storage
- Number of legitimate email messages mistakenly quarantined
- Number of legitimate email messages that reach the Microsoft Outlook user's Junk Email folder
- Number of offensive spam email messages that reach the Outlook user's Inbox or Junk Email folder
- Number of spam email messages that reach the Outlook user's Inbox

SCL threshold actions

By adjusting SCL threshold actions, you can escalate the content filtering action taken on messages that have a greater risk of being spam. To understand this functionality, it's helpful to understand the different SCL threshold actions and how they're implemented:

- **SCL delete threshold** When the SCL value for a specific message is equal to or higher than the SCL delete threshold, the Content Filter agent deletes the message. There's no protocol-level communication that tells the sending system or sender that the message was deleted. If the SCL value for a message is lower than the SCL delete threshold value, the Content Filter agent doesn't delete the message. Instead, the Content Filter agent compares the SCL value to the SCL reject threshold.
- **SCL reject threshold** When the SCL value for a specific message is equal to or higher than the SCL reject threshold, the Content Filter agent deletes the message and sends a rejection response to the sending system. You can customize the rejection response. In some cases, a non-delivery report (NDR) is sent to the original sender of the message. If the SCL value for a message is lower

than the SCL delete and SCL reject threshold values, the Content Filter agent doesn't delete or reject the message. Instead, the Content Filter agent compares the SCL value to the SCL quarantine threshold.

- **SCL quarantine threshold** When the SCL value for a specific message is equal to or higher than the SCL quarantine threshold, the Content Filter agent sends the message to a quarantine mailbox. Email administrators must periodically review the quarantine mailbox. If the SCL value for a message is lower than the SCL delete, reject, and quarantine threshold values, the Content Filter agent doesn't delete, reject, or quarantine the message. Instead, the Content Filter agent sends the message to the appropriate Mailbox server, where the per-recipient SCL Junk Email folder threshold value of the message is evaluated.
- **SCL Junk Email folder threshold** If the SCL value for a specific message exceeds the SCL Junk Email folder threshold, the message is delivered to the user's Junk Email folder. If the SCL value for a message is lower than the SCL delete, reject, quarantine, and Junk Email folder threshold values, the message is delivered to the user's Inbox.

The Content Filter agent and the Junk Email folder process the SCL threshold value differently. The Content Filter agent takes action on the SCL threshold value that you configure. The Junk Email folder takes action on the SCL threshold value that you configure plus 1. For example, if you configure the Delete action to an SCL of 8 on the Content Filter agent, all messages with an SCL of 8 or greater are deleted. However, if you configure the Junk Email folder with an SCL threshold of 4, all messages with an SCL of 5 or greater are moved to the Junk Email folder.

For example, if you set the SCL delete threshold to 8, the SCL reject threshold to 7, the SCL quarantine threshold to 6, and the SCL Junk Email folder threshold to 4, all messages with an SCL of 5 or lower are delivered to the user's mailbox. Messages with an SCL value of 5 are put in the user's Junk Email folder. Messages with an SCL value of 4 or lower are put in the user's Inbox.

You can configure the SCL delete, reject, quarantine, and Junk Email folder settings in the following locations:

- **On the Content Filter agent configuration (per-transport server SCL configuration)** You use the **Set-ContentFilterConfig** cmdlet to enable or disable and set the SCL delete, reject, and quarantine thresholds on the Exchange server where you're running the Content Filter agent. Over time, as you analyze the spam functionality and metrics provided by the anti-spam logging and reporting features, you can make additional adjustments to these SCL threshold configurations as needed.

The SCL parameters that are available on the **Set-ContentFilterConfig** cmdlet are described in the following table.

Parameter	Description
SCLDeleteEnabled	This parameter enables or disables deleting a message without a non-delivery report (NDR) when the SCL value of the message is greater than or equal to the value specified by the *SCLDeleteThreshold* parameter. Valid input for

	this parameter is `$true` or `$false`.
SCLDeleteThreshold	Valid input for this parameter is an integer from 0 through 9 inclusive. The value of this parameter should be greater than the other SCL threshold parameters. This parameter is only meaningful if the value of the *SCLDeleteEnabled* parameter is `$true`.
SCLRejectEnabled	This parameter enables or disables rejecting a message with an NDR when the SCL value of the message is greater than or equal to the value specified by the *SCLRejectThreshold* parameter. Valid input for this parameter is `$true` or `$false`.
SCLRejectThreshold	Valid input for this parameter is an integer from 0 through 9 inclusive. The value of this parameter should be less than the *SCLDeleteThreshold* parameter, but greater than the other SCL threshold parameters This parameter is only meaningful if the value of the *SCLRejectEnabled* parameter is `$true`.
SCLQuarantineEnabled	This parameter enables or disables sending a message to the spam quarantine mailbox when the SCL value of the message is greater than or equal to the value specified by the *SCLQuarantineThreshold* parameter. Valid input for this parameter is `$true` or `$false`. For more information about the spam quarantine mailbox, see Spam quarantine.
SCLQuarantineThreshold	Valid input for this parameter is an integer from 0 through 9 inclusive. The value of this parameter should be less than the *SCLRejectThreshold* parameter, but greater than the *SCLJunkThreshold* parameter on the

	Set-OrganizationConfig or **Set-Mailbox** cmdlets. This parameter is only meaningful if the value of the *SCLQuarantineThreshold* parameter is `$true`.

- **On the organization configuration settings (organization-wide SCL configuration)** You use the **Set-OrganizationConfig** cmdlet to set the SCL Junk Email folder threshold for all mailboxes in the organization.

The SCL parameter that's available on the **Set-OrganizationConfig** cmdlet is described in the following table. For an example of using SCLJunkThreshold, see Configure Anti-Spam Settings on Mailboxes.

Parameter	Description
SCLJunkThreshold	This parameter specifies the SCL value that a message must exceed for the message to be moved into the Junk Email folder of the recipient's mailbox. Valid input for this parameter is an integer from 0 through 9 inclusive. The value of this parameter should be less than the other SCL threshold parameters. For example, if you specify the value 4, then messages with an SCL value of 5 or higher are moved into the user's Junk Email folder.

- **On user mailboxes (per-recipient SCL configuration)** You use the **Set-Mailbox** cmdlet to enable or disable and set per-recipient SCL delete, reject, quarantine, and Junk Email folder thresholds on individual mailboxes. You can only use the **Set-Mailbox** cmdlet to enable or disable the SCL Junk Email folder threshold on individual mailboxes. The per-recipient SCL delete, reject, and quarantine thresholds are stored in Active Directory and are replicated to subscribed Edge Transport servers by the Microsoft Exchange EdgeSync service. The per-recipient SCL threshold configurations are used by the Content Filter agent even if you've set per-transport server SCL configurations. Therefore, if you've set per-recipient SCL thresholds, the Content Filter agent uses the per-recipient SCL thresholds for specific users instead of the SCL configuration on the Content Filter agent. For examples, see Configure Anti-Spam Settings on Mailboxes.

Note:
Per-recipient SCL thresholds are not enforced on mail received through distribution groups.

The same SCL parameters are available on the **Set-Mailbox** cmdlet that are available on the **Set-ContentFilterConfig** and **Set-OrganizationConfig** cmdlets:

- *SCLDeleteEnabled*
- *SCLDeleteThreshold*

- *SCLRejectEnabled*
- *SCLRejectThreshold*
- *SCLQuarantineEnabled*
- *SCLQuarantineThreshold*
- *SCLJunkThreshold*

However, all the SCL parameters on the **Set-Mailbox** cmdlet also accept the value `$null`. If an SCL setting on a mailbox is blank (`$null`), the corresponding Content Filter agent setting or organization configuration setting is applied to the mailbox. If an SCL setting on a mailbox has the value of `$true` or `$false`, the setting on the mailbox overrides the corresponding organization-wide setting on the Content Filter agent or the organization configuration.

The SCL parameter that's only available on the **Set-Mailbox** cmdlet is described in the following table.

Parameter	Description
SCLJunkEnabled	This parameter enables or disables delivering a message to the user's Junk Email folder when the SCL value of the message is greater than the value specified by the *SCLQuarantineThreshold* parameter. Valid input for this parameter is `$true`, `$false`, or `$null`. Note that junk email filtering is enabled by default for all user mailboxes in the organization. By default, the *Enabled* parameter is set to the value `$true` on the **Set-MailboxJunkEmailConfiguration** cmdlet for all user mailboxes.

For more information about configuring the SCL thresholds on a mailbox, see Configure Anti-Spam Settings on Mailboxes.

Monitoring the SCL thresholds

You can use several built-in scripts located in the `%ExchangeInstallPath%Scripts` folder, such as **get-AntispamSCLHistogram.ps1**, for gathering filtering result data. If the data indicates that you need to make immediate adjustments, reconfigure the SCL thresholds. Otherwise, collect data and analyze the spam reporting to determine whether adjustments are required.

Configure Anti-Spam Settings on Mailboxes

Topic Last Modified: *2014-02-28*

You can configure specific anti-spam settings on individual mailboxes that are different than the anti-spam settings that are applied to the rest of the mailboxes in your Exchange organization. When you configure an anti-spam setting on a mailbox, that setting overrides the corresponding organization-wide content filtering or organization configuration anti-spam setting.

What do you need to know before you begin?

- Estimated time to complete each procedure: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic, and the "Anti-spam" entry in the Recipients Permissions topic.
- By default, anti-spam features aren't enabled in the Transport service on a Mailbox server. Typically, you only enable the anti-spam features on a Mailbox server if your Exchange organization doesn't do any prior anti-spam filtering before accepting incoming messages. For more information, see Enable anti-spam functionality on Mailbox servers.
- You can only use the Shell to perform this procedure.
- The Junk Email Folder SCL threshold value behaves differently than the SCL delete, reject, and quarantine values. For more information, see Spam Confidence Level Threshold.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to configure anti-spam features on a single mailbox

To configure the anti-spam settings on a single mailbox, use the following syntax.

```
Set-Mailbox <MailboxIdentity> -AntispamBypassEnabled <$true
| $false> -RequireSenderAuthenticationEnabled <$true |
$false> -SCLDeleteEnabled <$true | $false | $null> -
SCLDeleteThreshold <0-9 | $null> -SCLJunkEnabled <$true |
$false | $null > -SCLJunkThreshold <0-9 | $null> -
SCLQuarantineEnabled <$true | $false | $null > -
SCLQuarantineThreshold <0-9 | $null> -SCLRejectEnabled
<$true | $false | $null > -SCLRejectThreshold <0-9 | $null>
```

This example configures the mailbox of a user named Jeff Phillips to bypass all the anti-spam filters and to have messages that meet or exceed a Junk Email folder SCL threshold of 5 delivered to his Junk Email folder in Microsoft Outlook.

```
Set-Mailbox "Jeff Phillips" -AntispamBypassEnabled $true -
SCLJunkEnabled $true -SCLJunkThreshold 4
```

How do you know this worked?

To verify that you have successfully configured the anti-spam features on a single mailbox, do the following:
1. Run the following command:

```
Get-Mailbox <MailboxIdentity> | Format-List
SCL*,Bypass*,*SenderAuth*
```

2. Verify the value displayed is the value you configured.

Use the Shell to configure anti-spam features on multiple mailboxes

To configure all the anti-spam settings on multiple mailboxes, use the following syntax.

```
Get-Mailbox [<Filter>]| Set-Mailbox <Anti-Spam Settings>
```

This example enables the SCL quarantine threshold with a value of 7 on all mailboxes in the Users container in the Contoso.com domain.

```
Get-Mailbox -OrganizationalUnit Contoso.com/Users | Set-
Mailbox -SCLQuarantineEnabled $true -SCLQuarantineThreshold
7
```

How do you know this worked?

To verify that you have successfully configured the anti-spam features on multiple mailboxes, do

the following:
1. Run the following command:

```
Get-Mailbox [<Filter>] | Format-List Name,SCL*,*SenderAuth*
```

2. Verify the values displayed are the values you configured.

Use the Shell to configure the junk email threshold for all mailboxes in your organization

Run the following command:

```
Set-OrganizationConfig -SCLJunkThreshold <Integer>
```

This example sets the organization's junk email threshold to 5.

```
Set-OrganizationConfig -SCLJunkThreshold 5
```

How do you know this worked?

To verify that you have successfully configured the junk email threshold for all mailboxes in your organization, do the following:
1. Run the following command:

```
Get-OrganizationConfig | Format-List SCLJunkThreshold
```

2. Verify the value displayed is the value you configured.

Sender reputation and the Protocol Analysis agent

Exchange Server 2013 > Anti-spam and anti-malware protection > Anti-spam protection >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

Sender reputation is part of the Exchange anti-spam functionality that blocks messages according to many characteristics of the sender. Sender reputation relies on persisted data about the sender to determine what action, if any, to take on an inbound message. The Protocol Analysis agent is the underlying agent for sender reputation functionality.

When you configure anti-spam agents on an Exchange server, the agents act on messages

cumulatively to reduce the number of unsolicited messages that enter the organization.

Contents

Calculation of the sender reputation level

A sender reputation level (SRL) is calculated from the following statistics:

- **HELO/EHLO analysis** The HELO and EHLO SMTP commands are intended to provide the domain name, such as Contoso.com, or IP address of the sending SMTP server to the receiving SMTP server. Malicious users, or *spammers*, frequently forge the HELO/EHLO statement in various ways. For example, they type an IP address that doesn't match the IP address from which the connection originated. Spammers also put domains that are known to be locally supported at the receiving server in the HELO statement in an attempt to appear as if the domains are in the organization. In other cases, spammers change the domain that's passed in the HELO statement. The typical behavior of a legitimate user may be to use a different, but relatively constant, set of domains in their HELO statements.

Therefore, analysis of the HELO/EHLO statement on a per-sender basis may indicate that the sender is likely to be a spammer. For example, a sender that provides many different unique HELO/EHLO statements in a specific time period is more likely to be a spammer. Senders who consistently provide an IP address in the HELO statement that doesn't match the originating IP address as determined by the Connection Filter agent are also more likely to be spammers. Remote senders who consistently provide a local domain name in the HELO statement that's in the same organization as the Exchange server are also more likely to be spammers.

- **Reverse DNS lookup** Sender reputation also verifies that the originating IP address from which the sender transmitted the message matches the registered domain name that the sender submits in the HELO or EHLO SMTP command.

Sender reputation performs a reverse DNS query by submitting the originating IP address to DNS. The result that's returned by DNS is the domain name that's registered by using the domain naming authority for that IP address. Sender reputation compares the domain name that's returned by DNS to the domain name that the sender submitted in the HELO/EHLO SMTP command. If the domain names don't match, the sender is likely to be a spammer, and the overall SRL rating for the sender is increased.

The Sender ID agent performs a similar task, but the success of the Sender ID agent relies on legitimate senders to update their DNS infrastructure to identify all the email-sending SMTP servers in their organization. By performing a reverse DNS lookup, you can help identify potential spammers.

- **Analysis of SCL ratings on messages from a particular sender** When the Content Filter agent

processes a message, it assigns a spam confidence level (SCL) rating to the message. The SCL rating is a number from 0 through 9. A higher SCL rating indicates that a message is more likely to be spam. Data about each sender and the SCL ratings that their messages yield is persisted for analysis by sender reputation. Sender reputation calculates statistics about a sender according to the ratio between all messages from that sender that had a low SCL rating in the past and all messages from that sender that had a high SCL rating in the past. Additionally, the number of messages that have a high SCL rating that the sender has sent in the last day is applied to the overall SRL.

- **Sender open proxy test** An *open proxy* is a proxy server that accepts connection requests from anyone anywhere and forwards the traffic as if it originated from the local hosts. Proxy servers relay TCP traffic through firewall hosts to provide user applications transparent access across the firewall. Because proxy protocols are lightweight and independent of user application protocols, proxies can be used by many different services. Proxies can also be used to share a single Internet connection by multiple hosts. Proxies are usually set up so that only trusted hosts inside the firewall can cross through the proxies. A legitimate sender may be an open proxy because of an unintentional misconfiguration or malware.

Open proxies provide an ideal way for malicious users to hide their true identities and launch denial of service attacks (DoS) or send spam. As more proxy servers are configured to be open by default, open proxies have become more common. Additionally, malicious users can use multiple open proxies together to hide the sender's originating IP address.

When sender reputation performs an open proxy test, it does so by formatting an SMTP request in an attempt to connect back to the Exchange server from the open proxy. If an SMTP request is received from the proxy, sender reputation verifies that the proxy is an open proxy and updates the open proxy test statistic for that sender.

Sender reputation weighs each of these statistics and calculates an SRL for each sender. The SRL is a number from 0 through 9 that predicts the probability that a specific sender is a spammer or otherwise malicious user. A value of 0 indicates that the sender isn't likely to be a spammer; a value of 9 indicates that the sender is likely to be a spammer.

You can configure a block threshold from 0 through 9 at which sender reputation issues a request to the Sender Filter agent, and, therefore, blocks the sender from sending a message into the organization. When a sender is blocked, the sender is added to the Blocked Senders list for a configurable period. How blocked messages are handled depends on the configuration of the Sender Filter agent. The following actions are the options for handling blocked messages:

- Reject
- Delete and archive
- Accept and mark as a blocked sender

If a sender is included in the IP Block list or Microsoft IP Reputation Service, sender reputation issues an immediate request to the Sender Filter agent to block the sender. To take advantage of this functionality, you must enable and configure the Microsoft Exchange Anti-spam Update Service.

By default, sender reputation sets a rating of 0 for senders that haven't been analyzed. After a

sender has sent 20 or more messages, sender reputation calculates an SRL that's based on the statistics listed earlier in this topic.

Return to top

Use of the SRL

Sender reputation acts on messages during two phases of the SMTP session:

- **At the MAIL FROM: SMTP command** Sender reputation acts on a message only if the message was blocked or otherwise acted on by the Connection Filter agent, Sender Filter agent, Recipient Filter agent, or Sender ID agent. In this case, sender reputation retrieves the sender's current SRL rating from the sender profile that's persisted about that sender on the Exchange server. After this rating is retrieved and evaluated, the Exchange server configuration dictates the behavior that occurs at a particular connection according to the block threshold.
- **After the "end of data" SMTP command** The end of data transfer (**EOD**) SMTP command is given when all the actual message data is sent. At this point in the SMTP session, many of the anti-spam agents have processed the message. As a by-product of anti-spam processing, the statistics that sender reputation relies on are updated. Therefore, sender reputation has the data to calculate or recalculate an SRL rating for the sender.

For more information, see Manage sender reputation.

Return to top

Enabling and configuring the detection of open proxy servers

Sender reputation evaluates several sender characteristics to calculate an SRL. Among the characteristics that sender reputation evaluates are the results of a test for open proxy servers. Frequently, spammers route messages through open proxy servers on the Internet. By routing spam through open proxy servers, spammers can send messages that appear to originate from a different server than their own.

When sender reputation calculates an SRL, sender reputation tries to connect to the sender's originating IP address by using a variety of common proxy protocols, such as SOCKS4, SOCKS5, HTTP, Telnet, Cisco, and Wingate. Sender reputation formats a protocol-specific request in an attempt to connect back to the Edge Transport server from the open proxy server by using an SMTP request. If an SMTP request is received from the proxy server, sender reputation verifies that the proxy server is an open proxy server and adjusts the SRL rating according to this result. By default, detection of open proxy servers is enabled on sender reputation.

For more information about how to enable and configure detection of open proxy servers, see Manage sender reputation.

Return to top

Setting the SRL block threshold

The SRL is a number from 0 through 9 that predicts the probability that a specific sender is a spammer or otherwise malicious user. You must set a threshold for sender blocking by SRL. This SRL block threshold defines the SRL value that must be exceeded for sender reputation to block a sender. By default, the SRL is set at 7. You should monitor the effectiveness of the agent at the default level. You may find that you can adjust the value to meet the needs of your organization. If you set other anti-spam agents aggressively, you may be able to set a higher SRL threshold for sender reputation than you would if the other anti-spam agents weren't set aggressively. For more information about how to adjust anti-spam configurations so that they work together to reduce spam, see Anti-spam protection.

On an Edge Transport server, if the SRL block threshold is exceeded for a particular sender, sender reputation adds the sender to the IP Block list on the Connection Filter agent. Sometimes, spammers send batches of spam from a single sender. In this scenario, if sender reputation calculates an SRL that exceeds the SRL block threshold, the sender is added to the Sender Block List for a configurable duration of time. The default duration is 24 hours. After 24 hours, the sender is removed from the Sender Block List and can send messages again.

When a sender is added to the IP Block list, sender reputation deletes the profile for the sender. Sender reputation deletes the profile because the blocked sender's existing profile indicates that the sender's SRL exceeds the SRL block threshold. This would cause the blocked sender to be added to the IP Block list again as soon as the duration for sender blocking ends.

For more information, see Manage sender reputation.

Return to top

Manage sender reputation

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Sender reputation is provided by the Protocol Analysis agent. Sender reputation blocks messages according to various characteristics of the sender. Sender reputation relies on persisted data about the sender to determine what action, if any, to take on an inbound message.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.
- You can only use the Shell to perform this procedure.
- By default, anti-spam features aren't enabled in the Transport service on a Mailbox server. Typically, you only enable the anti-spam features on a Mailbox server if your Exchange organization doesn't do any prior anti-spam filtering before accepting incoming messages. For more information, see Enable anti-spam functionality on Mailbox servers.
- The Protocol Analysis agent is the underlying agent for sender reputation functionality. When you disable sender reputation, the Protocol Analysis agent is still enabled.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable sender reputation

This example disables sender reputation.

```
Set-SenderReputationConfig -Enabled $false
```

This example enables sender reputation.

```
Set-SenderReputationConfig -Enabled $true
```

How do you know this worked?

To verify that you have successfully enabled or disabled sender reputation, do the following:
1. Verify the Protocol Analysis agent is installed and enabled by running the following command:

```
Get-TransportAgent
```

2. Verify the sender reputation values you configured by running the following command:

```
Get-SenderReputationConfig | Format-List
Enabled,*MailEnabled
```

Use the Shell to enable or disable sender reputation for internal or external messages

By default, sender reputation is enabled for external messages, and disabled for internal messages. A message is considered external if it comes from an unauthenticated connection that's external to your Exchange organization. A message is considered internal if it comes from authenticated connection, and the sender's domain is configured as an authoritative domain in your Exchange organization.

To disable sender reputation for external messages, run the following command:

```
Set-SenderReputationConfig -ExternalMailEnabled $false
```

To enable sender reputation for external messages, run the following command:

```
Set-SenderReputationConfig -ExternalMailEnabled $true
```

To disable sender reputation for internal messages, run the following command:

```
Set-SenderReputationConfig -InternalMailEnabled $false
```

To enable sender reputation for internal messages, run the following command:

```
Set-SenderReputationConfig -InternalMailEnabled $true
```

How do you know this worked?

To verify that you have successfully enabled or disabled sender reputation for internal and external messages, do the following:
1. Run the following command:

```
Get-SenderReputationConfig | Format-List
Enabled,*MailEnabled
```

2. Verify the values displayed match the values you configured.

Use the Shell to configure sender reputation properties

To configure the sender reputation properties, run the following command:

```
Set-SenderReputationConfig -SrlBlockThreshold <Value> -
SenderBlockingPeriod <Hours>
```

This example sets the sender reputation level (SRL) block threshold to 6 and configures sender reputation to add offending senders to the IP Block List for 36 hours:

```
Set-SenderReputationConfig -SrlBlockThreshold 6 -
SenderBlockingPeriod 36
```

How do you know this worked?

To verify that you have successfully configured the sender reputation properties, do the following:
1. Run the following command:

```
Get-SenderReputationConfig
```

2. Verify the values displayed match the values you configured.

Use the Shell to configure outbound access for the detection of open proxy servers

You may need to perform additional steps to allow sender reputation to traverse any firewalls that are between the Internet and the Exchange server that's running the Protocol Analysis agent. The following table lists the outbound ports that are required for sender reputation.

Protocols	Ports
SOCKS4, SOCKS5	1081, 1080
Wingate, Telnet, Cisco	23
HTTP CONNECT, HTTP POST	6588, 3128, 80

To configure outbound access for the detection of open proxy servers, run the following command:

```
Set-SenderReputationConfig -ProxyServerName <String> -
ProxyServerPort <Port> -ProxyServerType <String>
```

This example configures sender reputation to use the open proxy server named SERVER01 that uses the HTTP CONNECT protocol on port 80.

```
Set-SenderReputationConfig - ProxyServerName SERVER01 -
ProxyServerPort 80 -ProxyServerType HttpConnect
```

How do you know this worked?

To verify that you have successfully configured outbound access for detection of open proxy servers, do the following:
1. Run the following command:

```
Get-SenderReputationConfig | Format-List ProxyServer*
```

2. Verify the values displayed are the values you configured.

Connection Filtering on Edge Transport Servers

Exchange Server 2013 > Anti-spam and anti-malware protection > Anti-spam protection >

Topic Last Modified: *2014-02-10*

Connection filtering is an anti-spam feature in Microsoft Exchange Server 2013 that allows or blocks email based on the message source. Connection filtering is performed by the Connection Filtering agent that's available only on Edge Transport servers. The Connection Filtering agent relies on the IP address of the connecting mail server to determine what action, if any, to take on an inbound message.

By default, the Connection Filtering agent is the first anti-spam agent to evaluate an inbound message on an Edge Transport server. The source IP address of the SMTP connection is checked against the allowed and blocked IP addresses. If the source IP address is specifically allowed, the message is sent to the recipients in your organization without additional processing by other anti-spam agents. If the source IP address is specifically blocked, the SMTP connection is dropped. If the source IP address isn't specifically allowed or blocked, the message flows through the other anti-spam agents on the Edge Transport server.

Connection filtering compares the IP address of the source mail server to the values in the IP Allow list, the IP Block list, IP Allow list providers, and IP Block list providers. You need to configure at least one of these four IP address data stores for connection filtering to function. If you don't specify any IP address data, you should disable the Connection Filtering agent. For more information, see Manage Connection Filtering on Edge Transport Servers.

Contents

IP Block list

The IP Block list contains the IP addresses of email servers that you want to block. You manually maintain the IP addresses in the IP Block list. You can add individual IP addresses or IP address ranges. You can specify an expiration time that specifies how long the IP address entry will be blocked. When the expiration time is reached, the IP address entry in the IP Block list is disabled.

If the Connection Filtering agent finds the source IP address on the IP Block list, the SMTP connection will be dropped after all the **RCPT TO** headers (envelope recipients) in the message are processed.

IP addresses can also be automatically added to the IP Block list by the Sender Reputation feature of the Protocol Analysis agent. For more information, see Sender reputation and the Protocol Analysis agent.

IP Allow list

The IP Allow list contains the IP addresses of email servers that you want to designate as trustworthy sources of email. Email from mail servers that you specify in the IP Allow list is exempt from processing by other Exchange anti-spam agents.

You manually maintain the IP addresses in the IP Allow list. You can add individual IP addresses or IP address ranges. You can specify an expiration time that specifies how long the IP address entry will be allowed. When the expiration time is reached, the entry in the IP Allow list is disabled.

Return to top

IP Block List providers

IP Block List providers are frequently referred to as *real-time block lists*, or RBLs. IP Block List providers compile lists of mail server IP addresses that send spam. Many IP Block List providers also compile lists of mail server IP addresses that could be used for spam. Examples include mail servers that are configured for open relay, Internet service providers (ISPs) that assign dynamic IP addresses, and ISPs that allow SMTP mail server traffic from dial-up accounts.

When you configure connection filtering to use an IP Block List provider, the Connection Filtering agent compares the IP address of the connecting mail server to the list of IP addresses at the IP Block List provider. If there's a match, the message isn't allowed in your organization. You can configure connection filtering to use multiple IP Block List providers, and you assign different priority values to each provider.

The Connection Filtering agent checks the source IP address at the IP Allow list and the IP Block list. If the IP address doesn't exist on either list, the Connection Filtering agent queries the IP Block List provider according to the priority value that you assigned. If the IP address is defined at an IP Block List provider, the Edge Transport server waits for and processes the **RCPT TO** header, responds to the sending mail server with an `SMTP 550` error, and closes the connection. The connection isn't immediately dropped so that the connection attempt can be logged, and because you can specify recipients that are exempt from having messages blocked by any IP Block list providers.

If the IP address isn't defined at any of the IP Block List providers, the Content Filtering agent hands the message off to the next transport agent on the Edge Transport server.

For each IP Block List provider, you can customize the SMTP 550 error that's returned to the sender when a message is blocked. You should identify the IP Block List provider that identified the message source as spam. If a legitimate source mail server is erroneously identified as a spam source, the administrator can then contact the IP Block List provider and take the steps necessary to remove the mail server from the IP Block List provider.

IP Block List providers can return different codes to identify why an IP address is defined in their lists. Most IP Block List providers return bitmask or absolute value data types. Within these data types, the IP Block List provider can use multiple values to classify the IP address by threat type.

There are issues to consider when using IP Block list providers:

- Outages or delays at the IP Block list provider service can cause delays in the processing of messages on the Edge Transport server. You should always select reliable IP Block list providers.
- Source servers that you know to be legitimate can be erroneously identified as spam sources. For example, the mail server can be unintentionally configured to act as an open relay. You should always select IP Block list providers that provide clear procedures for evaluation and removal from their services.

Return to top

Bitmask and absolute value examples

This section shows an example of the status codes returned by most Block List providers. For details about the status codes that the provider returns, see the documentation from the specific provider.

For bitmask data types, the IP Block List provider service returns a status code of 127.0.0.*x*, where the integer *x* is any one of the values listed in the following table.

Values and status codes for bitmask data types

Value	Status code
1	The IP address is on an IP Block list.
2	The SMTP server is configured to act as an open relay.
4	The IP address supports a dial-up IP address.

For absolute value types, the IP Block List provider returns explicit responses that define why the IP address is defined in their block lists. The following table shows examples of absolute values and the explicit responses.

Values and status codes for absolute value data types

Value	Explicit response
127.0.0.2	The IP address is a direct spam source.
127.0.0.4	The IP address is a bulk mailer.
127.0.0.5	The remote server sending the message is known to support multistage open relays.

Return to top

IP Allow List providers

IP Allow List providers are also known as *safe lists* or *white lists*. IP Allow List providers are configured just like IP Block List providers, but the results are the opposite: they define mail server IP addresses that are definitely not associated with spam activity. If the IP address of the connecting mail server is defined at an IP Allow List provider, the message is exempt from processing by other Exchange anti-spam agents. For this reason, IP Block List providers are used much more frequently than IP Allow List providers. Choose your IP Allow List providers carefully.

Return to top

Test IP Block List providers and IP Allow List providers

After you configure connection filtering to use an IP Block List provider or an IP Allow List provider, you can run tests to verify that the providers are working correctly. Most providers provide test IP addresses that you can use to test their services. When you test a provider, the Connection Filtering agent issues a DNS query that should result in a specific response from the provider. For more information about how to test IP addresses against an IP Block List provider service or an IP Allow List provider service, see Manage Connection Filtering on Edge Transport Servers.

Return to top

Configure connection filtering on Edge Transport servers that aren't directly connected to the Internet

You can use connection filtering on Edge Transport servers that don't directly receive email from the Internet. In this scenario, the Edge Transport server is behind another mail server that receives and processes messages directly from the Internet. For example, your organization might send email traffic through an anti-spam server, service, or appliance before the messages reach the Edge Transport server. In this scenario, the Connection Filtering agent needs to extract the correct source IP address from the message. To do this, the Connection Filtering agent needs to parse the

Received header field values in the message header and compare those values to the known IP addresses of the mail server that sits between the Edge Transport server and the Internet.

Every mail server that accepts and relays an SMTP message along the delivery path adds its own **Received** header field in the message header. The **Received** header typically contains the domain name and IP address of the mail server that processed the message.

If the Edge Transport server doesn't accept messages directly from the Internet, you need to use the *InternalSMTPServers* parameter on the **Set-TransportConfig** cmdlet on an Exchange 2013 Mailbox server to identify the IP address of the mail server that sit between the Edge Transport server and the Internet. The IP address data is replicated to Edge Transport servers by EdgeSync. When messages are received by the Edge Transport server, the Connection Filtering agent assumes an IP address in a **Received** header field that doesn't match a value specified by the *InternalSMTPServers* parameter is the source IP address that needs to be checked. Therefore, you need specify all internal SMTP servers in order for connection filtering to function correctly.

Return to top

Manage Connection Filtering on Edge Transport Servers

Topic Last Modified: *2014-02-10*

Connection filtering is an anti-spam feature that's provided by the Connection Filtering agent, which is available only on Edge Transport servers in Microsoft Exchange 2013. Connection filtering enables the following features:

- IP Block list
- IP Block List providers
- IP Allow list
- IP Allow List providers

Each of these features can be enabled or disabled separately.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.
- You can only use the Shell to perform this procedure.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable connection filtering

To completely enable or disable connection filtering, you enable or disable the Connection Filtering agent. The change takes effect after you restart the Microsoft Exchange Transport service. When you restart the Microsoft Exchange Transport service on an Edge Transport server, mail flow on the server is temporarily interrupted.

To disable connection filtering, run the following command:

```
Disable-TransportAgent "Connection Filtering Agent"
```

To enable connection filtering, run the following command:

```
Enable-TransportAgent "Connection Filtering Agent"
```

To make the change take effect, restart the Microsoft Exchange Transport service by running the following command:

```
Restart-Service MSExchangeTransport
```

How do you know this worked?

To verify that you successfully enabled or disabled connection filtering, run the following command and verify that the value displayed is the value you configured.

```
Get-TransportAgent "Connection Filtering Agent" | Format-
List Enabled
```

IP Block list procedures

These procedures apply to the IP Block list that you manually configure. They don't apply to IP Block List providers.

Use the **IPBlockListConfig** cmdlets to view and configure how connection filtering uses the IP Block list. Use the **IPBlockListEntry** cmdlets to view and configure the IP addresses in the IP Block list.

Use the Shell to view the configuration of the IP Block list

To view the configuration of the IP Block list, run the following command:

```
Get-IPBlockListConfig | Format-List *Enabled,*Response
```

Use the Shell to enable or disable the IP Block list

To disable the IP Block list, run the following command:

```
Set-IPBlockListConfig -Enabled $false
```

To enable the IP Block list, run the following command:

```
Set-IPBlockListConfig -Enabled $true
```

How do you know this worked?

To verify that you successfully enabled or disabled the IP Block list, run the following command and verify that the value displayed is the value you configured.

```
Get-IPBlockListConfig | Format-List Enabled
```

Use the Shell to configure the IP Block list

To configure the IP Block list, use the following syntax:

```
Set-IPBlockListConfig [-ExternalMailEnabled <$true | $false>] [-InternalMailEnabled <$true | $false> -MachineEntryRejectionResponse "<Custom response text>"] [-StaticEntryRejectionResponse "<Custom response text>"]
```

This example configures the IP Block list with the settings as follows:

- The IP Block list filters incoming connections from internal and external mail servers. By default, connections are filtered from external mail servers only (*ExternalMailEnabled* is set to `$true`, and *InternalMailEnabled* is set to `$false`). Non-authenticated connections and authenticated connections from external partners are considered external.
- The custom response text for connections that were filtered by IP addresses that were automatically added to the IP Block list by the sender reputation feature of the Protocol Analysis agent is set to the value "Connection from IP address {0} was rejected by sender reputation."
- The custom response text for connections that were filtered by IP addresses that were manually added to the IP Block list is set to the value "Connection from IP address {0} was rejected by connection filtering."

```
Set-IPBlockListConfig -InternalMailEnabled $true -MachineEntryRejectionResponse "Connection from IP address {0} was rejected by sender reputation." -
```

```
StaticEntryRejectionResponse "Connection from IP address
{0} was rejected by connection filtering."
```

How do you know this worked?

To verify that you successfully configured the IP Block list, run the following command and verify that the values displayed are the values you configured.

```
Get-IPBlockListConfig | Format-List *MailEnabled,*Response
```

Use the Shell to view IP Block list entries

To view all IP Block list entries, run the following command:

```
Get-IPBlockListEntry
```

Note that each IP Block list entry is identified by an integer value. The identity integer is assigned in ascending order when you add entries to the IP Block list and the IP Allow list.

To view a specific IP Block list entry, use the following syntax:

```
Get-IPBlockListEntry <-Identity IdentityInteger | -
IPAddress IPAddress>
```

For example, to view the IP Block list entry that contains the IP address 192.168.1.13, run the following command:

```
Get-IPBlockListEntry -IPAddress 192.168.1.13
```

Note:

When you use the *IPAddress* parameter, the resulting IP Block list entry can be an individual IP address, an IP address range, or a Classless InterDomain Routing (CIDR) IP. To use the *Identity* parameter, you specify the integer value that's assigned to the IP Block list entry.

Use the Shell to add IP Block list entries

To add IP Block list entries, use the following syntax:

```
Add-IPBlockListEntry <-IPAddress IPAddress | -IPRange IP
range or CIDR IP> [-ExpirationTime <DateTime>] [-comment
"<Descriptive Comment>"]
```

The following example adds the IP Block list entry for the IP address range 192.168.1.10 through 192.168.1.15 and configures the IP Block list entry to expire on July 4, 2014 at 15:00.

```
Add-IPBlockListEntry -IPRange 192.168.1.10-192.168.1.15 -
ExpirationTime "7/4/2014 15:00"
```

How do you know this worked?

To verify that you successfully added an IP Block list entry, run the following command and verify that the new IP Block list entry is displayed.

```
Get-IPBlockListEntry
```

Use the Shell to remove IP Block list entries

To remove IP Block list entries, use the following syntax:

```
Remove-IPBlockListEntry <IdentityInteger>
```

The following example removes the IP Block list entry that has the *Identity* value 3.

```
Remove-IPBlockListEntry 3
```

The following example removes the IP Block list entry that contains the IP address 192.168.1.12 without using the *Identity* integer value. Note that the IP Block list entry can be an individual IP address or an IP address range.

```
Get-IPBlockListEntry -IPAddress 192.168.1.12 | Remove-
IPBlockListEntry
```

How do you know this worked?

To verify that you successfully removed an IP Block list entry, run the following command and verify that the IP Block list entry you removed is gone.

```
Get-IPBlockListEntry
```

IP Block List provider procedures

These procedures apply to IP Block List providers. They don't apply to the IP Block list.

Use the **IPBlockListProvidersConfig** cmdlets to view and configure how connection filtering uses all IP Block List providers. Use the **IPBlockListProvider** cmdlets to view, configure, and test IP Block List providers.

Use the Shell to view the configuration of all IP Block List providers

To view how connection filtering uses all IP Block List providers, run the following command:

```
Get-IPBlockListProvidersConfig | Format-List
*Enabled,Bypassed*
```

Use the Shell to enable or disable all IP Block List providers

To disable all IP Block List providers, run the following command:

```
Set-IPBlockListProvidersConfig -Enabled $false
```

To enable all IP Block List providers, run the following command:

```
Set-IPBlockListProvidersConfig -Enabled $true
```

How do you know this worked?

To verify that you enabled or disabled all IP Block List providers, run the following command an verify that the value displayed is the value you configured.

```
Get-IPBlockListProvidersConfig | Format-List Enabled
```

Use the Shell to configure all IP Block List providers

To configure how connection filtering uses all IP Block List providers, use the following syntax:

```
Set-IPBlockListProvidersConfig [-BypassedRecipients
<recipient1,recipient2...>] [-ExternalMailEnabled <$true |
$false>] [-InternalMailEnabled <$true | $false>]
```

The following example configures all IP Block List providers with the following settings:

- IP Block List providers filter incoming connections from internal and external mail servers. By default, connections are filtered from external mail servers only (*ExternalMailEnabled* is set to `$true`, and *InternalMailEnabled* is set to `$false`). Non-authenticated connections and authenticated connections from external partners are considered external.
- Messages sent to the internal recipients chris@fabrikam.com and michelle@fabrikam.com are excluded from filtering by IP Block List providers. Note that if you want to add recipients to the list without affecting existing recipients, use the syntax, `@{Add="<recipient1>","<recipient2>"...}`.

```
Set-IPBlockListProvidersConfig -BypassedRecipients
chris@fabrikam.com,michelle@fabrikam.com -
InternalMailEnabled $true
```

For more information, see Set-IPBlockListProvidersConfig.

How do you know this worked?

To verify that you successfully configured all IP Block List providers, run the following command and verify that the values displayed are the values you configured.

```
Get-IPBlockListProvidersConfig | Format-List
*MailEnabled,Bypassed*
```

Use the Shell to view IP Block List providers

To view the summary list of all the IP Block List providers, run the following command:

```
Get-IPBlockListProvider
```

To view the details of a specific provider, use the following syntax:

```
Get-IPBlockListProvider <IPBlockListProviderIdentity>
```

The following example show the details of the provider named Contoso IP Block List Provider.

```
Get-IPBlockListProvider "Contoso IP Block List Provider" |
Format-List
Name,Enabled,Priority,LookupDomain,*Match,*Response
```

Use the Shell to add an IP Block List provider

To add an IP Block List provider, use the following syntax:

```
Add-IPBlockListProvider -Name "<Descriptive Name>" -
LookupDomain <FQDN> [-Priority <Integer>] [-Enabled <$true
| $false>] [-AnyMatch <$true | $false>] [-BitmaskMatch
<IPAddress>] [-IPAddressesMatch
<IPAddressStatusCode1,IPAddressStatusCode2...>] [-
RejectionResponse "<Custom Text>"]
```

This example creates an IP Block List provider named "Contoso IP Block List Provider" with the following options:

- **FQDN to use the provider** rbl.contoso.com
- **Bitmask code to use from the provider** 127.0.0.1

```
Add-IPBlockListProvider -Name "Contoso IP Block List
Provider" -LookupDomain rbl.contoso.com -BitmaskMatch
127.0.0.1
```

Note:
When you add a new IP Block List provider, it's enabled by default (the value of *Enabled* is `$true`), and the priority value is incremented (the first entry has the *Priority* value 1).

For more information, see Add-IPBlockListProvider.

How do you know this worked?

To verify that you successfully added an IP Block List provider, run the following command and verify that the new IP Block List provider is displayed.

```
Get-IPBlockListProvider
```

Use the Shell to enable or disable an IP Block List provider

To enable or disable a specific IP Block List provider, use the following syntax:

```
Set-IPBlockListProvider <IPBlockListProviderIdentity> -
Enabled <$true | $false>
```

The following example disables the provider named Contoso IP Block List Provider.

```
Set-IPBlockListProvider "Contoso IP Block List Provider" -
Enabled $false
```

The following example enables the provider named Contoso IP Block List Provider.

```
Set-IPBlockListProvider "Contoso IP Block List Provider" -
Enabled $true
```

How do you know this worked?

To verify that you successfully enabled or disabled an IP Block List provider, run the following command and verify that the value displayed is the value you configured.

```
Get-IPBlockListProvider <IPBlockListProviderIdentity> |
Format-List Enabled
```

Use the Shell to configure an IP Block List provider

The configuration options that are available on the **Set-IPBlockListProvider** cmdlet are identical to those on the **New-IPBlockListProvider** cmdlet.

To configure an existing IP Block List provider, use the following syntax:

```
Set-IPBlockListProvider <IPBlockListProviderIdentity> -Name
"<Descriptive Name>" -LookupDomain <FQDN> [-Priority
<Integer>] [-AnyMatch <$true | $false>] [-BitmaskMatch
<IPAddress>] [-IPAddressesMatch
<IPAddressStatusCode1,IPAddressStatusCode2...>] [-
RejectionResponse "<Custom Text>"]
```

For example, to add the IP address status code 127.0.0.1 to the list of existing status codes for the provider named Contoso IP Block List Provider, run the following command:

```
Set-IPBlockListProvider "Contoso IP Block List Provider" -
```

```
IPAddressesMatch @{Add="127.0.0.1"}
```

For more information, see Set-IPBlockListProvider.

How do you know this worked?

To verify that you successfully configured an IP Block List provider, run the following command and verify that the values displayed are the values you configured.

```
Get-IPBlockListProvider <IPBlockListProviderIdentity> |
Format-List
```

Use the Shell to test an IP Block List provider

To test an IP Block List provider, use the following syntax.

```
Test-IPBlockListProvider <IPBlockListProviderIdentity> -
IPAddress <IPAddressToTest>
```

The following example tests the provider named Contoso IP Block List Provider by looking up the IP address 192.168.1.1.

```
Test-IPBlockListProvider "Contoso IP Block List Provider" -
IPAddress 192.168.1.1
```

Use the Shell to remove an IP Block List provider

To remove an IP Block List provider, use the following syntax:

```
Remove-IPBlockListProvider <IPBlockListProviderIdentity>
```

The following example removes the IP Block List provider named Contoso IP Block List Provider.

```
Remove-IPBlockListProvider "Contoso IP Block list Provider"
```

How do you know this worked?

To verify that you successfully removed an IP Block List provider, run the following command and verify that the IP Block List provider you removed is gone.

```
Get-IPBlockListProvider
```

IP Allow list procedures

These procedures apply to the IP Allow list that you manually configure. They don't apply to IP Allow List providers.

Use the **IPAllowListConfig** cmdlets to view and configure how connection filtering uses the IP Allow list. Use the **IPAllowListEntry** cmdlets to view and configure the IP addresses in the IP Allow list.

Use the Shell to view the configuration of the IP Allow list

To view the configuration of the IP Allow list, run the following command.

```
Get-IPAllowListConfig | Format-List *Enabled
```

Use the Shell to enable or disable the IP Allow list

To disable the IP Allow list, run the following command:

```
Set-IPAllowListConfig -Enabled $false
```

To enable the IP Allow list, run the following command:

```
Set-IPAllowListConfig -Enabled $true
```

How do you know this worked?

To verify that you successfully enabled or disabled the IP Allow list, run the following command and verify that the value displayed is the value you configured.

```
Get-IPAllowListConfig | Format-List Enabled
```

Use the Shell to configure the IP Allow list

To configure the IP Allow list, use the following syntax:

```
Set-IPAllowListConfig [-ExternalMailEnabled <$true |
$false>] [-InternalMailEnabled <$true | $false>
```

This example configures the IP Allow list to filter incoming connections from internal and external mail servers. By default, connections are filtered from external mail servers only (*ExternalMailEnabled* is set to `$true`, and *InternalMailEnabled* is set to `$false`). Non-authenticated connections and authenticated connections from external partners are considered external.

```
Set-IPAllowListConfig -InternalMailEnabled $true
```

How do you know this worked?

To verify that you successfully configured the IP Allow list, run the following command and verify that the values displayed are the values you configured.

```
Get-IPAllowListConfig | Format-List *MailEnabled
```

Use the Shell to view IP Allow list entries

To view all IP Allow list entries, run the following command:

```
Get-IPAllowListEntry
```

Note that each IP Allow list entry is identified by an integer value. The identity integer is assigned in ascending order when you add entries to the IP Block list and the IP Allow list.

To view a specific IP Allow list entry, use the following syntax:

```
Get-IPAllowListEntry <-Identity IdentityInteger | -
IPAddress IPAddress>
```

For example, to view the IP Allow list entry that contains the IP address 192.168.1.13, run the following command:

```
Get-IPAllowListEntry -IPAddress 192.168.1.13
```

Note:

When you use the *IPAddress* parameter, the resulting IP Allow list entry can be an individual IP address, an IP address range, or a Classless InterDomain Routing (CIDR) IP. To use the *Identity* parameter, you specify the integer value that's assigned to the IP Allow list entry.

Use the Shell to add IP Allow list entries

To add IP Allow list entries, use the following syntax:

```
Add-IPAllowListEntry <-IPAddress IPAddress | -IPRange IP
range or CIDR IP> [-ExpirationTime <DateTime>] [-Comment
"<Descriptive Comment>"]
```

This example adds the IP Allow list entry for the IP address range 192.168.1.10 through 192.168.1.15 and configures the IP Allow list entry to expire on July 4, 2014 at 15:00.

```
Add-IPAllowListEntry -IPRange 192.168.1.10-192.168.1.15 -
ExpirationTime "7/4/2014 15:00"
```

How do you know this worked?

To verify that you successfully added an IP Allow list entry, run the following command and verify that the new IP Allow list entry is displayed.

```
Get-IPAllowListEntry
```

Use the Shell to remove IP Allow list entries

To remove IP Allow list entries, use the following syntax:

```
Remove-IPAllowListEntry <IdentityInteger>
```

The following example removes the IP Allow list entry that has the *Identity* value 3.

```
Remove-IPAllowListEntry 3
```

This example removes the IP Allow list entry that contains the IP address 192.168.1.12 without using the *Identity* integer value. Note that the IP Allow list entry can be an individual IP address or an IP address range.

```
Get-IPAllowListEntry -IPAddress 192.168.1.12 | Remove-IPAllowListEntry
```

How do you know this worked?

To verify that you successfully removed an IP Allow list entry, run the following command and verify that the IP Allow list entry you removed is gone.

```
Get-IPAllowListEntry
```

IP Allow List provider procedures

These procedures apply to IP Allow List providers. They don't apply to the IP Allow list.

Use the **IPAllowListProvidersConfig** cmdlets to view and configure how connection filtering uses all IP Allow List providers. Use the **IPAllowListProvider** cmdlets to view, configure, and test IP Allow List providers.

Use the Shell to view the configuration of all IP Allow List providers

To view how connection filtering uses all IP Allow List providers, run the following command:

```
Get-IPAllowListProvidersConfig | Format-List *Enabled
```

Use the Shell to enable or disable all IP Allow List providers

To disable all IP Allow List providers, run the following command:

```
Set-IPAllowListProvidersConfig -Enabled $false
```

To enable all IP Allow List providers, run the following command:

```
Set-IPAllowListProvidersConfig -Enabled $true
```

How do you know this worked?

To verify that you enabled or disabled all IP Allow List providers, run the following command and verify that the value displayed is the value you configured.

```
Get-IPAllowListProvidersConfig | Format-List Enabled
```

Use the Shell to configure all IP Allow List providers

To configure how connection filtering uses all IP Allow List providers, use the following syntax:

```
Set-IPAllowListProvidersConfig [-ExternalMailEnabled <$true
| $false>] [-InternalMailEnabled <$true | $false>]
```

This example configures all IP Allow List providers to filter incoming connections from internal and external mail servers. By default, connections are filtered from external mail servers only (*ExternalMailEnabled* is set to `$true`, and *InternalMailEnabled* is set to `$false`). Non-authenticated connections and authenticated connections from external partners are considered external.

```
Set-IPAllowListProvidersConfig -InternalMailEnabled $true
```

For more information, see Set-IPBlockListProvidersConfig.

How do you know this worked?

To verify that you successfully configured all IP Allow List providers, run the following command and verify that the values displayed are the values you configured.

```
Get-IPAllowListProvidersConfig | Format-List *MailEnabled
```

Use the Shell to view IP Allow List providers

To view the summary list of all the IP Allow List providers, run the following command.

```
Get-IPAllowListProvider
```

To view the details of a specific provider, use the following syntax.

```
Get-IPAllowListProvider <IPAllowListProviderIdentity>
```

This example show the details of the provider named Contoso IP Allow List Provider.

```
Get-IPAllowListProvider "Contoso IP Allow List Provider" |
Format-List Name,Enabled,Priority,LookupDomain,*Match
```

Use the Shell to add an IP Allow List provider

To add an IP Allow List provider, use the following syntax:

```
Add-IPAllowListProvider -Name "<Descriptive Name>" -
LookupDomain <FQDN> [-Priority <Integer>] [-Enabled <$true
| $false>] [-AnyMatch <$true | $false>] [-BitmaskMatch
<IPAddress>] [-IPAddressesMatch
<IPAddressStatusCode1,IPAddressStatusCode2...>]
```

This example creates an IP Allow List provider named "Contoso IP Allow List Provider" with the following options:

- **FQDN to use the provider** allow.contoso.com
- **Bitmask code to use from the provider** 127.0.0.1

```
Add-IPAllowListProvider -Name "Contoso IP Allow List
Provider" -LookupDomain allow.contoso.com -BitmaskMatch
127.0.0.1
```

Note:

When you add a new IP Allow List provider, it's enabled by default (the value of *Enabled* is `$true`), and the priority value is incremented (the first entry has the *Priority* value 1).

For more information, see Add-IPBlockListProvider.

How do you know this worked?

To verify that you successfully added an IP Allow List provider, run the following command and verify that the new IP Allow List provider is displayed.

```
Get-IPAllowListProvider
```

Use the Shell to enable or disable an IP Allow List provider

To enable or disable a specific IP Allow List provider, use the following syntax.

```
Set-IPAllowListProvider <IPAllowListProviderIdentity> -
Enabled <$true | $false>
```

This example disables the provider named Contoso IP Allow List Provider.

```
Set-IPAllowListProvider "Contoso IP Allow List Provider" -
Enabled $false
```

This example enables the provider named Contoso IP Allow List Provider.

```
Set-IPAllowListProvider "Contoso IP Allow List Provider" -
Enabled $true
```

How do you know this worked?

To verify that you successfully enabled or disabled an IP Allow List provider, run the following command and verify that the value displayed is the value you configured.

```
Get-IPAllowListProvider <IPAllowListProviderIdentity> |
Format-List Enabled
```

Use the Shell to configure an IP Allow List provider

The configuration options that are available on the **Set-IPAllowListProvider** cmdlet are identical to those on the **New-IPAllowListProvider** cmdlet.

To configure an existing IP Allow List provider, use the following syntax:

```
Set-IPAllowListProvider <IPAllowListProviderIdentity> -Name
"<Descriptive Name>" -LookupDomain <FQDN> [-Priority
<Integer>] [-AnyMatch <$true | $false>] [-BitmaskMatch
<IPAddress>] [-IPAddressesMatch
<IPAddressStatusCode1,IPAddressStatusCode2...>]
```

For example, to add the IP address status code 127.0.0.1 to the list of existing status codes for the provider named Contoso IP Allow List Provider, run the following command:

```
Set-IPAllowListProvider "Contoso IP Allow List Provider" -
IPAddressesMatch @{Add="127.0.0.1"}
```

For more information, see Set-IPBlockListProvider.

How do you know this worked?

To verify that you successfully configured an IP Allow List provider, run the following command and verify that the values displayed are the values you configured.

```
Get-IPAllowListProvider <IPAllowListProviderIdentity> |
Format-List
```

Use the Shell to test an IP Allow List provider

To test an IP Allow List provider, use the following syntax:

```
Test-IPAllowListProvider <IPAllowListProviderIdentity> -
IPAddress <IPAddressToTest>
```

The following example tests the provider named Contoso IP Allow List Provider by looking up the IP address 192.168.1.1.

```
Test-IPAllowListProvider "Contoso IP Allow List Provider" -
```

```
IPAddress 192.168.1.1
```

Use the Shell to remove an IP Allow List provider

To remove an IP Allow List provider, use the following syntax:

```
Remove-IPAllowListProvider <IPAllowListProviderIdentity>
```

This example removes the IP Allow List provider named Contoso IP Allow List Provider.

```
Remove-IPAllowListProvider "Contoso IP Allow List Provider"
```

How do you know this worked?

To verify that you successfully removed an IP Allow List provider, run the following command and verify that the IP Allow List provider you removed is gone.

```
Get-IPAllowListProvider
```

Recipient filtering on Edge Transport servers

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-07*

Recipient filtering is an anti-spam feature in Microsoft Exchange Server 2013 that relies on the RCPT TO SMTP header to determine what action, if any, to take on an inbound message. Recipient filtering is performed by the Recipient Filter agent.

The Recipient Filter agent blocks messages according to the characteristics of the intended recipient in the organization. The Recipient Filter agent can help you prevent the acceptance of messages in the following scenarios:

- **Nonexistent recipients** You can prevent delivery to recipients that aren't in the organization's address book. For example, you may want to stop delivery to frequently misused account names, such as administrator@contoso.com or support@contoso.com.
- **Restricted distribution lists** You can prevent delivery of Internet mail to distribution lists that should be used only by internal users.
- **Mailboxes that should never receive messages from the Internet** You can prevent delivery of Internet mail to a specific mailbox or alias that's typically used inside the organization, such as Helpdesk.

The Recipient Filter agent acts on recipients stored in one or both of the following data sources:

- **Recipient Block list** An administrator-defined list of recipients who should never receive messages from the Internet.
- **Recipient Lookup** Queries Active Directory to verify the recipient exists in the organization. On an Edge Transport server, Recipient Lookup requires access to Active Directory information provided by EdgeSync to the local instance of Active Directory Lightweight Directory Services (AD LDS).

When you enable the Recipient Filter agent, one of the following actions is taken on inbound messages according to the characteristics of the recipients. These recipients are indicated by the RCPT TO header.

- If the inbound message contains a recipient that is on the Recipient Block list, the Exchange server sends a `550 5.1.1 User unknown` SMTP session error to the sending server.
- If the inbound message contains a recipient that doesn't match any recipients in Recipient Lookup, the Exchange server sends a `550 5.1.1 User unknown` SMTP session error to the sending server.
- If the recipient isn't on the Recipient Block list and the recipient is in Recipient Lookup, the Exchange server sends a `250 2.1.5 Recipient OK` SMTP response to the sending server, and the next anti-spam agent in the chain processes the message.

Contents

Configuring recipient lookup

One of the most effective ways to reduce spam is to validate recipients before accepting inbound messages from the Internet. You enable the blocking of messages sent to recipients who don't exist in the Exchange organization, and the blocking of specific recipients using the **Set-RecipientFilterConfig** cmdlet in the Exchange Management Shell. For more information, see Manage recipient filtering on Edge Transport servers.

If you have an Edge Transport server installed in your perimeter network, it's a good idea to configure the AD LDS instance that runs on the Edge Transport server to synchronize with Active Directory. By default, AD LDS is installed and configured on the Edge Transport server. However, you must configure AD LDS to communicate with an Active Directory domain-joined global catalog server by subscribing the Edge Transport server to your organization. For more information, see Use an Exchange 2010 or 2007 Edge Transport server in Exchange 2013.

Return to top

Tarpitting functionality

Recipient Lookup functionality enables the sending server to determine whether an email address is valid or invalid. As mentioned earlier, when the recipient of an inbound message is a known recipient, the Exchange server sends back a `250 2.1.5 Recipient OK` SMTP response to the sending server. This functionality provides an ideal environment for a directory harvest attack.

A *directory harvest attack* is an attempt to collect valid email addresses from a particular organization so that the email addresses can be added to a spam database. Because all spam income relies on trying to make people open email messages, addresses known to be active are a commodity that malicious users, or *spammers*, pay for. Because the SMTP protocol provides feedback for known senders and unknown senders, a spammer can write an automated program that uses common names or dictionary terms to construct email addresses to a specific domain. The program collects all email addresses that return a `250 2.1.5 Recipient OK` SMTP response and discards all email addresses that return a `550 5.1.1 User unknown` SMTP session error. The spammer can then sell the valid email addresses or use them as recipients for unsolicited messages.

To combat directory harvest attacks, Exchange 2013 includes tarpitting functionality. *Tarpitting* is the practice of artificially delaying server responses for specific SMTP communication patterns that indicate high volumes of spam or other unwelcome messages. The intent of tarpitting is to slow down the communication process for such email traffic so that the cost of sending spam increases for the person or organization sending the spam. Tarpitting makes directory harvest attacks too costly to automate efficiently.

If tarpitting isn't configured, the Exchange server immediately returns a `550 5.1.1 User unknown` SMTP session error to the sender when a recipient isn't located in Recipient Lookup. Alternatively, if tarpitting is configured, SMTP waits a specified number of seconds before it returns the `550 5.1.1 User unknown` error. This pause in the SMTP session makes automating a directory harvest attack more difficult and less cost-effective for the spammer. By default, tarpitting is configured for 5 seconds on Receive connectors.

To configure the delay before SMTP returns the `550 5.1.1 User unknown` error, you set the tarpitting interval using the *TarpitInterval* parameter on the **Set-ReceiveConnector** cmdlet. The syntax is:

```
Set-ReceiveConnector <Receive Connector> -TarpitInterval
<00:00:00 to 00:10:00>
```

The default value is `00:00:05` or 5 seconds. The name of the default Receive connector on an Edge Transport server is `Default internal receive connector <server name>`.

Use caution if you decide to change the tarpitting interval. An overly long interval could disrupt ordinary mail flow, whereas an overly brief interval may not be as effective in thwarting a directory harvest attack. If you change the tarpitting interval, do so in small increments and verify the results. For example, if 5 seconds isn't effective, try changing the interval to 10 seconds.

Return to top

Multiple namespaces

The Recipient Filter agent performs recipient lookups only for authoritative domains. If your organization accepts and forwards messages on behalf of another domain that's configured as an internal relay or external relay domain, the Recipient Filter agent doesn't perform a recipient lookup on recipients in those domains. However, if the recipient is specified in the Recipient Block list, the recipient will still be blocked by the Recipient Filter agent.

Note that you can also configure accepted domains locally on an Edge Transport server. If the domain is configured as internal relay or external relay domain, the Recipient Filter agent on the Edge Transport server also doesn't perform a recipient lookup on recipients in those domains.

Return to top

Manage recipient filtering on Edge Transport servers

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-23*

Recipient filtering is provided by the Recipient Filter agent. When recipient filtering is enabled on an Exchange server, it filters inbound messages that come from the Internet but aren't authenticated. These messages are handled as external messages.

Note:

Although the Recipient Filter agent is available on Mailbox servers, you shouldn't configure it. When recipient filtering on a Mailbox server detects one invalid or blocked recipient in a message that contains other valid recipients, the message is rejected. If you install the anti-spam agents on a Mailbox server, the Recipient Filter agent is enabled by default. However, it isn't configured to block any recipients. For more information, see Enable anti-spam functionality on Mailbox servers.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-

malware permissions topic.

- You can only use the Shell to perform this procedure.
- By default, anti-spam features aren't enabled in the Transport service on a Mailbox server. Typically, you only enable the anti-spam features on a Mailbox server if your Exchange organization doesn't do any prior anti-spam filtering before accepting incoming messages. For more information, see Enable anti-spam functionality on Mailbox servers.
- The *AddressBookEnabled* parameter on the **Set-AcceptedDomain** cmdlet enables or disables recipient filtering for recipients in an accepted domain. By default, recipient filtering is enabled for authoritative domains, and disabled for internal relay domains and external relay domains. To view the status of the *AddressBookEnabled* parameter for the accepted domains in your organization, run the following command:

```
Get-AcceptedDomain | Format-List Name,AddressBookEnabled
```

- If you disable recipient filtering using the procedure in this topic, recipient filtering functionality will be disabled, but the underlying Recipient Filter agent will remain enabled.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable recipient filtering

To disable recipient filtering, run the following command:

```
Set-RecipientFilterConfig -Enabled $false
```

To enable recipient filtering, run the following command:

```
Set-RecipientFilterConfig -Enabled $true
```

Note:

When you disable recipient filtering, the underlying Recipient Filter agent is still enabled. To disable the Recipient Filter agent, run the command: `Disable-TransportAgent "Recipient Filter Agent"`.

How do you know this worked?

To verify that you have successfully enabled or disabled recipient filtering, do the following:

1. Run the following command:

```
Get-RecipientFilterConfig | Format-List Enabled
```

2. Verify the value displayed is the value you configured.

Use the Shell to enable or disable the Recipient Block list

Run the following command:

```
Set-RecipientFilterConfig -BlockListEnabled <$true |
$false>
```

This example enables the Recipient Block list:

```
Set-RecipientFilterConfig -BlockListEnabled $true
```

How do you know this worked?

To verify that you have successfully enabled or disabled the Recipient Block list, do the following:
1. Run the following command:

```
Get-RecipientFilterConfig | Format-List BlockListEnabled
```

2. Verify the value displayed is the value you configured.

Use the Shell to configure the Recipient Block list

To replace the existing values, run the following command:

```
Set-RecipientFilterConfig -BlockedRecipients
<recipient1,recipient2...>
```

This example configures the Recipient Block list with the valuesmark@contoso.com and kim@contoso.com:

```
Set-RecipientFilterConfig -BlockedRecipients
mark@contoso.com,kim@contoso.com
```

To add or remove entries without modifying any existing values, run the following command:

```
Set-RecipientFilterConfig -BlockedRecipients
@{Add="<recipient1>","<recipient2>"...;
Remove="<recipient1>","<recipient2>"...}
```

This example adds chris@contoso.com to the list of recipients, and removes michelle@contoso.com from the list of recipients in the Recipient Block list:

```
Set-RecipientFilterConfig -BlockedRecipients
@{Add="chris@contoso.com"; Remove="michelle@contoso.com"}
```

How do you know this worked?

To verify that you have successfully configured the Recipient Block list, do the following:
1. Run the following command:

```
Get-RecipientFilterConfig | Format-List BlockedRecipients
```

2. Verify the values displayed are the values you configured.

Use the Shell to enable or disable Recipient Lookup

Run the following command:

```
Set-RecipientFilterConfig -RecipientValidationEnabled
<$true | $false>
```

To block messages to recipients that don't exist in your organization, run the following command:

```
Set-RecipientFilterConfig -RecipientValidationEnabled $true
```

How do you know this worked?

To verify that you have successfully enabled or disabled Recipient Lookup, do the following:
1. Run the following command:

```
Get-RecipientFilterConfig | Format-List
RecipientValidationEnabled
```

2. Verify the value displayed is the value you configured.

Attachment filtering on Edge Transport servers

Exchange Server 2013 > Anti-spam and anti-malware protection > Anti-spam protection >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-10*

In Microsoft Exchange Server 2013, you can use attachment filtering on Edge Transport servers to control the attachments that users receive in email messages. Attachment filtering is performed by

the Attachment Filtering agent, which is available only on Edge Transport servers.

Types of attachment filtering

You can use the following types of attachment filtering to control attachments that enter or leave your organization through an Edge Transport server:

- **Filtering based on file name or file name extension** You specify the exact file name or file name extension that you want to filter. An example of a file name filter is `BadFileName.exe`. An example of a file name extension filter is `*.exe`.
- **Filtering based on file MIME content type** You specify the MIME content type value that you want to filter. The MIME content type value indicates what the attachment is—for example, a JPEG image, an executable file, or a Microsoft Excel file. Content types are expressed as `type/subtype`. For example, a JPEG image file is expressed as `image/jpeg`.

To view a complete list of file name extensions and content types that attachment filtering can detect, run the following command on the Edge Transport server:

```
Get-AttachmentFilterEntry | Format-List
```

After you define the files to look for, you can configure the action to take on messages that contain these attachments. You can't specify different actions for different types of attachments. You configure one of the following actions for all the messages that match any of the attachment filters:

- **Reject (block) the message** The message is blocked. The sender receives a non-delivery report (NDR) that explains that the message wasn't delivered because it contained an unacceptable attachment. You can customize the text in the NDR. The default text is: `Message rejected due to unacceptable attachments`.
- **Strip the attachment but allow the message through** The attachment is removed from the message. However, the message itself and any other attachments that don't match the filter are allowed through. If an attachment is stripped, it's replaced with a text file that explains why the attachment was removed. This is the default action.
- **Silently delete the message** The message is deleted. Neither the sender nor the recipient receives notification.

For more information, see Manage attachment filtering on Edge Transport servers.

Note:

You can't retrieve messages that have been blocked or attachments that have been stripped. When you configure attachment filters, carefully examine all possible file name matches and verify that legitimate attachments won't be affected by the filter.
Also, don't remove attachments from digitally signed, encrypted, or rights-protected email messages. If you remove attachments from such messages, you invalidate the digitally signed messages and make encrypted and rights-protected messages unreadable.

Manage attachment filtering on Edge Transport servers

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-10*

Attachment filtering is provided by the Attachment Filter agent that's available only on Edge Transport servers. Attachment filtering can help prevent files that are attached in email messages from entering your organization. You can configure one or more attachment filter entries to filter attachments either by content type or by file name.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions and the "Transport agents" entry in the Mail flow permissions topic.
- Configuration changes that you make to attachment filtering on an Edge Transport server are made only to the local computer. If you have multiple Edge Transport servers in your perimeter network, you need to configure attachment filtering on each Edge Transport server separately.
- You can only use the Shell to perform this procedure.
- When you disable attachment filtering and restart the Microsoft Exchange Transport service, all attachment filtering features stop working.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable attachment filtering

When you enable or disable the Attachment Filtering agent, the change takes effect after you restart the Microsoft Exchange Transport service. When you restart the Microsoft Exchange Transport service on an Edge Transport server, mail flow on the server is temporarily interrupted.

To disable attachment filtering, run the following command:

```
Disable-TransportAgent "Attachment Filtering Agent"
```

To enable attachment filtering, run the following command:

```
Enable-TransportAgent "Attachment Filtering Agent"
```

After you enable or disable attachment filtering, restart the Microsoft Exchange Transport service by running the following command:

```
Restart-Service MSExchangeTransport
```

How do you know this worked?

To verify that you successfully enabled or disabled attachment filtering, do the following:
1. Run the following command:

```
Get-TransportAgent "Attachment Filtering Agent"
```

2. If the value of **Enabled** is `True`, attachment filtering is enabled. If the value is `False`, attachment filtering is disabled.

Use the Shell to view attachment filtering entries

Attachment filtering entries define the message attachments that you want to keep out of your organization. To view the attachment filtering entries that are used by the Attachment Filtering agent, run the following command:

```
Get-AttachmentFilterEntry | Format-Table
```

To view a specific MIME content type entry, use the following syntax:

```
Get-AttachmentFilteringEntry ContentType:<MIMEContentType>
```

For example, to view the content type entry for JPEG images, run the following command:

```
Get-AttachmentFilteringEntry ContentType:image/jpeg
```

To view a specific file name or file name extension entry, use the following syntax:

```
Get-AttachmentFilteringEntry FileName:<FileName or
FileNameExtension>
```

For example, to view the file name extension entry for JPEG attachments, run the following command:

```
Get-AttachmentFilteringEntry FileName:*.jpg
```

Use the Shell to add attachment filtering entries

To add an attachment filtering entry that filters attachments by MIME content type, use the following syntax:

```
Add-AttachmentFilterEntry -Name <MIMEContentType> -Type
ContentType
```

The following example adds a MIME content type entry that filters JPEG images.

```
Add-AttachmentFilterEntry -Name image/jpeg -Type
ContentType
```

To add an attachment filtering entry that filters attachments by file name or file name extension, use the following syntax:

```
Add-AttachmentFilterEntry -Name <FileName or
FileNameExtension> -Type FileName
```

The following example filters attachments that have the .jpg file name extension.

```
Add-AttachmentFilterEntry -Name *.jpg -Type FileName
```

How do you know this worked?

To verify that you successfully added an attachment filtering entry, do the following:
1. Run the following command to verify that the filtering entry exists.

```
Get-AttachmentFilterEntry | Format-Table
```

2. Send a test message that contains a prohibited attachment from an external mailbox to an internal recipient and verify that the message is rejected, stripped, or deleted.

Use the Shell to remove attachment filtering entries

To remove an attachment filtering entry that filters attachments by MIME content type, use the following syntax:

```
Remove-AttachmentFilterEntry ContentType:<ContentType>
```

The following example removes the MIME content type entry for JPEG images.

```
Remove-AttachmentFilterEntry ContentType:image/jpeg
```

To remove an attachment filtering entry that filters attachments by file name or file name extension, use the following syntax:

```
Remove-AttachmentFilterEntry FileName:<FileName or FileNameExtension>
```

The following example removes the file name entry for the .jpg file name extension.

```
Remove-AttachmentFilterEntry FileName:*.jpg
```

How do you know this worked?

To verify that you successfully removed an attachment filtering entry, do the following:

1. Run the following command to verify that the filtering entry was removed.

```
Get-AttachmentFilterEntry | Format-Table
```

2. Send a test message that contains an allowed attachment from an external mailbox to an internal recipient and verify that the message was successfully delivered with the attachment.

Use the Shell to view the attachment filtering action

To view the attachment filtering action that's used when a prohibited attachment is detected in a message, run the following command:

```
Get-AttachmentFilterListConfig
```

Use the Shell to configure the attachment filtering action

To configure the attachment filtering action that will be used when a prohibited attachment is detected in a message, use the following syntax:

```
Set-AttachmentFilterListConfig [-Action <Reject | Strip | SilentDelete>] [-RejectResponse "<Message text>"] [-AdminMessage "<Replacement file text>"] [-ExceptionConnectors <ConnectorGUID>]
```

This example makes the following changes to the attachment filtering configuration:

- Reject (block) messages that have prohibited attachments.
- Use a custom response for rejected messages.

```
Set-AttachmentFilterListConfig -Action Reject -RejectResponse "This message contains a prohibited attachment. Your message can't be delivered. Please resend
```

```
the message without the attachment."
```

For more information, see Set-AttachmentFilterListConfig.

How do you know this worked?

To verify that you successfully configured the attachment filtering action, send a test message that contains a prohibited attachment from an external mailbox to an internal recipient and verify that the message and the attachment are processed as you expect.

Spam quarantine

Exchange Server 2013 > Anti-spam and anti-malware protection > Anti-spam protection >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-12*

Many organizations are bound by legal or regulatory requirements to preserve or deliver all legitimate email messages. In Microsoft Exchange Server 2013, spam quarantine is a feature of the Content Filter agent that reduces the risk of losing legitimate messages. Spam quarantine provides a temporary storage location for messages identified as spam that shouldn't be delivered to a user mailbox inside the organization.

Messages identified by the Content Filter agent as spam are wrapped in a non-delivery report (NDR) and delivered to a spam quarantine mailbox inside the organization. You can manage messages delivered to the spam quarantine mailbox and take appropriate actions. For example, you can delete messages or let messages flagged as false positives in anti-spam filtering be routed to their intended recipients. In addition, you can configure the spam quarantine mailbox to automatically delete messages after a designated time period.

Contents

Spam confidence level

When an external user sends email messages to a server running Exchange that runs the anti-spam features, the anti-spam features cumulatively evaluate characteristics of the messages and act as follows:

- Those messages suspected to be spam are filtered out.
- A rating is assigned to messages based on the probability that a message is spam. This rating is stored with the message as a message property called the spam confidence level (SCL) rating.

Spam quarantine uses the SCL rating to determine whether mail has a high probability of being spam. The SCL rating is a numeric value from 0 through 9, where 0 is considered less likely to be spam, and 9 is considered most likely to be spam.

You can configure mail that has a certain SCL rating to be deleted, rejected, or quarantined. The rating that triggers any of these actions is referred to as the *SCL quarantine threshold*. Within content filtering, you can configure the Content Filter agent to base its actions on the SCL quarantine threshold. For example, you can set the following conditions:

- SCL delete threshold is set to 8.
- SCL reject threshold is set to 7.
- SCL quarantine threshold is set to 6.
- SCL Junk Email folder threshold is set to 5.

Based on the preceding SCL thresholds, all email with an SCL of 6 will be delivered to the spam quarantine mailbox.

For more information, see Manage content filtering.

Return to top

Spam quarantine

When messages are received by the Exchange server that's running all default anti-spam agents, the content filter is applied as follows:

- If the SCL rating is greater than or equal to the SCL quarantine threshold but less than either the SCL delete threshold or SCL reject threshold, the message goes to the spam quarantine mailbox.
- If the SCL rating is lower than the spam quarantine threshold, it's delivered to the recipient's Inbox.

The message administrator uses Microsoft Outlook to monitor the spam quarantine mailbox for false positives. If a false positive is found, the administrator can send the message to the recipient's mailbox.

The message administrator can review the anti-spam stamps if either of the following conditions is true:

- Too many false positives are filtered into the spam quarantine mailbox.
- Not enough spam is being rejected or deleted.

For more information, see Anti-spam stamps.

You can then adjust the SCL settings to more accurately filter the spam coming into the organization. For more information, see Spam Confidence Level Threshold.

To use spam quarantine, you need to follow these steps:

1. Verify content filtering is enabled.
2. Create a dedicated mailbox for spam quarantine.
3. Specify the spam quarantine mailbox.
4. Configure the SCL quarantine threshold.

5. Manage the spam quarantine mailbox.
6. Adjust the SCL quarantine threshold as needed.

For detailed instructions, see Configure a spam quarantine mailbox.

Return to top

Configure a spam quarantine mailbox

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-19*

Messages determined to be spam by the Content Filter agent can be directed to a spam quarantine mailbox. If the spam confidence level (SCL) quarantine threshold is enabled, all messages that are quarantined are wrapped as non-delivery reports (NDR) and are sent to the SMTP address that you specify as the spam quarantine mailbox. You can review quarantined messages and release them to their intended recipients by using the Send Again feature in Microsoft Outlook.

What do you need to know before you begin?

- Estimated time to complete this task: 45 minutes.
- By default, anti-spam features aren't enabled in the Transport service on a Mailbox server. Typically, you only enable the anti-spam features on a Mailbox server if your Exchange organization doesn't do any prior anti-spam filtering before accepting incoming messages. For more information, see Enable anti-spam functionality on Mailbox servers.
- The person responsible for the spam quarantine mailbox can view potentially private and sensitive messages, and then send mail on behalf of anybody in the Exchange organization.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Verify content filtering is enabled

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

1. Run the following command to verify the Content Filter agent is installed and enabled on the Exchange server:

```
Get-TransportAgent "Content Filter Agent"
```

2. Run the following command to verify content filtering is enabled:

```
Get-ContentFilterConfig | Format-List Enabled
```

For more information, see Manage content filtering.

Step 2: Create a dedicated mailbox for spam quarantine

To create a dedicated spam quarantine mailbox, follow these steps:

- **Create a dedicated Exchange database** We recommend that you create a dedicated database for the spam quarantine mailbox. The spam quarantine mailbox should have a large database, because if the storage quota limit is reached, messages will be lost. For more information, see Manage mailbox databases in Exchange 2013.
- **Create a dedicated mailbox and user account** We recommend that you create a dedicated mailbox and Active Directory user account for the spam quarantine mailbox. For more information, see Create user mailboxes.

You may apply recipient policies, such as messaging records management, mailbox quotas, and delegation rights, according to your organization's compliance policies and needs. For more information, see Messaging records management.

Note:

If a quarantined message is rejected because of a storage quota, the message will be lost. Exchange doesn't generate NDRs for quarantined messages because the quarantined messages are wrapped as NDRs.

- **Configure Outlook** You need to configure the Outlook delegate access permissions to meet the needs of your organization. In addition, we recommend that you configure the Outlook profile to show the original `Sender[#0x0069001E]`, `Recipient[#0x0E04001E]`, and `Bcc[#0x0E02001E]` fields in the **Message** view. For more information, see Release quarantined messages from the spam quarantine mailbox.

Step 3: Specify the spam quarantine mailbox

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Run the following command:

```
Set-ContentFilterConfig -QuarantineMailbox <SmtpAddress>
```

This example sends all messages that exceed the spam quarantine threshold to spamQ@contoso.com.

```
Set-ContentFilterConfig -QuarantineMailbox
spamQ@contoso.com
```

How do you know this step worked?

To verify that you have successfully specified the spam quarantine mailbox, do the following:
1. Run the following command:

```
Get-ContentFilterConfig | Format-List QuarantineMailbox
```

2. Verify the value displayed is the value you configured.

Step 4: Configure the SCL quarantine threshold

The SCL quarantine threshold is the value at which a particular message identified as potential spam is delivered to the spam quarantine mailbox. You can set the SCL quarantine threshold to a value from 0 through 9, where 0 is considered less likely to be spam, and 9 is considered most likely to be spam.

For more information about how to adjust SCL thresholds to suit your organization's requirements and how to adjust per-recipient SCL thresholds, see Manage content filtering.

Step 5: Manage the spam quarantine mailbox

When you manage your spam quarantine mailbox, follow these guidelines:

- Release items that have been sent to the spam quarantine mailbox by using the Send Again feature in Outlook to resend the original message.

For more information, see Release quarantined messages from the spam quarantine mailbox.

- Monitor the spam quarantine mailbox so that the size of the spam quarantine mailbox remains in an acceptable range. The volume of email messages can change because of a larger set of recipients, the natural trend of larger messages, or the threshold on the SCL quarantine action.
- Monitor the spam quarantine mailbox for false positives. If your spam quarantine mailbox includes many false positives, adjust your SCL quarantine threshold. For more information about how to determine why false positives are being delivered to the spam quarantine mailbox, see Anti-spam stamps.
- Use the same Outlook profile to recover quarantined messages from the spam quarantine mailbox. Applying permissions to a different Outlook profile to recover messages isn't supported. You can't use a different Outlook profile to recover or release messages from the spam quarantine mailbox.

Important:

NDRs identified as spam are deleted, even if their SCL rating indicates that they should be quarantined. NDRs aren't delivered to the spam quarantine mailbox. To track such messages, use the agent log or the message tracking log. For more information, see Anti-spam agent logging.

Step 6: Adjust the SCL quarantine threshold

After you configure the SCL quarantine threshold, periodically monitor the settings and adjust them based on your organization's needs. For example, if too many false positives are filtered into the spam quarantine mailbox, raise the SCL quarantine threshold to a larger number. For more information about how to adjust the SCL quarantine threshold, see Spam Confidence Level Threshold.

Configure Outlook to show the original sender in the quarantine mailbox

Anti-spam and anti-malware protection > Anti-spam protection > Spam quarantine >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

Spam quarantine is a feature of the Content Filter agent that reduces the risk of losing legitimate messages. Spam quarantine provides a temporary storage location for messages that are identified as spam and that shouldn't be delivered to a user mailbox inside the organization.

When a message meets the spam quarantine threshold, it's wrapped in a non-delivery report (NDR) and delivered to the spam quarantine mailbox. Because the quarantined messages are stored as NDRs in the quarantine mailbox, the postmaster address of your organization will be listed as the From: address for all messages. However, having the original sender address, the original recipient address, and the original spam confidence level (SCL) in the field list would make it easier to locate the message you want to recover.

By default, you can't select these fields in Microsoft Outlook. Before you can add these fields in the message view, you must first create an Outlook form that adds the original sender, original recipient, and original SCL as optional fields you can select. After you create this custom form, you can configure Outlook to display these fields in the message view.

What do you need to know before you begin?

- Estimated time to complete this procedure: 15 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "Mailbox access" entry in the Mail flow permissions topic.

- This procedure requires that you've configured the anti-spam quarantine mailbox. For more information, see Configure a spam quarantine mailbox.
- To access the quarantine mailbox, you need to configure an Outlook profile for that mailbox and then open the mailbox using Outlook. For more information about configuring and using multiple Outlook profiles, see Overview of Outlook e-mail profiles.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Step 1: Use Notepad to create a custom Outlook form

1. Open Notepad, and copy the following code into the document.

```
[Description]
MessageClass=IPM.Note
CLSID={00020D31-0000-0000-C000-000000000046}
DisplayName=Quarantine Extension Form
Category=Standard
Subcategory=Form
Comment=This form allows the Original Sender
(ReceivedRepresentingEmailAddress), Original Recipient
(To), and Original SCL (OriginalScl) values to be viewed as
columns.
LargeIcon=IPML.ico
SmallIcon=IPMS.ico
Version=3.0
Locale=enu
Hidden=1
Owner=Microsoft Corporation
Contact=Your Name
[Platforms]
Platform1=Win16
Platform2=NTx86
Platform9=Win95
[Platform.Win16]
```

```
CPU=ix86
OSVersion=Win3.1
[Platform.NTx86]
CPU=ix86
OSVersion=WinNT3.5
[Platform.Win95]
CPU=ix86
OSVersion=Win95
[Properties]
Property01=ReceivedRepresentingEmailAddress
Property02=DisplayTo
Property03=OriginalScl
[Property.ReceivedRepresentingEmailAddress]
Type=31
NmidInteger=0x0078
DisplayName=ReceivedRepresentingEmailAddress
[Property.DisplayTo]
Type=31
NmidInteger=0x0E04
DisplayName=DisplayTo
[Property.OriginalScl]
Type=3
NmidPropset={41F28F13-83F4-4114-A584-EEDB5A6B0BFF}
NmidString=OriginalScl
DisplayName=OriginalScl
[Verbs]
Verb1=1
[Verb.1]
DisplayName=&Open
Code=0
Flags=0
Attribs=2
[Extensions]
Extensions1=1
[Extension.1]
Type=31
NmidPropset={00020D0C-0000-0000-C000-000000000046}
NmidInteger=1
Value=1000000000000000
```

2. Save the file in your Office Forms folder using the following values:
   - **Path** <*Office Install Path*>\<*OfficeVersion*>\Forms\<*LCID*>
     - <*Office Install Path*> For 32-bit versions of Office on 32-bit versions of Microsoft Windows, or 64-bit versions of Office on 64-bit versions of Windows, the default path is `C:\Program Files\Microsoft Office`. For 32-bit versions of Office on 64-bit versions of Windows, the default path is `C:\Program Files (x86)\Microsoft Office`.
     - <*OfficeVersion*> For Outlook 2007, the value is `office12`. For Outlook 2010, the value is `office14`. For Outlook 2013, the value is `office15`.
     - <*LCID*> This is your locale ID (LCID) value. For example, the LCID for US English is 1033. For more information, see KB221435: List of supported locale identifiers in Word.
   - **Name** For the rest of this procedure, assume the file is named `QTNE.cfg`. The name of the file isn't important, but be sure to enclose the value in quotation marks so the file is saved as QTNE.cfg and not QTNE.cfg.txt.

For example, for a 32-bit US English version of Outlook 2013 installed on a 64-bit version of Windows, save the file as:

```
"C:\Program Files (x86)\Microsoft Office\Office15\Forms
\1033\QTNE.cfg"
```

Step 2: Configure Outlook to use the custom Outlook form

Use one of the following procedures based on the version of Outlook that's installed on your computer.

Configure Outlook 2010 or Outlook 2013

1. In Outlook 2010 or Outlook 2013, click **File** > **Options** > **Advanced**.
2. In the **Developers** section, click **Custom Forms**.
3. In the **Options** dialog box, click **Manage Forms**.
4. In the **Forms Manager** dialog box, click **Install**. Browse to the location of the `QTNE.cfg` file, select it, click **Open**, and then click **OK** to install the Quarantine Extension Form in your Personal Forms library.
5. In the **Forms Manager** dialog box, click **Close**. Click **OK** twice to close the remaining dialog boxes and return to the main Outlook interface.
6. On the **Home** tab in the **Mail** view of the Inbox, right-click the column heading row (you may need to expand the width of the message list to see the columns), and then select **View Settings**.
7. In the **Advanced View Settings** dialog box, click **Columns**.
8. In the **Show Columns** dialog box, in the **Select available columns from** drop-down list, scroll to the end of the list and select **Forms**.
9. In the **Select Enterprise forms for this folder** dialog box, in the **Selected Forms** field, select **Message** and click **Remove**. In the **Personal Forms** field, select **Quarantine Extension Form**, and then click **Add**. When you are finished, click **Close**.
10. In the **Show Columns** dialog box, in the **Available Columns** section, select one or more of the

following fields and click **Add** after each field you select.

- **ReceivedRepresentingEmailAddress** Original sender
- **To** Original recipient
- **OriginalScl** Original SCL

Use the **Move Up** or **Move Down** buttons to position the columns in the view. When you are finished, click **OK** twice to return to the main Outlook interface.

Configure Outlook 2007

1. In Outlook 2007, click **Tools** > **Options**.
2. In the **Options** dialog box, click the **Other** tab, and then under **General**, click **Advanced Options**.
3. In the **Advanced Options** dialog box, click **Custom Forms**, and then in the **Custom Forms** dialog box, click **Manage Forms**.
4. In the **Forms Manager** dialog box, click **Install**. Browse to the location of the `QTNE.cfg` file, select it, click **Open**, and then click **OK** to install the Quarantine Extension Form in your Personal Forms library.
5. Close the **Forms Manager** dialog box, and then click **OK** to close the remaining dialog boxes and return to the main Outlook 2007 interface.
6. In the default message view of Outlook 2007, in the Inbox, right-click the column heading row, and then select **Field Chooser**.

Note:
If you don't see Field Chooser as an option, you may need to expand the width of your message list.

7. In the **Field Chooser** drop-down menu, click **Forms**. You may have to scroll to find **Forms**.
8. In the **Select Enterprise forms for this Column** dialog box, from the drop-down menu, select **Personal Forms**, expand the **Standard** form, and then select **Quarantine Extension Form**. Click **Add**, and then click **Close**.

Note:
In some cases, you may have to remove the default **Message** form to add the **Quarantine Extension Form**.

9. In the **Field Chooser** dialog box, drag one or more of the following properties into the column heading row.
   - **ReceivedRepresentingEmailAddress** Original sender
   - **To** Original recipient
   - **OriginalScl** Original SCL

How do you know this worked?

You know this procedure worked if you can see the original sender, original recipient, or original SCL values for quarantined messages in the spam quarantine mailbox using Outlook.

Release quarantined messages from the spam quarantine mailbox

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

You can use Microsoft Outlook to recover a quarantined message from the spam quarantine mailbox.

Spam quarantine is a feature of the Content Filter agent that reduces the risk of losing legitimate messages. When a message meets the spam quarantine threshold, it's wrapped in a non-delivery report (NDR) and delivered to the spam quarantine mailbox. For more information about the spam quarantine feature, see Spam quarantine.

What do you need to know before you begin?

- Estimated time to complete each procedure: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox access" entry in the Mail flow permissions topic.
- This procedure requires that you've configured the anti-spam quarantine mailbox. For more information, see Configure a spam quarantine mailbox.
- To access the quarantine mailbox, you need to configure an Outlook profile for that mailbox and then open the mailbox using Outlook. For more information about configuring and using multiple Outlook profiles, see Overview of Outlook e-mail profiles.
- To make it easier to locate the message you want to recover, you can create a custom Outlook form and modify the default view to expose the original sender address in the list of messages. For detailed steps, see Configure Outlook to show the original sender in the quarantine mailbox.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use Outlook 2010 or Outlook 2013 to release a message

from the spam quarantine mailbox

1. Open the quarantine mailbox using Outlook 2010 or Outlook 2013 on a client computer.
2. In the **Mail** view, find the message you want to recover in the **Inbox**, and then double-click the message to open it.
3. In the **Move** section of the Ribbon, click **Actions** > **Resend this Message**.
4. When the message opens, click **Send** to resend the message to the intended recipient.

Use Outlook 2007 to release a message from the spam quarantine mailbox

1. Open the quarantine mailbox using Outlook 2007 on a client computer.
2. In the **Mail Folders** view, find the message you want to recover in the **Inbox**, and then double-click the message to open it.
3. On the **Report** tab, in the **Respond** group, click **Send Again**.
4. When the message opens, click **Send** to resend the message to the intended recipient.

How do you know this worked?

To verify that you have successfully released the message from the spam quarantine mailbox, contact the recipient and verify they received the message.

Anti-spam stamps

Exchange Server 2013 > Anti-spam and anti-malware protection > Anti-spam protection >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

Anti-spam stamps help you diagnose spam-related problems by applying diagnostic metadata, or stamps, such as sender-specific information, puzzle validation results, and content filtering results, to messages as they pass through the anti-spam features that filter inbound messages from the Internet. There are three anti-spam stamps: the phishing confidence level stamp, the spam confidence level stamp, and the Sender ID stamp.

You can use anti-spam stamps as diagnostic tools to determine what actions to take on false-positives and on suspected spam messages that individuals receive in their mailboxes.

Viewing anti-spam stamps

You can view anti-spam stamps by using Microsoft Outlook. For more information, see View anti-spam stamps in Outlook.

Understanding the anti-spam report

The anti-spam report is a summary report of the anti-spam filter results that have been applied to an email message. The Content Filter agent applies this stamp to the message envelope in the form of an X-header as follows.

```
X-MS-Exchange-Organization-Antispam-Report:
DV:<DATVersion>;CW:CustomList;PCL:PhishingVerdict
<verdict>;P100:PhishingBlock;PP:Presolve;SID:SenderIDStatus
<status>;TIME:<SendReceiveDelta>;MIME:MimeCompliance
```

The following table describes the filter information that can appear in an anti-spam report.

Note:

The anti-spam report only displays information from the filters that were applied to the specific message. An anti-spam report doesn't usually contain all the information listed in the following table. For example, you may receive the following anti-spam report: `DV:3.1.3924.1409;SID:SenderIDStatus Fail;PCL:PhishingLevel SUSPICIOUS;CW:CustomList;PP:Presolved;TIME:TimeBasedFeatures`.

Filter information in an anti-spam report

Stamp	Description
SID	The Sender ID (SID) stamp is based on the sender policy framework (SPF) that authorizes the use of domains in email. The SPF is displayed in the message envelope as `Received-SPF`. The Sender ID evaluation process generates a Sender ID status for the message. This status can be returned as one of the following values: • **Pass** Both the IP address and Purported Responsible Address (PRA) passed the Sender ID verification check. • **Neutral** Published Sender ID data is explicitly inconclusive. • **Soft fail** The IP address for the PRA may be in the not permitted set. • **Fail** The IP Address is not permitted; no PRA is found in the incoming mail or the sending domain does not exist. • **None** No published SPF data exists in the sender's DNS. • **TempError** A temporary DNS failure occurred, such as an unavailable DNS server. • **PermError** The DNS record is invalid, such as an error in the record format.

	The Sender ID stamp is displayed as an X-Header in the message envelope as follows: `X-MS-Exchange-Organization-SenderIdResult:<status>` For more information about Sender ID, see Sender ID.
DV	The DAT version (DV) stamp indicates the version of the spam definition file that was used when scanning the message.
SA	The signature action (SA) stamp indicates that the message was either recovered or deleted because of a signature that was found in the message.
SV	The signature DAT version (SV) stamp indicates the version of the signature file that was used when scanning the message.
PCL	The phishing confidence level (PCL) stamp displays the rating of the message based on its content and is applied when the message is processed by the Content Filter agent. This status can be returned as one of the following values: • **Neutral** The message's content isn't likely to be phishing. • **Suspicious** The message's content is likely to be phishing. The PCL value can range from 1 through 8. A PCL rating from 1 through 3 returns a status of `Neutral`. This means that the message's content isn't likely to be phishing. A PCL rating from 4 through 8 returns a status of `Suspicious`. This means that the message is likely to be phishing. The values are used to determine what action Outlook takes on messages. Outlook uses the PCL stamp to block the content of suspicious messages. The PCL stamp is displayed as an X-header in the message envelope as follows: `X-MS-Exchange-Organization-PCL:<status>`
SCL	The spam confidence level (SCL) stamp of the message displays the rating of the message based on its content. The Content Filter agent uses Microsoft SmartScreen technology to assess the contents of a message and to assign an SCL rating to each message. The SCL value is from 0 through 9, where 0 is considered less likely to be spam, and 9 is considered more likely to be spam. The actions that Exchange and Outlook take depend on your SCL threshold settings.

	The SCL stamp is displayed as an X-header in the message envelope as follows: `X-MS-Exchange-Organization-SCL:<status>` For more information about SCL thresholds and actions, see Spam Confidence Level Threshold.
CW	The custom weight (CW) stamp of a message indicates that the message contains an unapproved word or phrase and that the SCL value, or weight, of that unapproved word or phrase was applied to the final SCL score: • Unapproved phrases, or Block phrases, have maximum weight and change the SCL score to 9. • Approved words or phrases, or Allow phrases, have minimum weight and change the SCL score to 0. For more information about how to add approved and unapproved words or phrases to the Content Filtering agent, see Manage content filtering.
PP	The presolved puzzle (PP) stamp indicates that if a sender's message contains a valid, solved computational postmark, based on Outlook E-mail Postmark validation functionality, it's unlikely that the sender is a malicious sender. In this case, the Content Filter agent would reduce the SCL rating. The Content Filter agent doesn't change the SCL rating if the E-mail Postmark validation feature is enabled and either of the following conditions is true: • An inbound message doesn't contain a computational postmark header. • The computational postmark header isn't valid. For more information about the postmark validation feature, see Content filtering.
TIME:TimeBasedFeatures	The TIME stamp indicates that there was a significant time delay between the time that the message was sent and the time that the message was received. The TIME stamp is used to determine the final SCL rating for the message.
MIME:MIMECompliance	The MIME stamp indicates that the email message isn't MIME compliant.
P100:Phish	The P100 stamp indicates that the message contains a URL that's present in a

ingBlock	phishing definition file.
IPOnAllowList	The IPOnAllowList stamp indicates that the sender's IP address is on the IP Allow list. For more information about the IP Allow list, see Understanding Connection Filtering.
MessageSecurityAntispamBypass	The MessageSecurityAntispamBypass stamp indicates that the message wasn't filtered for content and that the sender has been granted permission to bypass the anti-spam filters.
SenderBypassed	The SenderBypassed stamp indicates that the Content Filter agent doesn't process any content filtering for messages that are received from this sender. For more information, see Manage content filtering.
AllRecipientsBypassed	The AllRecipientsBypassed stamp indicates that one of the following conditions was met for all recipients listed in the message: • The *AntispamBypassedEnabled* parameter on the recipient's mailbox is set to `$true`. This is a per-recipient setting that can only be set by an administrator using the **Set-Mailbox** cmdlet. • The message sender is in the recipient's Outlook Safe Senders List. For more information about the Safe Senders List, see Manage safelist aggregation. • The Content Filter agent doesn't process any content filtering for messages that are sent to this recipient. For more information about recipient exceptions, see Manage content filtering.

View anti-spam stamps in Outlook

Anti-spam and anti-malware protection > Anti-spam protection > Anti-spam stamps >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

You can use Microsoft Outlook to view the anti-spam stamps that Microsoft Exchange applied to an email message. Anti-spam stamps help you diagnose spam-related problems by applying diagnostic metadata, or stamps, such as sender-specific information, puzzle validation results, and content filtering results to messages as the messages pass through the anti-spam agents that filter inbound messages from the Internet.

What do you need to know before you begin?

- Estimated time to complete each procedure: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox access" entry in the Mail flow permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use Outlook 2010 or Outlook 2013 to view anti-spam stamps

1. In Outlook 2010 or Outlook 2013, on a client computer, in the **Mail** view, double-click a message to open it.
2. In the **Tags** section of the Ribbon, click the **Options** icon to display the message **Properties** dialog box.
3. In the **Properties** dialog box, in the **Internet headers** section, use the scroll bar to view the anti-spam stamps as shown in the following example.

```
X-MS-Exchange-Organization-PCL:7
X-MS-Exchange-Organization-SCL:6
X-MS-Exchange-Organization-Antispam-Report:
DV:3.1.3924.1409;SID:SenderIDStatus Fail;PCL:PhishingLevel
SUSPICIOUS;CW:CustomList;PP:Presolved;TIME:TimeBasedFeature
s
```

Use Outlook 2007 to view anti-spam stamps

1. In Outlook 2007, on a client computer, in the **Mail** view, double-click a message to open it.
2. On the **Message** tab, in the **Options** group, click **Message Options**.
3. In the **Message Options** dialog box, in the **Internet headers** section, use the scroll bar to view the anti-spam stamps as shown in the following example.

```
X-MS-Exchange-Organization-PCL:7
X-MS-Exchange-Organization-SCL:6
X-MS-Exchange-Organization-Antispam-Report:
```

DV:3.1.3924.1409;SID:SenderIDStatus Fail;PCL:PhishingLevel SUSPICIOUS;CW:CustomList;PP:Presolved;TIME:TimeBasedFeatures

Anti-malware protection

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-08-07*

The Microsoft Exchange Server 2013 anti-malware protection feature helps combat malware in your email messaging environment. *Malware* is comprised of viruses and spyware. *Viruses* infect other programs and data, and they spread throughout your computer looking for programs to infect. *Spyware* refers to malware that gathers your personal information, such as sign-in information and personal data, and sends it back to its author.

There are several anti-malware protection options in Exchange 2013:

- **Built-in anti-malware protection in Exchange 2013** You can use the built-in Exchange on-premises anti-malware protection feature in order to help you combat malware. This basic anti-malware protection can be turned off, replaced, or paired with a cloud-based service (such as Microsoft Exchange Online Protection or Microsoft Forefront Online Protection for Exchange) to provide a layered defense. For more information about typical questions and answers regarding the product's built-in anti-malware capabilities, see Anti-malware FAQ. For information about configuring your anti-malware policies so that they are tailored to best meet the needs of your organization, see Configure anti-malware policies.
- **Cloud-hosted anti-malware protection** You can elect to purchase the Microsoft Forefront Online Protection for Exchange (FOPE) hosted email filtering service or the next version of this service, Exchange Online Protection (EOP). The service leverages partnerships with several best of breed anti-malware engines, thereby providing efficient, cost effective, multi-layered anti-malware protection.

Note:

EOP anti-malware protection features are included in Microsoft Exchange Online. For more information about these features, see **Anti-Malware Protection**.

- **Third-party anti-malware protection** You can also use a third-party anti-malware protection program. In this case, you may want to disable the built-in anti-malware protection; for more information, see Disable or bypass anti-malware scanning.

For more information

Download engine and definition updates

Rescan messages already malware scanned by the hosted filtering service

Anti-Virus Software in the Operating System on Exchange Servers

Anti-malware FAQ

Exchange Server 2013 > Anti-spam and anti-malware protection > Anti-malware protection >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-07*

This topic provides frequently asked questions about malware filtering (scanning) in Microsoft Exchange Server 2013.

Q. Where does malware scanning occur?

A. Malware scanning is performed on messages sent to or received from a mailbox server. Malware scanning is not performed on a message accessed from a mailbox because it should have already been scanned. If a message is re-sent from a mailbox, it's rescanned.

Q. Do I need Internet access in order to download engine and definition updates?

A. To download updates, you must be able to access the Internet and be able to establish a connection on TCP port 80 (HTTP). We strongly recommend that you manually download anti-malware engine and definition updates on your Exchange server prior to placing it in production. For more information, see Download engine and definition updates.

Q. How often are the malware definitions updated?

A. Each server checks for new malware definitions every hour.

What are some advantages of pairing the built-in malware scanning feature with the FOPE cloud-hosted email filtering service (or the next version of this service, Exchange Online Protection (EOP))?

A. There are several advantages:

- The service uses multiple anti-malware engines whereas the built-in anti-malware protection uses a single engine.
- The service has reporting capabilities including malware statistics.
- The service provides the message trace feature for self-troubleshooting mail flow problems including malware detections.

Q. Why did this malware make it past the filter?

A. There are two possible reasons why you may have received malware.

The first, and more likely scenario, is that the attachment received does not contain any active malicious code. In these situations, some anti-malware engines that run on computers may be more

aggressive and stop messages with truncated payloads.

The second is that the malware you received is a new variant and our anti-malware engine has not yet released a pattern file for the service to deploy.

Q. How can I submit malware that made it past the filter to Microsoft?

A. If you have received malware such as a virus that made it past the filter, please save a copy of the email message with its attached virus, go to the Malware Protection Center and submit a sample using the detailed instructions on that page. When submitting the file, in the **Product** drop-down list select **Other**, select the **I believe this file contains malware** option, and in the **Comments** field specify **Exchange Server 2013**. After we receive the sample, we'll investigate and if it's determined that the sample contains malware, we'll take corrective action to prevent the virus from going undetected.

Q. How can I submit a file that I believe was incorrectly detected as malware?

A. Similar to submitting malware, go to the Malware Protection Center and submit a sample using the detailed instructions on that page. When submitting the file, in the **Product** drop-down list select **Other**, select the **I believe this file should not be detected as malware** option, and in the **Comments** field specify **Exchange Server 2013**. After we receive the sample, we'll investigate and if it's determined that the sample is clean, we'll take corrective action to prevent the file from being detected as malware.

Q. I received an email with an attachment that I am not familiar with. Is this malware or can I disregard this attachment?

A. We strongly advise that you do not open any attachments that you do not recognize. If you would like us to investigate the attachment, go to the Malware Protection Center and submit the possible malware to us as described previously.

Q. Where can I get the messages that have been deleted by the malware filter?

A. The messages contain active malicious code and therefore we do not allow access to these messages. They are simply deleted.

Q. I am not able to receive a specific attachment because it's being falsely filtered by your malware filter. Can I allow this attachment through via Exchange transport rules?

A. No. Transport rules cannot be used to bypass the malware filter. If you would like this attachment to bypass the malware filter, send the attachment to the intended recipient within a password protected .zip file. Any password protected file is bypassed by malware filtering.

Q. Can I turn off the product's built-in anti-malware protection?

A. The built-in anti-malware scanning can be permanently disabled or temporarily bypassed by following the steps in Disable or bypass anti-malware scanning.

For more information

Configure anti-malware policies

Download engine and definition updates

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-10-22*

Microsoft Exchange Server 2013 administrators can manually download anti-malware engine and definition (signature) updates. We strongly recommend that you download engine and definition updates on your Exchange server prior to placing it in production.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- To download updates, your computer must be able to access the Internet and be able to establish a connection on TCP port 80 (HTTP).
- You can only use the Shell to perform this procedure.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to manually download engine and definition updates

To download engine and definition updates, run the following command:

```
& $env:ExchangeInstallPath\Scripts\Update-MalwareFilteringServer.ps1 -Identity <FQDN of server>
```

This example manually downloads the engine and definition updates on a server named mailbox01.contoso.com:

```
& $env:ExchangeInstallPath\Scripts\Update-
MalwareFilteringServer.ps1 -Identity mailbox01.contoso.com
```

Optionally, you can specify the –EngineUpdatePath parameter which lets you download updates from somewhere other than the default of http://forefrontdl.microsoft.com/server/scanengineupdate. This can be an HTTP address or a UNC path; if the latter then the network service must have access to the path. This example manually downloads engine and definition updates from a local directory onto a server named mailbox01.contoso.com:

```
& $env:ExchangeInstallPath\Scripts\Update-
MalwareFilteringServer.ps1 -Identity mailbox01.contoso.com
-EngineUpdatePath \\Server\sharename
```

How do you know this worked?

In order to verify that updates were downloaded successfully, you need to access Event Viewer and view the event log. We recommend that you filter only FIPFS events, as described in the following procedure.

1. From the **Start** menu, click **All Programs** > **Administrative Tools** > **Event Viewer**.
2. In Event Viewer, expand the **Windows Logs** folder, and then click **Application**.
3. In the **Actions** menu, click **Filter Current Log**.
4. In the **Filter Current Log** dialog box, from the **Event sources** drop-down list, select the **FIPFS** check box, and then click **OK**.

If engine updates were downloaded successfully, you will see Event ID 6033, which will appear similar to the following:

```
MS Filtering Engine Update process performed a successful scan engine update.

Scan Engine: Microsoft

Update Path: http://forefrontdl.microsoft.com/server/scanengineupdate

Last Update time: 2012-08-16T13:22:17.000Z

Engine Version: 1.1.8601.0

Signature Version: 1.131.2169.0
```

For more information

Configure anti-malware policies

Configure anti-malware policies

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-08-07*

By default, malware filtering is enabled in Microsoft Exchange Server 2013. The default anti-malware policy controls your company-wide malware filtering settings. As an administrator, you can view and edit, but not delete, the default anti-malware policy so that it is tailored to best meet the needs of your organization. For greater granularity, you can also create custom malware filter policies and apply them to specified users, groups, or domains in your organization. Custom policies always take precedence over the default policy, but you can change the priority (that is, the running order) of your custom policies. For more information, see Use the EAC to configure the default anti-malware policy.

What do you need to know before you begin?

- We recommend that you manually download anti-malware engine and definition updates on your Exchange server prior to placing it in production. For more information, see Download engine and definition updates.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to configure the default anti-malware policy

1. In the Exchange admin center (EAC), navigate to **Protection** > **Malware filter**.
2. Do one of the following:
   - Double-click the default policy in order to edit this company-wide policy.
   - Click the **+ New** icon in order to create a new policy that can be applied to users, groups, and domains in your organization. You can also edit existing custom policies by double-clicking them.
3. For custom policies only, specify a name for this policy. You can optionally specify a more detailed description as well. You cannot rename the default policy.

Note:

When creating a new policy, all configuration settings appear on a single screen, whereas when editing a policy you must navigate through different screens. The settings are the same in either case, but the rest of this procedure describes how to access these settings when editing a policy.

4. Click the **Settings** menu option. In the **Malware Detection Response** section, use the option buttons to select the action to take when malware is detected in a message:
   - **Delete the entire message** Prevents the entire message, including attachments, from being delivered to the intended recipients. This is the default value.
   - **Delete all attachments and use default alert text** Deletes all message attachments, not just the infected one, and inserts the following default alert text into a text file that replaces the attachments: "Malware was detected in one or more attachments included with this email. All attachments have been deleted."
   - **Delete all attachments and use custom alert text** Deletes all message attachments, not just the infected one, and inserts a custom message into a text file that replaces the attachments. Selecting this option enables the **Custom alert text** field where you must type a custom message.

Important:

If malware is detected in the message body, the entire message, including all attachments, will be deleted regardless of which option you select. This action is applied to both inbound and outbound messages.

5. In the **Notifications** section, you have the option to send a notification email message to senders or administrators when a message is detected as malware and is not delivered. These notifications are only sent when the entire message is deleted.
   - In the **Sender Notifications** section, select the check boxes to **Notify internal senders** (those within your organization) or to **Notify external senders** (those outside your organization) when a detected message is not delivered.
   - Similarly, in the **Administrator Notifications** section, select the check boxes to **Notify administrator about undelivered messages from internal senders** or to **Notify administrator about undelivered messages from external senders**. Specify the email address or addresses of the administrator in their respective **Administrator email address** fields after selecting one or both of these check boxes. Use a semicolon to separate multiple addresses.

The default notification text is "This message was created automatically by mail delivery software. Your email message was not delivered to the intended recipients because malware was detected." The language in which the default notification text is sent is dependent on the locale of the message being processed.

   - In the **Customize Notifications** section, you can create customized notification text to be used in place of the default notification text for sender and administrator notifications. Select the **Use customized notification text** check box, and then specify values in the following required fields:
     - **From name** The name you want to be used as the sender of the customized notification.
     - **From address** The email address you want to be used as the sender of the customized

notification.

- **Messages from internal senders** The **Subject** and **Message** of the notification if the detected message originated from an internal sender.
- **Messages from external senders** The **Subject** and **Message** of the notification if the detected message originated from an external sender.

Note:

The default Subject text is "Undeliverable message."

- For custom policies only, click the **Apply to** menu item and then create a condition-based rule to specify the users, groups, and/or domains for whom to apply this policy. You can create multiple conditions provided that they are unique.
  - To select users, select **The recipient is**. In the subsequent dialog box, select one or more senders from your company from the user picker list and then click **add**. To add senders who aren't on the list, type their email addresses and click **Check names**. In this box, you can also use wildcards for multiple email addresses (for example: *@*domainname*). When you are done with your selections, click **ok** to return to the main screen.
  - To select groups, select **The recipient is a member of** and then, in the subsequent dialog box, select or specify the groups. Click **ok** to return to the main screen.
  - To select domains, select **The recipient domain is** and then, in the subsequent dialog box, add the domains. Click **ok** to return to the main screen.

You can create exceptions within the rule, for example you can filter messages from all domains except for a certain domain. Click **add exception** and then create your exception conditions similar to the way you created the other conditions.

- Click **Save**. A summary of your default policy settings appears in the right pane.

Tip:

- You can select or clear the check boxes in the **ENABLED** column to enable or disable your custom policies. All policies are enabled by default, and the default policy cannot be disabled.
- To delete a custom policy, select the policy, click the **Delete** icon, and then confirm that you want to delete the policy. The default policy cannot be deleted.
- Custom policies always take precedence over the default policy. Custom policies run in the reverse order that you created them (from oldest to newest), but you can change the priority (running order) of your custom policies by clicking the up arrow and down arrow. The policy with a **PRIORITY** of **0** will run first, followed by **1**, then **2**, and so on.

How do you know this worked?

The following procedure provides instructions for using the EICAR.TXT antivirus test file to verify that malware filtering is working correctly.

Important:

The EICAR.TXT file is not a virus. However, because users often have the need to test that installations function correctly, the antivirus industry, through the European Institute for

Computer Antivirus Research, has adopted the EICAR standard in order to meet this need.

1. Create a new text file, and then name the file EICAR.TXT.
2. Copy the following line into the text file:

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

Make sure that this is the only string in the file. When done, you will have a 68-byte file.

Note:

If you are using a desktop antivirus program, make sure that the folder you are saving the file to is excluded from scanning.

3. Attach this file to an email message that will be filtered by Exchange 2013.

Check the recipient mailbox of the test message. Depending on the malware detection response you have configured, the entire message will be deleted, or the attachment will be deleted and replaced with the alert text file. Any configured notifications will also be distributed.

4. Delete the EICAR.TXT file after testing is completed so that other users are not unnecessarily alarmed.

For more information

Anti-malware FAQ

Rescan messages already malware scanned by the hosted filtering service

Disable or bypass anti-malware scanning

Rescan messages already malware scanned by the hosted filtering service

Exchange Server 2013 > Anti-spam and anti-malware protection > Anti-malware protection >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-18*

You can configure Microsoft Exchange Server 2013 to rescan email messages already scanned for malware by the hosted email filtering service. Enabling this functionality provides another layer of defense against malware because the cloud-hosted filtering only scans inbound and outbound messages. Internal messages are scanned by the built-in anti-malware protection capabilities of Exchange 2013. By default, messages scanned in the cloud are not resubmitted for malware scanning on-premises.

Note:
This topic only applies to Microsoft Exchange Server 2013 customers who are using cloud-hosted email filtering.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You can only use the Shell to perform this procedure.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to rescan messages already malware scanned by the hosted filtering service

1. To rescan messages already malware scanned by the hosted filtering service, run the following command:

```
Set-MalwareFilteringServer -ForceRescan $true
```

Note:
To return to the default setting of not rescanning messages, re-set the above parameter to `$false`.

How do you know this step worked?

To verify that your Exchange 2013 server is rescanning messages for malware, run the following command and confirm that it returns a value of True:

```
Get-MalwareFilteringServer | Format-List ForceRescan
```

Disable or bypass anti-malware scanning

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-03*

In Microsoft Exchange Server 2013, you can disable or bypass malware filtering of all email messages in transit on a server. This must be done on a Mailbox server.

You may want to disable Exchange 2013 malware filtering if you are using another product for malware filtering. When malware filtering is disabled, the Exchange malware agent is unhooked and not running, and engine updates are not kept up-to-date.

Important:

Bypassing malware filtering should *only* be done when troubleshooting a problem. When malware filtering is bypassed, the Exchange malware agent remains hooked, and engine updates are kept up-to-date. However, malware filtering is skipped while you attempt to resolve whatever problems you are encountering. After you have finished troubleshooting, you should restore malware filtering.

What do you need to know before you begin?

- Estimated time to complete each procedure: 15 minutes
- You can only use the Shell to perform this procedure.
- Disabling or enabling malware filtering restarts the Microsoft Exchange Transport service on the server. This may temporarily disrupt mail flow in your organization.
- Bypassing or restoring malware filtering doesn't require you to restart any services. However, changes to the setting may take up to 10 minutes to take effect.
- If you have multiple Exchange servers performing malware filtering, you must perform these steps on each server.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to disable malware filtering on a specific Exchange server

To disable malware filtering, run the following command:

```
& $env:ExchangeInstallPath\Scripts\Disable-
Antimalwarescanning.ps1
```

Note:

To re-enable malware filtering, use `Enable-Antimalwarescanning.ps1` instead of `Disable-Antimalwarescanning.ps1`.

How do you know this step worked?

To verify that malware filtering is disabled, run the following command and confirm that it returns a value of False:

```
Get-TransportAgent "Malware Agent"
```

Use the Shell to temporarily bypass malware filtering on a specific Exchange server

Important:

Bypassing malware filtering should *only* be done when troubleshooting a problem. You should restore malware filtering after you have finished troubleshooting.

To temporarily bypass malware filtering, run the following command:

```
Set-MalwareFilteringServer <ServerIdentity> -
BypassFiltering $true
```

To restore malware filtering, run the following command:

```
Set-MalwareFilteringServer <ServerIdentity> -
BypassFiltering $false
```

How do you know this step worked?

To verify that malware filtering is being bypassed, run the following command and confirm that it returns a value of True:

```
Get-MalwareFilteringServer | Format-List BypassFiltering
```

Anti-Virus Software in the Operating

System on Exchange Servers

Topic Last Modified: *2014-09-02*

This topic describes the effects of file-level antivirus programs on computers that are running Microsoft Exchange Server 2013. If you implement the recommendations described in this topic, you can help enhance the security and health of your Exchange organization.

File-level scanners are frequently used. However, if they are configured incorrectly, they can cause problems in Exchange 2013. There are two types of file-level scanners:

- *Memory-resident file-level scanning* refers to a part of file-level antivirus software that is loaded in memory at all times. It checks all the files that are used on the hard disk and in computer memory.
- *On-demand file-level scanning* refers to a part of file-level antivirus software that you can configure to scan files on the hard disk manually or on a schedule. Some versions of antivirus software start the on-demand scan automatically after virus signatures are updated to make sure that all files are scanned with the latest signatures.

The following problems may occur when you use file-level scanners with Exchange 2013:

- File-level scanners may scan a file when the file is being used or at a scheduled interval. This can cause the scanners to lock or quarantine an Exchange log or a database file while Exchange 2013 tries to use the file. This behavior may cause a severe failure in Exchange 2013 and may also cause -1018 event log errors.
- File-level scanners don't provide protection against email viruses, such as Storm Worm. Storm Worm was a backdoor Trojan horse program that propagated itself through email messages. The worm joined the infected computer to a botnet, where the computer was used to send spam in periodic bursts.

Recommendations for using file-level scanning with Exchange 2013

If you're deploying file-level scanners on Exchange 2013 servers, make sure that the appropriate exclusions, such as directory exclusions, process exclusions, and file name extension exclusions, are in place for both memory-resident and file-level scanning. This section describes recommended directory exclusions, process exclusions, and file name extension exclusions.

Contents

Directory exclusions

You must exclude specific directories for each Exchange server on which you run a file-level antivirus scanner. This section describes the directories that you should exclude from file-level scanning.

Mailbox servers

- **Mailbox databases**
  - Exchange databases, checkpoint files, and log files. By default, these are located in sub-folders under the %ExchangeInstallPath%Mailbox folder. To determine the location of a mailbox database, transaction log, and checkpoint file, run the following command: `Get-MailboxDatabase -Server <servername>| Format-List *path*`
  - Database content indexes. By default, these are located in the same folder as the database file.
  - Group Metrics files. By default, these files are located in the %ExchangeInstallPath% GroupMetrics folder.
  - General log files, such as message tracking and calendar repair log files. By default, these files are located in subfolders under the %ExchangeInstallPath%TransportRoles\Logs folder and % ExchangeInstallPath%Logging folder. To determine the log paths being used, run the following command in the Exchange Management Shell: `Get-MailboxServer <servername> | Format-List *path*`
  - The Offline Address Book files. By default, these are located in subfolders under the % ExchangeInstallPath%ClientAccess\OAB folder.
  - IIS system files in the %SystemRoot%\System32\Inetsrv folder.
  - The Mailbox database temporary folder: %ExchangeInstallPath%Mailbox\MDBTEMP
- **Members of Database Availability Groups**
  - All the items listed in the **Mailbox databases** list, and the cluster quorum database that exists at %Windir%\Cluster.
  - The witness directory files. These files are located on another server in the environment, typically a Client Access server that isn't installed on the same computer as a Mailbox server. By default, the witness directory files are located in %SystemDrive%:\DAGFileShareWitnesses \<DAGFQDN>.
- **Transport service**
  - Log files, for example, message tracking and connectivity logs. By default, these files are located in subfolders under the %ExchangeInstallPath%TransportRoles\Logs folder. To determine the log paths being used, run the following command in the Exchange Management Shell: `Get-TransportService <servername> | Format-List *logpath*,*tracingpath*`
  - Pickup and Replay message directory folders. By default, these folders are located under the % ExchangeInstallPath%TransportRoles folder. To determine the paths being used, run the following command in the Exchange Management Shell: `Get-TransportService <servername>| Format-List *dir*path*`
  - The queue databases, checkpoints, and log files. By default, these are located in the % ExchangeInstallPath%TransportRoles\Data\Queue folder.
  - The Sender Reputation database, checkpoint, and log files. By default, these are located in the %ExchangeInstallPath%TransportRoles\Data\SenderReputation folder.

- The temporary folders that are used to perform conversions:
  - By default, content conversions are performed in the Exchange server's %TMP% folder.
  - By default, rich text format (RTF) to MIME/HTML conversions are performed in %ExchangeInstallPath%Working\OleConverter folder.
- The content scanning component is used by the Malware agent and data loss prevention (DLP). By default, these files are located in the %ExchangeInstallPath%FIP-FS folder.

○ **Mailbox Transport service**

- Log files, for example, connectivity logs. By default, these files are located in subfolders under the %ExchangeInstallPath%TransportRoles\Logs\Mailbox folder. To determine the log paths being used, run the following command in the Exchange Management Shell: `Get-MailboxTransportService <servername> | Format-List *logpath*`

○ **Unified Messaging**

- The grammar files for different locales, for example en-EN or es-ES. By default, these are stored in the subfolders in the %ExchangeInstallPath%UnifiedMessaging\grammars folder.
- The voice prompts, greetings and informational message files. By default, these are stored in the subfolders in the %ExchangeInstallPath%UnifiedMessaging\Prompts folder
- The voicemail files that are temporarily stored in the %ExchangeInstallPath%UnifiedMessaging\voicemail folder.
- The temporary files generated by Unified Messaging. By default, these are stored in the %ExchangeInstallPath%UnifiedMessaging\temp folder.

Client Access servers

○ **Web components**

- For servers using Internet Information Services (IIS) 7.0, the compression folder that is used with Microsoft Outlook Web App. By default, the compression folder for IIS 7.0 is located at %SystemDrive%\inetpub\temp\IIS Temporary Compressed Files.
- IIS system files in the %SystemRoot%\System32\Inetsrv folder
- Inetpub\logs\logfiles\w3svc

○ **POP3 and IMAP4 protocol logging**

- POP3 folder: %ExchangeInstallPath%Logging\POP3
- IMAP4 folder: %ExchangeInstallPath%Logging\IMAP4

○ **Front End Transport service**

- Log files, for example, connectivity logs and protocol logs. By default, these files are located in subfolders under the %ExchangeInstallPath%TransportRoles\Logs\FrontEnd folder. To determine the log paths being used, run the following command in the Exchange Management Shell: `Get-FrontEndTransportService <servername> | Format-List *logpath*`

Return to top

Process exclusions

Many file-level scanners now support the scanning of processes, which can adversely affect Microsoft Exchange if the incorrect processes are scanned. Therefore, you should exclude the following processes from file-level scanners.

Process	Path	Comments	Servers
Dsamain.exe	%SystemRoot%\System32	Active Directory Lightweight Directory Services (AD LDS) on subscribed Edge Transport servers.	Edge Transport servers
EdgeTransport.exe	%ExchangeInstallPath%Bin	Exchange transport worker process (MSExchangeTransport)	Mailbox servers Edge Transport servers
fms.exe	%ExchangeInstallPath%FIP-FS\Bin	Content scanning component that's used by the Malware agent and DLP.	Mailbox servers
hostcontrollerservice.exe	%ExchangeInstallPath%Bin\Search\Ceres\HostController	Microsoft Exchange Search Host Controller service (HostControllerService)	Mailbox servers Client Access servers
inetinfo.exe	%SystemRoot%\System32\inetsrv	Internet Information Services (IIS)	Mailbox servers Client Access servers
Microsoft.Exchange.AntispamUpdateSvc.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Anti-spam Update service (MSExchangeAntispamUpdate)	Mailbox servers Edge Transport servers
Microsoft.Exchange.ContentFilter.Wrapper.exe	%ExchangeInstallPath%TransportRoles\agents\Hygiene	Content Filter agent	Mailbox servers Edge Transport servers
Microsoft.Exchange.Dia	%ExchangeInstallPath%	Microsoft Exchange	Mailbox servers

gnostics.Service.exe	Bin	Diagnostics service (MSExchangeDiagnostics)	Client Access servers Edge Transport servers
Microsoft.Exchange.Directory.TopologyService.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Active Directory Topology service (MSExchangeADTopology)	Mailbox servers Client Access servers
Microsoft.Exchange.EdgeCredentialSvc.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Credential Service (MSExchangeEdgeCredential)	Edge Transport servers
Microsoft.Exchange.EdgeSyncSvc.exe	%ExchangeInstallPath%Bin	Microsoft Exchange EdgeSync service (MSExchangeEdgeSync)	Mailbox servers
Microsoft.Exchange.Imap4.exe	ExchangeInstallPath%FrontEnd\PopImap	Microsoft Exchange IMAP4 service (MSExchangeImap4)	Client Access servers
Microsoft.Exchange.Imap4service.exe	%ExchangeInstallPath%ClientAccess\PopImap	Microsoft Exchange IMAP4 Backend service (MSExchangeIMAP4BE)	Mailbox servers
Microsoft.Exchange.Pop3.exe	%ExchangeInstallPath%FrontEnd\PopImap	Microsoft Exchange POP3 service (MSExchangePop3)	Client Access servers
Microsoft.Exchange.Pop3service.exe	%ExchangeInstallPath%ClientAccess\PopImap	Microsoft Exchange POP3 Backend service (MSExchangePOP3BE)	Mailbox servers
Microsoft.Exchange.Pro	%ExchangeInstallPath%	Microsoft Exchange	Mailbox servers

tectedServiceHost.exe	Bin	Service Host service (MSExchangeServiceHost)	Client Access servers Edge Transport servers
Microsoft.Exchange.RPCClientAccess.Service.exe	%ExchangeInstallPath%Bin	Microsoft Exchange RPC Client Access service (MSExchangeRPC)	Mailbox servers
Microsoft.Exchange.Search.Service.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Search service (MSExchangeFastSearch)	Mailbox servers
Microsoft.Exchange.Servicehost.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Service Host service (MSExchangeServiceHost)	Mailbox servers Client Access servers Edge Transport servers
Microsoft.Exchange.Store.Service.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Information Store service (MSExchangeIS)	Mailbox servers
Microsoft.Exchange.Store.Worker.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Information Store service (MSExchangeIS)	Mailbox servers
Microsoft.Exchange.UM.CallRouter.exe	%ExchangeInstallPath%FrontEnd\CallRouter	Microsoft Exchange Unified Messaging Call Router service (MSExchangeUMCR)	Client Access servers
MSExchangeDagMgmt.exe	%ExchangeInstallPath%Bin	Microsoft Exchange DAG Management service (MSExchangeDagMgmt	Mailbox servers

		)	
MSExchangeDelivery.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Mailbox Transport Delivery service (MSExchangeDelivery)	Mailbox servers
MSExchangeFrontendTransport.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Frontend Transport service (MSExchangeFrontEndTransport)	Client Access servers
MSExchangeHMHost.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Health Manager service (MSExchangeHM)	Mailbox servers Client Access servers Edge Transport servers
MSExchangeHMWorker.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Health Manager service (MSExchangeHM)	Mailbox servers Client Access servers Edge Transport servers
MSExchangeMailboxAssistants.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Mailbox Assistants service (MSExchangeMailboxAssistants)	Mailbox servers
MSExchangeMailboxReplication.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Mailbox Replication service (MSExchangeMailboxReplication)	Mailbox servers
MSExchangeMigrationWorkflow.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Migration Workflow	Mailbox servers

		service (MSExchangeMigrationWorkflow)	
MSExchangeRepl.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Replication service (MSExchangeRepl)	Mailbox servers
MSExchangeSubmission.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Mailbox Transport Submission service (MSExchangeSubmission)	Mailbox servers
MSExchangeTransport.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Transport service (MSExchangeTransport)	Mailbox servers Edge Transport servers
MSExchangeTransportLogSearch.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Transport Log Search service (MSExchangeTransportLogSearch)	Mailbox servers Edge Transport servers
MSExchangeThrottling.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Throttling service (MSExchangeThrottling)	Mailbox servers
Noderunner.exe	%ExchangeInstallPath%Bin\Search\Ceres\Runtime\1.0	Microsoft Exchange Search service (MSExchangeFastSearch)	Mailbox servers
OleConverter.exe	%ExchangeInstallPath%	Content conversion.	Mailbox servers

	Bin	Used to convert rich text format (RTF) messages to MIME/ HTML for external recipients.	
ParserServer.exe	%ExchangeInstallPath%Bin\Search\Ceres\ParserServer	Microsoft Exchange Search service (MSExchangeFastSearch)	Mailbox servers
Powershell.exe	C:\Windows\System32\WindowsPowerShell\v1.0	Exchange Management Shell	Mailbox servers Client Access servers Edge Transport servers
ScanEngineTest.exe	%ExchangeInstallPath%FIP-FS\Bin	Content scanning component that's used by the Malware agent and DLP.	Mailbox servers
ScanningProcess.exe	%ExchangeInstallPath%FIP-FS\Bin	Content scanning component that's used by the Malware agent and DLP.	Mailbox servers
TranscodingService.exe	%ExchangeInstallPath%ClientAccess\Owa\Bin\DocumentViewing	WebReady Document Viewing in Outlook Web App.	Mailbox servers
UmService.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Unified Messaging service (MSExchangeUM)	Mailbox servers
UmWorkerProcess.exe	%ExchangeInstallPath%Bin	Microsoft Exchange Unified Messaging	Mailbox servers

		service (MSExchangeUM)	
UpdateService.exe	%ExchangeInstallPath%FIP-FS\Bin	Content scanning component that's used by the Malware agent and DLP.	Mailbox servers
W3wp.exe	%SystemRoot%\System32\inetsrv	Internet Information Services (IIS)	Mailbox servers Client Access servers

Return to top

File name extension exclusions

In addition to excluding specific directories and processes, you should exclude the following Exchange-specific file name extensions in case directory exclusions fail or files are moved from their default locations.

Application-related extensions:

- .config
- .dia
- .wsb

Database-related extensions:

- .chk
- .edb
- .jrs
- .jsl
- .log
- .que

Offline address book-related extensions:

- .lzx

Content Index-related extensions:

- .ci
- .dir
- .wid
- .000
- .001
- .002

Unified Messaging-related extensions:

- .cfg

- .grxml

Group Metrics-related extensions:

- .dsc
- .txt

Return to top

Mail flow

Topic Last Modified: *2014-02-17*

In Microsoft Exchange Server 2013, mail flow occurs through the transport pipeline. The *transport pipeline* is a collection of services, connections, components, and queues that work together to route all messages to the categorizer in the Transport service on a Mailbox server inside the organization.

Looking for a list of all mail flow topics? See Mail flow documentation.

For information about how to configure mail flow in a new Exchange 2013 organization, see Configure mail flow and client access.

Contents

The transport pipeline

The transport pipeline consists of the following services:

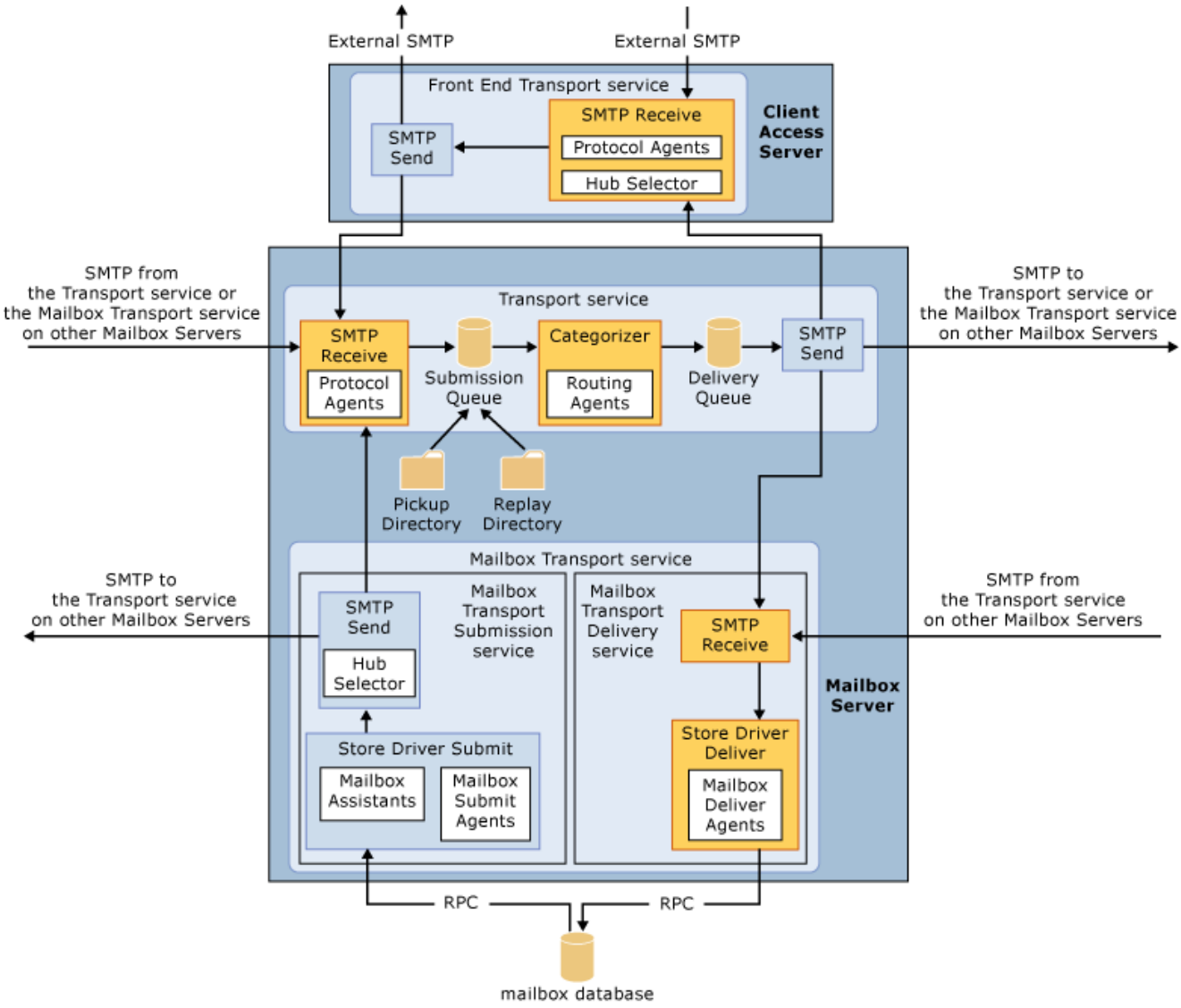
- **Front End Transport service on Client Access servers** This service acts as a stateless proxy for all inbound and (optionally) outbound external SMTP traffic for the Exchange 2013 organization. The Front End Transport service doesn't inspect message content, doesn't communicate with the Mailbox Transport service on Mailbox servers, and doesn't queue any messages locally.
- **Transport service on Mailbox servers** This service is virtually identical to the Hub Transport server role in previous versions of Exchange. The Transport service handles all SMTP mail flow for the organization, performs message categorization, and performs message content inspection. Unlike previous versions of Exchange, the Transport service never communicates directly with mailbox databases. That task is now handled by the Mailbox Transport service. The Transport service routes messages between the Mailbox Transport service, the Transport service, the Front End Transport service, and (depending on your configuration) the Transport service on Edge

Transport servers. The Transport service on Mailbox servers is described in more detail later in this topic.

- **Mailbox Transport service on Mailbox servers** This service consists of two separate services: the Mailbox Transport Submission service and Mailbox Transport Delivery service. The Mailbox Transport Delivery service receives SMTP messages from the Transport service on the local Mailbox server or on other Mailbox servers, and connects to the local mailbox database using an Exchange remote procedure call (RPC) to deliver the message. The Mailbox Transport Submission service connects to the local mailbox database using RPC to retrieve messages, and submits the messages over SMTP to the Transport service on the local Mailbox server, or on other Mailbox servers. The Mailbox Transport Submission service has access to the same routing topology information as the Transport service. Like the Front End Transport service, the Mailbox Transport service also doesn't queue any messages locally.
- **Transport service on Edge Transport servers** This service is very similar to the Transport service on Mailbox servers. If you have an Edge Transport server installed in the perimeter network, all mail coming from the Internet or going to the Internet flows through the Transport service Edge Transport server. This service is described in more detail later in this topic.

The following figure shows the relationships among the components in the Exchange 2013 transport pipeline.

Overview of the transport pipeline in Exchange 2013.



Messages from external senders

Messages from outside the organization enter the transport pipeline through a Receive connector in the Front End Transport service on the Client Access server and are then routed to the Transport service on the Mailbox server.

If you have an Exchange 2013 Edge Transport server installed in the perimeter network, messages from outside the organization enter the transport pipeline through a Receive connector in the Transport service on the Edge Transport server. Where the messages go next depends on how your internal Exchange servers are configured.

- **Mailbox server and Client Access server installed on the same computer** In this configuration, the Client Access server is used for inbound mail flow. Mail flows from the Transport service on the Edge Transport server to the Front End Transport service on the Client Access server, and then to the Transport service on the Mailbox server.
- **Mailbox server and Client Access server installed on different computers** In this configuration, the Client Access server is bypassed for inbound mail flow. Mail flows from the Transport service on the Edge Transport server to the Transport service on the Mailbox server.

Note:

If you have an Exchange 2010 or Exchange 2007 Edge Transport server installed in your perimeter network, mail flow always occurs directly between the Edge Transport server and the Transport service on the Mailbox server. For more information, see Use an Exchange 2010 or 2007 Edge Transport server in Exchange 2013.

Messages from internal senders

SMTP messages from inside the organization enter the transport pipeline through the Transport service on a Mailbox server in one of the following ways:

- Through a Receive connector.
- From the Pickup directory or the Replay directory.
- From the Mailbox Transport service.
- Through agent submission.

The message is routed based on the routing destination or delivery group. For more information, see Mail routing.

If the message has external recipients, the message is routed from the Transport service on the Mailbox server to the Internet, or from the Mailbox server to the Front End Transport service on a Client Access server and then to the Internet if the Send connector is configured to proxy outbound connections through the Client Access server. For more information, see Create a Send connector for email sent to the Internet.

If you have an Edge Transport server installed in the perimeter network, messages that have external recipients are never routed through the Front End Transport service on a Client Access server. The message is routed from the Transport service on a Mailbox server to the Transport service on the Edge Transport server

Transport service on Mailbox servers

Every message that's sent or received in an Exchange 2013 organization must be categorized in the Transport service on a Mailbox server before it can be routed and delivered. After a message has been categorized, it's put in a delivery queue for delivery to the destination mailbox database, the destination database availability group (DAG), Active Directory site, or Active Directory forest, or to the destination domain outside the organization.

The Transport service on a Mailbox server consists of the following components and processes:

- **SMTP Receive** When messages are received by the Transport service, message content inspection is performed, transport rules are applied, and anti-spam and anti-malware inspection is performed if they are enabled. The SMTP session has a series of events that work together in a specific order to validate the contents of a message before it's accepted. After a message has passed completely through SMTP Receive and isn't rejected by receive events, or by an anti-spam and anti-malware agent, it's put in the Submission queue.
- **Submission** Submission is the process of putting messages into the Submission queue. The categorizer picks up one message at a time for categorization. Submission happens in three ways:

- From SMTP Receive through a Receive connector.
- Through the Pickup directory or the Replay directory. These directories exist on Mailbox servers and Edge Transport servers. Correctly formatted message files that are copied into the Pickup directory or the Replay directory are put directly into the Submission queue.
- Through a transport agent.

- **Categorizer** The categorizer picks up one message at a time from the Submission queue. The categorizer completes the following steps:
  - Recipient resolution, which includes top-level addressing, expansion, and bifurcation.
  - Routing resolution.
  - Content conversion.

Additionally, mail flow rules that are defined by the organization are applied. After messages have been categorized, they're put into a delivery queue that's based on the destination of the message. Messages are queued by the destination mailbox database, DAG, Active Directory site, Active Directory forest or external domain.

- **SMTP Send** How messages are routed from the Transport service depends on the location of the message recipients relative to the Mailbox server where categorization occurred. The message could be routed to one of the following locations:
  - To the Mailbox Transport service on the same Mailbox server.
  - To the Mailbox Transport service on a different Mailbox server that's part of the same DAG.
  - To the Transport service on a Mailbox server in a different DAG, Active Directory site, or Active Directory forest.
  - For delivery to the Internet through a Send connector on the same Mailbox server, through the Transport service on a different Mailbox server, through the Front End Transport service on a Client Access server, or through the Transport service on an Edge Transport server in the perimeter network.

Transport service on Edge Transport servers

The components of the Transport service on Edge Transport servers are identical to the components of the Transport service on Mailbox servers. However, what actually happens during each stage of processing on Edge Transport servers is different. The differences are described in the following list.

- **SMTP Receive** When an Edge Transport server is subscribed to an internal Active Directory site, the default Receive connector is automatically configured to accept mail from internal Mailbox servers and from the Internet. When Internet messages arrive at the Edge Transport server, anti-spam agents filter connections and message contents, and help identify the sender and the recipient while the message is being accepted into the organization. The anti-spam agents are installed and enabled by default. Additional attachment filtering and connection filtering features are available, but built-in malware filtering is not. Also, transport rules are controlled by the Edge Rule agent. Compared to the Transport Rule agent on Mailbox servers, only a small subset of transport rule conditions are available on Edge Transport servers. But, there are unique transport rule actions related to SMTP connections that are available only on Edge Transport servers.

- **Submission** On an Edge Transport server, messages typically enter the Submission queue through a Receive connector. However, the Pickup directory and the Replay directory are also available.
- **Categorizer** On an Edge Transport server, categorization is a short process in which the message is put directly into a delivery queue for delivery to internal or external recipients.
- **SMTP Send** When an Edge Transport server is subscribed to an internal Active Directory site, two Send connectors are automatically created and configured. One is responsible for sending outbound mail to Internet recipients; the other is responsible for sending inbound mail from the Internet to internal recipients. Inbound mail is sent to the Transport service on an available Mailbox server in the subscribed Active Directory site.

Mail flow documentation

The following table contains links to topics that will help you learn about and manage mail flow in Exchange 2013.

Topic	Description
Mail routing	Mail routing describes how messages are transmitted between messaging servers.
Connectors	Connectors define where and how messages are transmitted to and from Exchange servers.
Domains	Accepted domains define the SMTP address spaces that are used in the Exchange organization. Remote domains configure message formatting and encoding settings for messages sent to external domains.
Transport agents	Transport agents act on messages as they travel through the Exchange transport pipeline.
Transport high availability	Transport high availability describes how Exchange 2013 keeps redundant copies of messages during transit and after delivery.
Transport logs	Transport logs record what happens to messages as they flow through the transport pipeline.

Managing message approval	Moderated transport requires approval for messages sent to specific recipients.
Content conversion	Content conversion controls the Transport Neutral encoding format (TNEF) message conversion options for external recipients, and the MAPI conversion options for internal recipients.
DSNs and NDRs	Delivery status notifications (DSNs) are the system messages that are sent to message senders, for example, non-delivery reports (NDRs).
Track messages with delivery reports	Delivery Reports is a message tracking tool that you can use to search for delivery status on email messages sent to or from users in your organization's address book, with a certain subject. You can track delivery information about messages sent by or received from any specific mailbox in your organization.
Message size limits	This topic describes the size and individual component limits that are imposed on messages.
Queue Viewer	You use the Queue Viewer in the Exchange Toolbox to view and act upon queues and message in queues.
Pickup directory and Replay directory	The pickup and replay directories are used to insert message files into the transport pipeline.
Use an Exchange 2010 or 2007 Edge Transport server in Exchange 2013	This topic describes the considerations for using an Edge Transport server from previous

	versions of Exchange in Exchange 2013.

Mail routing

Exchange Server 2013 > Mail flow >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-17*

The primary task of the Transport service that exists on all Mailbox servers in your Microsoft Exchange Server 2013 organization is to route messages received from users and external sources to their ultimate destinations. Routing decisions are made during message categorization. The categorizer is a component of the Transport service on a Mailbox server that processes all incoming messages and determines what to do with the message based on information about their destinations.

Routing in Exchange 2013 is now fully aware of Database Availability Groups (DAGs), and uses DAG membership as a routing boundary. Why? In Exchange 2013, all Mailbox servers host the Transport service. Therefore, when a Mailbox server belongs to a DAG, the primary mechanism for routing messages is closely aligned with the DAG. And when a DAG spans multiple Active Directory sites, using the Active Directory site as the primary routing boundary is inefficient. Exchange 2013 also uses Active Directory site membership as a routing boundary for Mailbox servers that don't belong to DAGs, and for routing interoperability with previous versions of Exchange. Other notable changes to Exchange 2013 routing include:

- The Transport service on a Mailbox server never communicates directly with a mailbox database. Instead, the Transport service communicates with the Mailbox Transport service on the Mailbox server. Only the Mailbox Transport service communicates with the mailbox database on the local Mailbox server. When the Mailbox server is a member of a DAG, only the Mailbox Transport service on the Mailbox server that holds the active copy of the mailbox database accepts the message for the destination recipient.
- Remote procedure calls (RPCs) are only used by the Mailbox Transport service when sending messages to or receiving messages from the local mailbox database. When the Mailbox server is a member of a DAG, the Mailbox Transport service only uses RPCs to communicate locally with the active copies of the mailbox databases. In other words, RPC is never used for cross-server communication. Instead, the Mailbox Transport service and the Transport service on different Mailbox servers always communicate using SMTP.
- Exchange 2013 uses more precise queuing for remote destinations. Instead of using one queue for all destinations in a remote Active Directory site, Exchange 2013 queues messages for specific destinations within the Active Directory site, such as individual Send connectors.
- Linked connectors have been deprecated. A linked connector was a Receive connector that was

linked to a Send connector. All messages received by the Receive connector were automatically forwarded to the Send connector.

Contents

Routing components

When a message is received by the Transport service on an Exchange 2013 Mailbox server, the message must be categorized. The first phase of message categorization is recipient resolution. After the recipient has been resolved, the ultimate destination can be determined. The next phase, routing, determines how to best reach that destination. Routing in Exchange 2013 has been generalized for increased flexibility and decreased complexity by introducing the concepts of routing destinations and delivery groups.

Routing destinations

In Exchange 2013, the ultimate destination for a message is called a *routing destination*. The following routing destinations exist in Exchange 2013:

- **A mailbox database** This is the routing destination for any recipient with a mailbox on a Mailbox server in the Exchange organization. In Exchange 2013, public folders are a type of mailbox, so routing messages to public folder recipients is the same as routing messages to mailbox recipients.
- **A connector** A connector is a Send connector for SMTP messages when used as a routing destination. A Delivery Agent connector or a Foreign connector is used as a routing destination for non-SMTP messages.
- **A distribution group expansion server** This is the routing destination when a distribution group has a designated expansion server that's responsible for expanding the membership list of the group. A distribution group expansion server is always a Hub Transport server or an Exchange 2013 Mailbox server.

Note that these same routing destinations also existed in previous versions of Exchange.

Return to top

Delivery groups

Each routing destination in Exchange 2013 has a collection of one or more transport servers that are responsible for delivering messages to that routing destination. This collection of transport servers is called a *delivery group*. A transport server could be an Exchange 2013 Mailbox server, or an Exchange 2010 server or Exchange 2007 server that has the Hub Transport server role installed. When the routing destination is a mailbox database, the transport servers in the delivery group are the same version of Exchange as the mailbox database. When the routing destination is a connector or a distribution group expansion server, the delivery group may contain a mixture of Exchange 2013 Mailbox servers and Exchange 2010 or Exchange 2007 Hub Transport servers. How the message is routed depends on the relationship between the source transport server and the destination delivery group:

- If the source transport server is in the destination delivery group, the routing destination itself is the next hop for the message. The message is delivered by the source transport server to the mailbox database or connector on a transport server in the delivery group. Note that when a distribution group expansion server is the routing destination, the distribution group is already expanded by the time messages reach the routing stage of categorization on the distribution group expansion server. Therefore, the routing destination from the distribution group expansion server is always a mailbox database or a connector.
- If the source transport server is outside the destination delivery group, the message is relayed along the least-cost routing path to the destination delivery group. Depending on the size and complexity of the Exchange topology, the message is relayed to other transport servers along the least cost routing path, or the message is relayed directly to a transport server in the destination delivery group.

The following types of delivery groups exist in Exchange 2013:

- **Routable DAG** This is a collection of Exchange 2013 Mailbox servers that belong to one DAG. The mailbox databases in the DAG are the routing destinations that are serviced by this delivery group. After the message arrives at the Transport service on a Mailbox server that belongs to the DAG, the Transport service routes the message to the Mailbox Transport service on the Mailbox server in the DAG that currently holds the active copy of the destination mailbox database. The Mailbox Transport service on the destination Mailbox server then delivers the message to the local mailbox database. Although a DAG may contain Mailbox servers located in different Active Directory sites, the DAG is the delivery group boundary.
- **Mailbox delivery group** This is a collection of Exchange servers of the same version located in one Active Directory site. The Active Directory site is the delivery group boundary. The routing destinations and the delivery groups that service them are separated by the major release versions of Exchange in the Active Directory site. The mailbox databases located on Exchange 2010 Mailbox servers are serviced by the Exchange 2010 Hub Transport servers located in the Active Directory site. The mailbox databases located on Exchange 2007 Mailbox servers are serviced by the Exchange 2007 Hub Transport servers located in the Active Directory site. The mailbox databases located on Exchange 2013 Mailbox servers in Active Directory site that don't

belong to a DAG are serviced by the Transport service on Exchange 2013 Mailbox servers in the Active Directory site. How the message is delivered to the mailbox database depends on version of Exchange:

  - **Exchange 2013** After the message arrives at the destination Mailbox server in the destination Active Directory site, the Transport service uses SMTP to transfer the message to the Mailbox Transport service. The Mailbox Transport service then delivers the message to the local mailbox database using RPC.
  - **Exchange 2010 or Exchange 2007** After the message arrives at a random Hub Transport server of the same version in the destination Active Directory site, the store driver on the Hub Transport server uses RPC to write the message to the mailbox database.

- **Connector source servers** This is a mixed collection of Exchange 2010 or Exchange 2007 Hub Transport servers, or Exchange 2013 Mailbox servers that are scoped as the source server for a Send connector, a Delivery Agent connector or a Foreign connector. The connector is the routing destination that's serviced by this routing group. When a connector is scoped to a specific server, only that server is allowed to route messages to destination defined by the connector. This delivery group may contain Exchange 2010 or Exchange 2007 Hub Transport servers, or Exchange 2013 Mailbox servers located in different Active Directory sites.
- **AD site** In some circumstances, an Active Directory site isn't the ultimate destination of a message, but the message must pass through an Exchange 2010 or Exchange 2007 Hub Transport server or Exchange 2013 Mailbox server in that Active Directory site. Those circumstances include:
  - When the Active Directory site is configured as a hub site. When the hub site exists on the least-cost routing path for message delivery, the messages queue and are processed by a transport server in the hub site before they're relayed to their ultimate destination.
  - When an Edge Transport server is subscribed to the Active Directory site. These subscribed Edge Transport servers aren't directly accessible from other Active Directory sites. Note that the Edge Transport server could be Exchange 2013, Exchange 2010 or Exchange 2007.

Note:
Delayed fan-out is only used when the delivery group is an Active Directory site. Delayed fan-out attempts to reduce the number of message transmissions when multiple recipients share any part of the least-cost routing path.

- **Server list** This is a collection of one or more Exchange 2010 or Exchange 2007 Hub Transport servers or Exchange 2013 Mailbox servers that are configured as distribution group expansion servers. The distribution group expansion server is the routing destination serviced by this delivery group.

Delivery group membership isn't mutually exclusive. For example, an Exchange 2013 Mailbox server that's a member of a DAG can also be the source server of a scoped Send connector. This Mailbox server would belong to the routable DAG delivery group for the mailbox databases in the DAG, and also a connector source server delivery group for the scoped Send connector.

The following table maps the routing destinations to the delivery group based on the version of Exchange involved:

	Exchange 2013	**Exchange 2010 or**	**Edge Transport**

	Mailbox server	Exchange 2007 Hub Transport server	server in the perimeter network
Mailbox database in a DAG	Routable DAG	Mailbox delivery group	n/a
Mailbox database not in a DAG	Mailbox delivery group	Mailbox delivery group	n/a
Connector	Connector source servers	Connector source servers	AD site
Distribution group expansion server	Server list	Server list	n/a

Return to top

Queues

From the perspective of the sending server, each delivery queue represents the destination for a particular message. When the Transport service on the Exchange 2013 Mailbox server selects the destination for a message, the destination is stamped on the recipient as the **NextHopSolutionKey** attribute. If a single message is being sent to more than one recipient, each recipient has the **NextHopSolutionKey** attribute. The receiving server also performs message categorization and queues the message for delivery. After a message is queued, you can examine the delivery type for a particular queue to determine whether a message will be relayed again when it reaches the next hop destination. Every unique value of the **NextHopSolutionKey** attribute corresponds to a separate delivery queue.

For more information, see the "NextHopSolutionKey" section in the Queues topic.

Return to top

Routing messages

When a message needs to be delivered to a remote delivery group, a routing path must be determined for the message. Exchange 2013 uses the following logic to select the routing path for a message. This logic is basically unchanged from Exchange 2010:

1. Calculate the least-cost routing path by adding the cost of the IP site links that must be traversed to reach the destination. If the destination is a connector, the cost assigned to the address space is added to the cost to reach the selected connector. If multiple routing paths are possible, the routing path with the lowest aggregate cost is used.
2. If more than one routing path has the same aggregate cost, the number of hops in each path is

evaluated and the routing path with the least number of hops is used.

3. If more than one routing path is still available, the name assigned to the Active Directory sites before the destination is considered. The routing path where the Active Directory site nearest the destination is lowest in alphanumeric order is used. If the site nearest the destination is the same for all routing paths being evaluated, an earlier site name is considered.

In Exchange 2010, each message recipient is always associated with only one Active Directory site, and there is only one least cost routing from the source Active Directory site to the destination Active Directory site. In Exchange 2013, a delivery group may span multiple Active Directory sites, and there may be multiple least-cost routing paths to those multiple Active Directory sites. Exchange 2013 designates a single Active Directory site in the destination delivery group as the *primary site*. The primary site is closest Active Directory site based on the routing logic described earlier. To successfully route messages between delivery groups, Exchange 2013 takes the following issues into consideration:

- **The presence of one or more hub sites along the least-cost routing path** If the least-cost routing path to the primary site contains any hub sites, the message must be routed through the hub sites. The closest hub site along the least-cost routing path is selected as a new delivery group of the type **AD site**, which includes all transport servers in the hub site. After the message traverses the hub site, routing of the message along the least-cost routing path continues. If the primary site happens to be a hub site, the primary site is still considered a hub site for the following reasons:
  - If the destination delivery group spans multiple Active Directory sites, the source server should only attempt to connect to the servers in the hub site.
  - The servers in the hub site that actually belong to the target delivery group are preferred.

As in previous version of Exchange, any hub sites that aren't in the least-cost routing path to the primary site are ignored.

- **The target Exchange server to select in the destination routing group** When the destination delivery group spans multiple Active Directory sites, the routing path to specific servers within the delivery group may have different costs. Servers located in the closest Active Directory site are selected as the target servers for the delivery group based on the least-cost routing path, and the Active Directory site those servers are in is selected as the primary site.
- **Fallback options when connection attempts to all servers in the destination routing group fail** If the destination delivery group spans multiple Active Directory sites, the first fallback option is all other servers in the destination delivery group in other Active Directory sites that aren't selected as target servers. Server selection is made based on the cost of the routing path to those other Active Directory sites. If the destination delivery group has any servers in the local Active Directory site, there are no other fallback options because the message is already as close to the target routing destination as possible. If the destination delivery group has servers in remote Active Directory sites, the option is to try to connect to all other servers in the primary site. If that fails, a backoff path in the least-cost routing path to the primary site is used. Exchange 2013 tries to deliver the message as close to the destination as possible by backing off, hop by hop, along the least-cost routing path until a connection is made.

Return to top

Routing messages between Active Directory sites

The way that Exchange 2013 routes messages between Active Directory sites is virtually the same as Exchange 2010. For more information, see Route mail between Active Directory sites.

Return to top

Routing in the Front End Transport service on Client Access servers

This acts as a stateless proxy for all inbound and (optionally) outbound external SMTP traffic for the Exchange 2013 organization. For outgoing messages, the Transport service uses Send connectors to communicate with the Front End Transport service on a Client Access server. Specifically, outgoing messages are proxied through the Front End Transport service when the *FrontEndProxyEnabled* parameter on an applicable Send connector is set to `$true`, or when the **Proxy through Client Access server** option is selected in the Send Connector properties in the Exchange admin center (EAC). Any Client Access server in the local Active Directory site will be selected. Note that the Front End Transport service doesn't have Send connectors.

For incoming messages, the Front End Transport service must quickly find a single, healthy Transport service on a Mailbox server to receive the message transmission, regardless of the number or type of recipients. Failure to do so results in the email service being perceived as unavailable by the external senders. Like the Transport service, the Front End Transport service loads routing tables based on information from Active Directory, and uses delivery groups to determine how to route messages. However, the routing tables used by the Front End Transport service have the following unique characteristics:

- The Front End Transport service is never considered a member of a delivery group, even when the Mailbox server and the Client access server are installed on the same physical server. This forces the Front End Transport service to communicate with the Transport service only.
- The routing tables don't contain any Send connector routes.
- The routing tables contain a special list of Mailbox servers in the local Active Directory site for fast fail-over purposes.

Routing in the Front End Transport service resolves message recipients to mailbox databases. The list of Mailbox servers used by the Front End Transport service is based on the mailbox databases of the message recipients. Note that it's possible that none of the recipients have mailboxes, for example, if the recipient is a distribution group or a mail user. For each mailbox database, the Front End Transport service looks up the delivery group and the associated routing information. The delivery groups used by the Front End Transport service are:

- Routable DAG
- Mailbox delivery group

- AD site

Depending on the number and type of recipients, the Front End Transport service performs one of the following actions:

- For messages with a single mailbox recipient, select a Mailbox server in the target delivery group, and give preference to the Mailbox server based on the proximity of the Active Directory site. Routing the message to the recipient may involve routing the message through a hub site.
- For messages with multiple mailbox recipients, use the first 20 recipients to select a Mailbox server in the closest delivery group, based on the proximity of the Active Directory site. Note that message bifurcation doesn't occur in Front-End Transport, so only one Mailbox server is ultimately selected, regardless of number of recipients in a message.
- If the message has no mailbox recipients, select a random Mailbox server in the local Active Directory site.

Return to top

Routing in the Mailbox Transport service on Mailbox servers

This consists of two separate services: the Mailbox Transport Submission service and Mailbox Transport Delivery service. For incoming messages, the Mailbox Transport Delivery service receives SMTP messages from the Transport service, and connects to the local mailbox database using RPC to deliver the message. For outgoing messages, the Mailbox Transport Submission service connects to the local mailbox database using RPC to retrieve messages, and submits the messages over SMTP to the Transport service. The Mailbox Transport service is stateless, and doesn't queue any messages locally.

Like the Transport service, the Mailbox Transport service loads routing tables based on information from Active Directory, and uses delivery groups to determine how to route messages. However, there are routing aspects that are unique to the Mailbox Transport service:

- Because the Transport service and the Mailbox Transport service exist on the same Exchange 2013 Mailbox server, the Mailbox Transport service always belongs to the same delivery group as the Mailbox server. This delivery group is referred to as the *local delivery group*.
- The Mailbox Transport Submission service doesn't automatically send messages to the Transport service on the local Mailbox server or on other Mailbox servers in its own local delivery group. The Mailbox Transport Submission service has access to the same routing topology information as the Transport service, so the Mailbox Transport submission service can send messages to the Transport service on Mailbox servers outside the delivery group. The Mailbox servers in the local delivery group are used as fallback options, and for delivery to non-mailbox recipients.
- The Mailbox Transport service only communicates with the Transport service on Exchange 2013 Mailbox servers.
- The Mailbox Transport service only communicates with mailbox databases on the local Exchange 2013 Mailbox server. The Mailbox Transport service never communicates with mailbox databases

on other Mailbox servers.

When a user sends a message from their mailbox, the Mailbox Transport Submission service resolves the message recipients to mailbox databases. The list of Mailbox servers used by the Mailbox Transport Submission service is based on the mailbox databases of the message recipients. Note that it's possible that none of the recipients have mailboxes, for example, if the recipient is a distribution group or a mail user. For each mailbox database, the Mailbox Transport Submission service looks up the delivery group and the associated routing information. The delivery groups used by the Mailbox Transport Submission service are:

- Routable DAG
- Mailbox delivery group
- AD site

Depending on the number and type of recipients, the Mailbox Transport Submission service performs one of the following actions:

- For messages with a single mailbox recipient, select a Mailbox server in the target delivery group, and give preference to the Mailbox server based on the proximity of the Active Directory site. Routing the message to the recipient may involve routing the message through a hub site.
- For messages with multiple mailbox recipients, use the first 20 recipients to select a Mailbox server in the closest delivery group, based on the proximity of the Active Directory site.
- If the message has no mailbox recipients, select a Mailbox server in the local delivery group.

When the Mailbox Transport Delivery service receives a message from the Transport service, it accepts or rejects the message for delivery to a local mailbox database. The Mailbox Transport Delivery service can deliver the message if the recipient resides in an active copy of a local mailbox database. But, if the recipient doesn't reside in an active copy of a local mailbox database, the Mailbox Transport Delivery service can't deliver the message, and must provide a non-delivery response to the Transport service. For example, if the active copy of the mailbox database recently moved to another server, the Transport service might erroneously transmit a message to a Mailbox server that now holds an inactive copy of the mailbox database. The non-delivery responses that the Mailbox Transport Delivery service returns to the Transport service include:

- Retry delivery
- Generate an NDR
- Reroute the message

Return to top

Routing in the Transport service on Edge Transport servers

If you have an Edge Transport server installed in your perimeter network, the Transport service on the Edge Transport server provides SMTP relay and smart host services for all Internet-facing mail flow. Messages that come and go from the Internet are queued locally on the Edge Transport server. The queues correspond to external domains or Send connectors. For more information, see the "NextHopSolutionKey" section in the Queues topic.

Typically, when you install an Edge Transport server in your perimeter network, you subscribe the Edge Transport server to an Active Directory site. The Active Directory site contains the Mailbox servers that will relay messages to and from the Edge Transport server. The Edge Subscription process creates an Active Directory site membership affiliation for the Edge Transport server. The site affiliation enables the Mailbox servers in the Active Directory site to relay messages to the Edge Transport server for delivery to the Internet without having to configure explicit Send connectors.

In multi-site configurations, outbound mail from internal recipients to external recipients is first routed to the subscribed Active Directory site. The target Active Directory site is the delivery group. The routing destination is the intra-organization Send connector in the Transport service on any of the Mailbox servers in the subscribed Active Directory site. The *intra-organization Send connector* is special Send connector that exists in the Transport service on every Mailbox server. This Send connector is implicitly created, invisible, requires no management, and is used to relay messages between Exchange servers.

Outbound mail for external recipients is routed from the Mailbox server to the Edge Transport server. The Client Access server is not involved in routing mail to a subscribed Edge Transport server. Mail is transmitted from the intra-organization Send connector in the Transport service on the Mailbox server to a Receive connector in the Transport service on the Edge Transport server. On a subscribed Edge Transport server, the default Receive connector is configured to listen for connections from internal Mailbox servers in the subscribed Active Directory site and anonymous connections from the Internet. After the message is categorized by the Transport service on the Edge Transport server, the message is queued locally for delivery to the Internet by using the dedicated Send connector that's created during the Edge Subscription.

Inbound mail from external recipients arrives on the Edge Transport through the default Receive connector, and the messages are categorized and queued for delivery. The messages are relayed through the dedicated Send connector that's created by the Edge Subscription to send mail into the Exchange organization. Where the messages go next depends on how the internal Exchange servers are configured.

- **Mailbox server and Client Access server installed on the same computer** In this configuration, the Client Access server is used for inbound mail flow. Mail flows from the Send connector in the Transport service on the Edge Transport server to the default Receive connector in the Front End Transport service on the Client Access server, and then to the default Receive connector in the Transport service on the Mailbox server.
- **Mailbox server and Client Access server installed on different computers** In this configuration, the Client Access server is bypassed for inbound mail flow. Mail flows from the Send connector in the Transport service on the Edge Transport server to the default Receive connector in the Transport service on the Mailbox server.

If you have an Exchange 2007 or Exchange 2010 Edge Transport server installed in the perimeter network, inbound and outbound mail flow always occurs directly between the Edge Transport server and the Mailbox server. The Client Access server isn't used.

Return to top

Planning to use Active Directory sites for routing mail

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-05-21*

Microsoft Exchange Server 2013 recognizes Active Directory sites and database availability groups (DAGs) as routing boundaries. However, Exchange 2013 still uses Active Directory site topology to determine how messages are transported between Exchange servers in different DAGs or sites within the organization. For more information, see Mail routing.

The Transport service on a Mailbox server provides message transport inside the Exchange organization. When you're deploying a pure Exchange 2013 organization, or introducing Exchange 2013 into a pure Exchange Server 2010 organization, no additional configuration is required to establish routing in the forest.

Contents

How Exchange 2013 uses Active Directory site membership

Exchange 2013 is a site-aware application. Site-aware applications can determine their own Active Directory site membership and the site membership of other servers by querying Active Directory. Exchange 2013 uses site membership to determine which domain controllers and global catalog servers to use for processing Active Directory queries. Additionally, when a server running Exchange has to determine the Active Directory site membership of another Exchange server, it can query Active Directory to retrieve the site name.

In Exchange 2013, the Microsoft Exchange Active Directory Topology service is responsible for updating the site attribute of the Exchange server object. Because the Active Directory site membership is a server object attribute, Exchange doesn't have to query DNS to resolve a server address to a subnet associated with an Active Directory site. Stamping the Active Directory site

attribute on an Exchange server object also enables Active Directory site membership to be assigned to a server that isn't a domain member, such as a subscribed Edge Transport server.

The Exchange 2013 servers use Active Directory site membership information as follows:

- **Mail submission** The Mailbox Transport Submission service on an Exchange 2013 Mailbox server uses Active Directory site membership information to determine the other Exchange 2013 Mailbox servers that are located in the same Active Directory site. If the source Mailbox server doesn't belong to a DAG, or if the DAG doesn't span multiple Active Directory sites, the Mailbox Transport Submission service on the source Mailbox server submits messages for routing and transport to the Transport service on an Exchange 2013 Mailbox server in the same Active Directory site.
- **Mail delivery** The Transport service on an Exchange 2013 Mailbox server performs recipient resolution and queries Active Directory to match an email address to a recipient account. The recipient account information includes the fully qualified domain name (FQDN) of the user's mailbox database. The Transport service queries Active Directory to determine the Active Directory site of the user's mailbox database. If the mailbox database doesn't belong to a DAG, it will deliver the message to that Mailbox server. Otherwise, it will relay the message to another Mailbox server in the same site as the target Mailbox server for delivery.
- **Message routing** The Transport service on Exchange 2013 Mailbox servers retrieves information from Active Directory to determine how mail should be routed inside the organization. Exchange 2013 uses the concept of *delivery groups* to determine where and how to route messages. Depending on the destination, the message could be routed to the Transport service or the Mailbox Transport Delivery service on an Exchange 2013 Mailbox server, or to a Hub Transport server that's running a previous version of Exchange. For more information, see Mail routing.
- **Unified Messaging message submission** The Unified Messaging service on Exchange 2013 Mailbox servers uses Active Directory site membership information to find the other Mailbox servers that are located in the same Active Directory site. The Unified Messaging service submits messages for routing to the Transport service on a Mailbox server within the same Active Directory site. The Transport service server performs recipient resolution and queries Active Directory to match a telephone number, E.164 number, or a SIP address to a recipient account. After the recipient resolution completes, the Transport service delivers the message to the target mailbox in the same way as a regular email message.
- **Client connections to Client Access server** When the Client Access server receives a user connection request, it queries Active Directory to determine which Mailbox server is hosting the user's mailbox. The Client Access server then retrieves the Active Directory site membership of that Mailbox server and proxies the connection to the Mailbox server.

Determine Active Directory site membership

Active Directory clients assume site membership by matching their assigned IP address to a subnet defined in Active Directory Sites and Services and associated with an Active Directory site. The client then uses this information to determine which domain controllers and global catalog servers exist in that site and communicates with those directory servers for authentication and

authorization purposes. Exchange 2013 takes advantage of this relationship by also preferring to retrieve information about recipients from directory servers in the same site as the Exchange 2013 server.

All computers that are part of the same Active Directory site are considered well connected, with a high-speed, reliable network connection. By default, when an Active Directory forest is first deployed, there's a single site named `Default-First-Site-Name`. If no other sites are manually configured by the administrator, all server and client computers in the forest are considered members of `Default-First-Site-Name`.

When more than one site is defined, the Active Directory administrator must define the subnets present in the organization and associate those subnets with Active Directory sites.

The Microsoft Exchange Active Directory Topology service checks the site membership attribute on the Exchange server object when the server starts. If the site attribute has to be updated, the Microsoft Exchange Active Directory Topology service stamps the attribute with the new value. The Microsoft Exchange Active Directory Topology service verifies the site attribute value every 15 minutes and updates the value if site membership has changed. The Microsoft Exchange Active Directory Topology service uses the Net Logon service to obtain current site membership. The Net Logon service updates site membership every five minutes. This means that up to a 20 minute latency period may pass between the time that site membership changes and the new value is stamped on the site attribute.

Overview of IP site links

Relationships between Active Directory sites are defined by IP site links. The IP site link consists of two or more Active Directory sites. All Active Directory sites that are part of the link communicate at the same cost. The IP site link properties include a cost assignment, a schedule, and an interval. The schedule and interval properties are only used for determining Active Directory replication frequency. Exchange 2013 uses the cost assignment to determine the lowest cost route for traffic to follow when multiple paths exist to the destination. The cost of the route is determined by aggregating the cost of all site links in a transmission path. The Active Directory administrator assigns the cost to a link based on relative network speed and available bandwidth compared to other available connections.

By default, the Transport service on a Mailbox server always tries a direct connection to the Transport service or the Mailbox Transport Delivery service on a Mailbox server in another Active Directory site. Messages in transport don't relay through the Transport service on each Mailbox server in a site link path. However, Mailbox servers in intermediate Active Directory sites along the routing path may perform message relay in the following scenarios:

- Direct relay between Mailbox servers won't occur when a hub site exists along the least cost routing path. You can configure an Active Directory site as a hub site so that messages are routed through the hub site before the messages are relayed to the target server. Hub sites are discussed later in this topic.

- Exchange 2013 uses the routing path derived from IP site link information when communication to the destination Active Directory site fails. If no Mailbox server in the destination Active Directory site responds, message delivery backs off along the least cost routing path until a connection is made to a Mailbox server in an Active Directory site along the routing path. The messages are queued in that Active Directory site and the queue will be in a retry state. This behavior is called *queue at point of failure*.
- In Exchange 2013 organizations without DAGs, the Transport service on a Mailbox server can also use the IP site link information to optimize routing of messages sent to multiple recipients. The Mailbox server delays bifurcation of messages until it reaches a fork in the routing paths to the recipients. The bifurcated message is relayed to each recipient destination by a Mailbox server in the Active Directory site that represents the fork in the individual routing paths. This functionality is called *delayed fan-out*.

Designate hub sites

You can use the **Set-AdSite** cmdlet to configure an Active Directory site as a hub site. When a hub site exists along the least cost routing path between two transport servers, messages are routed through the hub site for processing before they are relayed to the destination server. For this routing behavior to occur, the hub site must exist along the least cost routing path between two transport servers. This configuration should only be used when it's required by the network topology, such as when firewalls exist between Active Directory sites and prevent direct relay of SMTP communications. For more information, see Configure Exchange mail routing settings in Active Directory.

Set an Exchange-specific cost on an IP site link

You can use the **Set-AdSiteLink** cmdlet in the Exchange Management Shell to configure an Exchange-specific cost to an Active Directory IP site link. The Exchange-specific cost is a separate attribute used instead of the Active Directory-assigned cost to determine the Exchange routing path. This configuration is useful when the Active Directory IP site link costs don't result in an optimal Exchange message routing topology. For more information, see Configure Exchange mail routing settings in Active Directory.

Set message size restrictions on IP site links

By default, Exchange 2013 doesn't impose a maximum message size limit on messages relayed between Mailbox servers in different Active Directory sites. If you use the **Set-AdSiteLink** cmdlet to configure a maximum message size on an Active Directory IP site link, routing generates a non-delivery report (NDR) for any message that has a size larger than the maximum message size limit configured on any Active Directory site link in the least cost routing path. This configuration is useful for restricting the size of messages sent to remote Active Directory sites that must communicate over low-bandwidth connections.

Exchange 2013 server placement in Active Directory sites

For message routing between Exchange 2013 servers to occur correctly, all Exchange servers deployed in the forest must belong to an Active Directory site. Make sure that the IP addresses that you have assigned are in subnets that are correctly associated with Active Directory sites.

The first step in planning the placement of Exchange 2013 servers in the Active Directory site topology is to document the current topology. Your documentation should include the following:

- Sites
- Subnets and their site association
- IP site links and their member sites
- IP site link costs
- Directory servers in each site
- Physical network connections
- Firewall locations

After you have diagrammed these objects, plan the placement of Exchange servers. Consider the following information when deciding where to put servers:

- A Mailbox server needs to communicate directly with a global catalog server to perform Active Directory lookups.
- We recommend that you deploy more than one Mailbox server in each Active Directory site to provide load balancing and fault tolerance.
- DAGs and site resilience
- The Unified Messaging service on Mailbox servers submits voice mail messages to the Transport service on Mailbox servers for delivery to mailboxes. The Client Access server that's running the Unified Messaging Call Router service may be located in a hub site or near the IP or Voice over IP (VoIP) gateway, IP Private Branch eXchange (IP PBX), SIP-enabled PBX, or session border controllers (SBC). The Transport service on a Mailbox server that has the same site membership as the Client Access server will receive voice mail messages for transport and route the messages to the Transport service on other Mailbox servers in the organization.
- Client Access servers provide a connectivity point to the Exchange organization for users who are accessing Exchange remotely. A Client Access server must be deployed in each Active Directory site that contains Mailbox servers.

After you plan your Exchange 2013 server placement, you may identify areas where you can modify the Active Directory site topology to improve communication flow. You may want to adjust IP site links and site link costs to optimize mail delivery. An efficient Active Directory topology doesn't require any changes to support Exchange 2013

Configure Exchange mail routing

settings in Active Directory

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

By default Microsoft Exchange Server 2013 references the IP site link objects in Active Directory to help determine the least-cost routing path. However, if you determine the Active Directory IP site link costs and traffic flow patterns aren't optimal for mail routing in Exchange, you can configure settings in Active Directory that are only used by Exchange to help optimize mail flow.

What do you need to know before you begin?

- Estimated time to complete each procedure: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Active Directory site and site link management" entry in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to configure an Exchange-specific cost on an Active Directory IP site link

Determine the name of the Active Directory IP site link for which you want to set an Exchange cost. A lower cost value indicates a more preferred route. You can examine the contents of the routing table logs and view the data in the **ADTopologyPath ID** section to view details about the calculated least cost routing path between two Active Directory sites.

To set an Exchange-specific cost on an Active Directory site link, run the following command:

```
Set-AdSiteLink <ADSiteLinkIdentity> -ExchangeCost <Integer
| $null>
```

This example sets an Exchange-specific cost of 10 on the IP site link named IPSiteLinkAB.

```
Set-AdSiteLink IPSiteLinkAB -ExchangeCost 10
```

This example clears the Exchange cost from the IP site link named IPSiteLinkAB.

```
Set-AdSiteLink IPSiteLinkAB -ExchangeCost $null
```

How do you know this worked?

To verify that you have successfully set an Exchange cost on an Active Directory site link, do the following:
1. Run the following command:

```
Get-AdSiteLink | Format-List Name,ExchangeCost
```

2. Verify the Exchange cost is configured on the Active Directory site link.

Use the Shell to configure an Active Directory site as a hub site

When a hub site exists along the least cost routing path for a message, the message must be routed through the hub site. Examine the contents of the routing table logs and view the data in the **ADTopologyPath ID** section to verify that the selected site exists along the least cost routing path between two Active Directory sites. If this isn't the case, you need to assign Exchange-specific costs to the IP site links to make the least cost routing path go through the selected sites.

To configure an Active Directory site as a hub site, run the following command:

```
Set-AdSite <ADSiteIdentity> -HubSiteEnabled $true
```

This example configures the Active Directory site named Site A as a hub site.

```
Set-AdSite "Site A" -HubSiteEnabled $true
```

This example removes the hub site attribute from the Active Directory site named Site B.

```
Set-AdSite "Site B" -HubSiteEnabled $false
```

How do you know this worked?

To verify that you have successfully configured an Active Directory site as a hub site, do the following:
1. Run the following command:

```
Get-AdSite | Format-List Name,HubSiteEnabled
```

2. Verify the *HubSiteEnabled* value is `True` for the Active Directory site.

Route mail between Active Directory sites

Topic Last Modified: *2012-12-10*

An Active Directory site is a logical configuration component that's based on the physical aspects of the network. The primary purpose for creating an Active Directory site is to define which subnets in the network are connected in a way that optimizes control of Active Directory replication traffic. Microsoft Exchange Server 2013 recognizes both database availability groups (DAGs) and Active Directory sites as routing boundaries, and Exchange 2013 servers make routing decisions based on the Active Directory site topology.

By default, an Active Directory forest contains only one Active Directory site. The default name for this Active Directory site is `Default-First-Site-Name`. If no other Active Directory sites are created, all domain member computers in the forest are members of `Default-First-Site-Name`. You don't have to configure a subnet-to-site association. If additional Active Directory sites are created, you need specify the subnets that are assigned to that Active Directory site.

Contents

Determining site membership

Each Active Directory site is associated with one or more IP subnets. An administrator assigns Active Directory site membership to computers that are configured as domain controllers and global catalog servers. Other domain member computers, such as Exchange servers, are assigned Active Directory site membership automatically when they're configured to use an IP address that's in an IP subnet that's associated with an Active Directory site. Computers that have the same Active Directory site membership are presumed to have good network connectivity. A server is always a member of a single Active Directory site.

When an application can determine the Active Directory site membership of the computer where it's installed and of other computers in the forest, and then use that information to control communication flow, it's a site-aware application. When site-aware applications must use the

services of another server, such as a domain controller or global catalog server, priority is given to the servers that have the same Active Directory site membership as the computer that's requesting those services.

Exchange 2013 is a site-aware application and uses the Active Directory topology for message routing and to communicate with the services that are running on other Exchange 2013 computers. The Active Directory site isn't only a routing boundary. It's also a service discovery boundary.

Determining site membership for a domain member computer depends on a series of DNS queries to compare the local IP address to defined subnets and then determine the appropriate site membership association. To reduce the overhead that's associated with DNS queries, the Exchange 2013 Active Directory schema additions include the **msExchServerSite** attribute for the Exchange server object. The value of this attribute is the distinguished name of the Active Directory site of an Exchange server. This attribute is a property of each Exchange server object. When site membership affinity is stored as an attribute of the server object, the current topology can be read directly from Active Directory instead of relying on DNS queries and a site membership association is enabled for a non-domain computer, such as a subscribed Edge Transport server.

The value for the **msExchServerSite** attribute is populated and kept up to date by the Microsoft Exchange Active Directory Topology service. When a Windows-based computer starts, the Net Logon service determines site membership for the computer. The Net Logon service uses that information to locate domain controllers that are located in the same Active Directory site as the local computer and then directs authorization and authentication requests to those servers. The Microsoft Exchange Active Directory Topology service uses the **DsGetSiteName** API call to retrieve the site membership value from the Net Logon service and writes the Active Directory site's distinguished name to the **msExchServerSite** attribute for the Exchange server object in Active Directory.

The following table shows how an organization might define Active Directory sites. In this example, three Active Directory sites are defined, and each Active Directory site is associated with more than one IP subnet.

Example of an Active Directory site-to-subnet association

Active Directory site name	Associated IP subnets
Site A	192.168.1.0/24 192.168.2.0/24
Site B	192.168.3.0/24 192.168.4.0/24
Site C	192.168.5.0/24 192.168.6.0/24

If a server named Mailbox01 has the IP address of 192.168.1.1, it's a member of Site A. By changing the IP address of a server, you may change its site membership. If you change the IP address of Mailbox01 to 192.168.2.1, it won't change the server's Active Directory site membership because that subnet is also associated with Site A. However, if you move the server and the IP address changes to 192.168.3.1, the server would be considered a member of Site B.

A change in site membership can also occur if you change the association of subnets to Active Directory sites. For example, if you remove the subnet 192.168.3.0 from association with Site B and associate it with Site A, the site membership of a server that has the IP address of 192.168.3.1 also changes to Site A. Whenever a change in site membership occurs, Exchange must update its configuration data so that the change is considered when Exchange makes routing decisions. Some latency occurs between the time that a change in an Active Directory site membership occurs and the topology change is fully propagated.

Return to top

IP site links

Site links are logical paths between Active Directory sites. A site link object represents a set of sites that can communicate at a uniform cost. Site links don't correspond to the actual path taken by network packets on the physical network. However, the cost assigned to the site link by the administrator typically relates to the underlying network reliability, speed, and available bandwidth. For example, the Active Directory administrator would assign a lower cost to a network connection with a speed of 100 megabits per second (Mbps) than to a network connection with a speed of 10 Mbps.

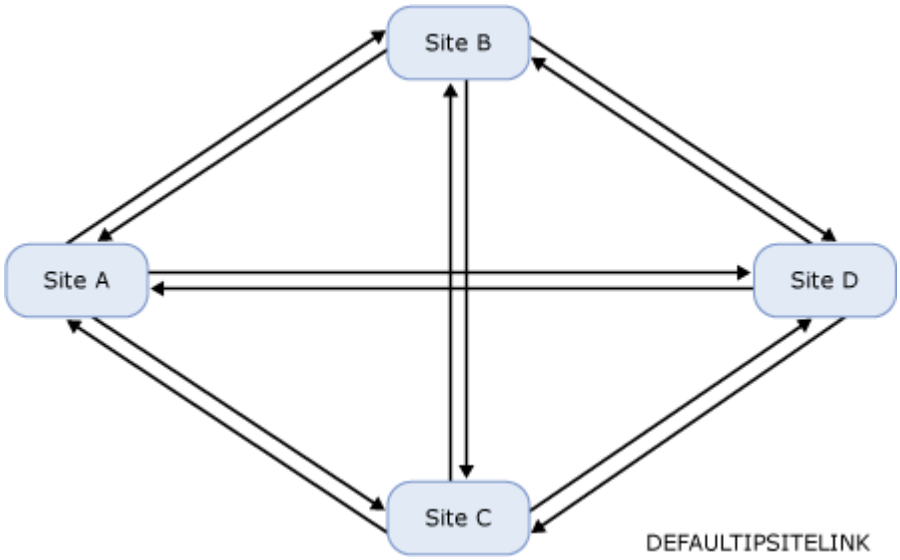
By default, all site links are transitive. This means that if Site A has a link to Site B, and Site B has a link to Site C, Site A is transitively linked to Site C. The transitive link between Site A and Site C is also known as a *site-link bridge*.

Exchange uses only IP site links to determine its Active Directory site routing topology. The cost that's assigned to the IP site link will be considered by the routing component of Exchange when calculating a routing table. These costs are used to calculate the least-cost routing path to the ultimate destination for a message.

Every Active Directory site must be associated with at least one IP site link. There is a single default IP site link named `DEFAULTIPSITELINK`. When you create an Active Directory site, you need to associate that site to an IP site link. You can create additional IP site links to implement the desired topology, or you can associate every Active Directory site to the `DEFAULTIPSITELINK`. Each Active Directory site that's part of an IP site link can communicate directly with every other site in that link at a uniform cost.

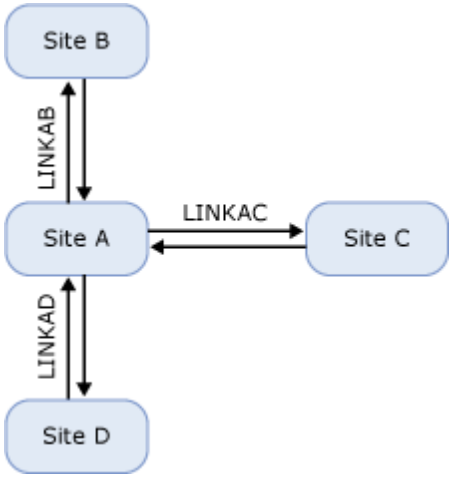
In the following figure, four Active Directory sites are configured in the forest. Every site has been associated with the `DEFAULTIPSITELINK`. Therefore, each Active Directory site communicates directly with every other site by using the same cost metric. More than one communication path is indicated, but only a single IP site link is defined.

Full mesh topology with a single IP site link



In the following figure, four Active Directory sites are configured in the forest. In this topology, the administrator has configured IP site links to create a *hub-and-spoke topology* of Active Directory sites. Each spoke site can communicate directly with the central site, and the spoke sites can communicate with one another by using the transitive IP site links.

Hub-and-spoke topology of Active Directory IP site links



It's important to note that Exchange uses site links when determining the least-cost path, but will always try to deliver messages directly to the destination Exchange server. For example, if a user in Site B in the topology shown in the preceding figure sends a message to another user in Site C, the Mailbox server in Site B will connect directly to the Mailbox server in Site C. If you want to force messages to go through Site A, you must enable that site as a hub site. For more information about hub sites, see "Implementing Hub Sites" later in this topic.

An Active Directory administrator implements the topology that best represents the connectivity and communication requirements of the forest. Because the same topology is used by Exchange, you need to make sure that the current topology supports efficient messaging communication.

The default cost for a site link is 100. A valid site link cost can be any number from 1 through 99,999. If you specify redundant links, the link with the lowest cost assignment is always preferred. An Exchange organization administrator can assign an Exchange-specific cost to an IP site link. If an

Exchange cost is assigned to an IP site link, it will be used by Exchange. Otherwise, the Active Directory cost is used. For more information about how to set an Exchange cost on an IP site link, see "Controlling IP Site Link Costs" later in this topic. An administrator who has membership in the Enterprise Administrators group can create additional IP site links.

For more information about Active Directory site configuration, see Designing the Site Topology.

Return to top

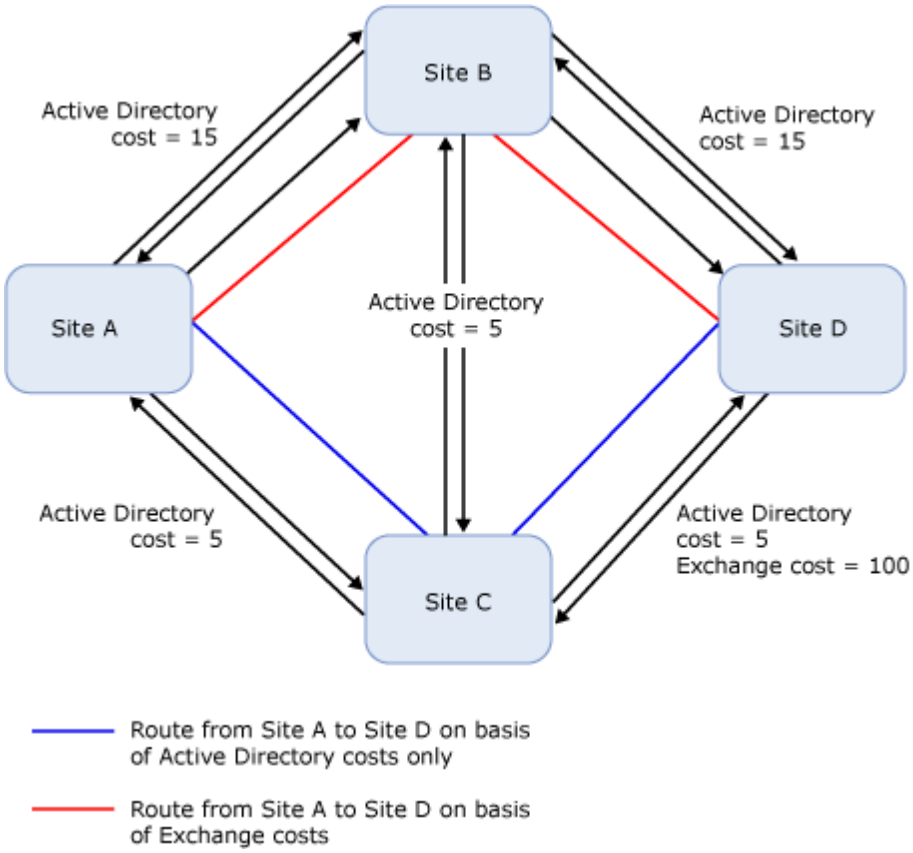
Controlling IP site link costs

Active Directory IP site links costs are based on relative network speed compared to all network connections in the WAN and are designed to produce a reliable and efficient replication topology. Therefore, in most cases, the existing IP site link costs should work well for Exchange message routing. However, if after documenting the existing Active Directory site and IP site link topology, you verify that the Active Directory IP site link costs and traffic flow patterns aren't optimal for Exchange, you can make adjustments to the costs evaluated by Exchange. Changing the cost assigned to the IP site link by using Active Directory tools would impact the entire environment. Instead, use the **Set-AdSiteLink** cmdlet in the Exchange Management Shell to assign an Exchange-specific cost to the IP site link. For example, to configure the Exchange-specific cost value of 25 on the IP site link named SITELINKAB, run the following command in the Shell: `Set-AdSiteLink SITELINKAB -ExchangeCost 25`.

When an Exchange-specific cost is assigned to an IP site link, the Exchange cost overrides the Active Directory cost for message routing purposes, and routing only considers the Exchange cost when it evaluates the least-cost routing path.

Adjusting IP site link costs can be useful when the message routing topology has to diverge from the Active Directory replication topology. Exchange costs can be used to force all message routes to use a hub site. Exchange costs can also be used to control where messages are queued when communication to an Active Directory site fails. The following figure shows an Active Directory topology with four sites.

Topology with Exchange costs configured on IP site links



In the preceding figure, the network connection between Site C and Site D is a low bandwidth connection that's only used for Active Directory replication and shouldn't be used for message routing. However, the Active Directory IP site link costs cause that link to be included in the least-cost routing path from any other Active Directory site to Site D. Therefore, messages are delivered to the Site D queue in Site C. The Exchange administrator prefers that the least-cost routing path include Site B instead so that if Site D is unavailable, the messages will queue at Site B. Configuring a high Exchange cost on the IP site link between Site C and Site D prevents that IP site link from being included in the least-cost routing path to Site D.

Exchange provides support for configuration of a maximum message size limit on an Active Directory IP site link. By default, Exchange doesn't impose a maximum message size limit on messages that are relayed between Exchange servers in different Active Directory sites. If you use the **Set-AdSiteLink** cmdlet to configure a maximum message size on an Active Directory IP site link, routing generates a non-delivery report (NDR) for any message that has a size larger than the maximum message size limit that's configured on any Active Directory site link in the least-cost routing path. This configuration is useful for restricting the size of messages that are sent to remote Active Directory sites that must communicate over low-bandwidth connections. For more information, see Message size limits.

Return to top

Implementing hub sites

In your Exchange organization, you may want to force all message delivery through a specific Active Directory site. You can use the Shell to designate an Active Directory site as a hub site. When

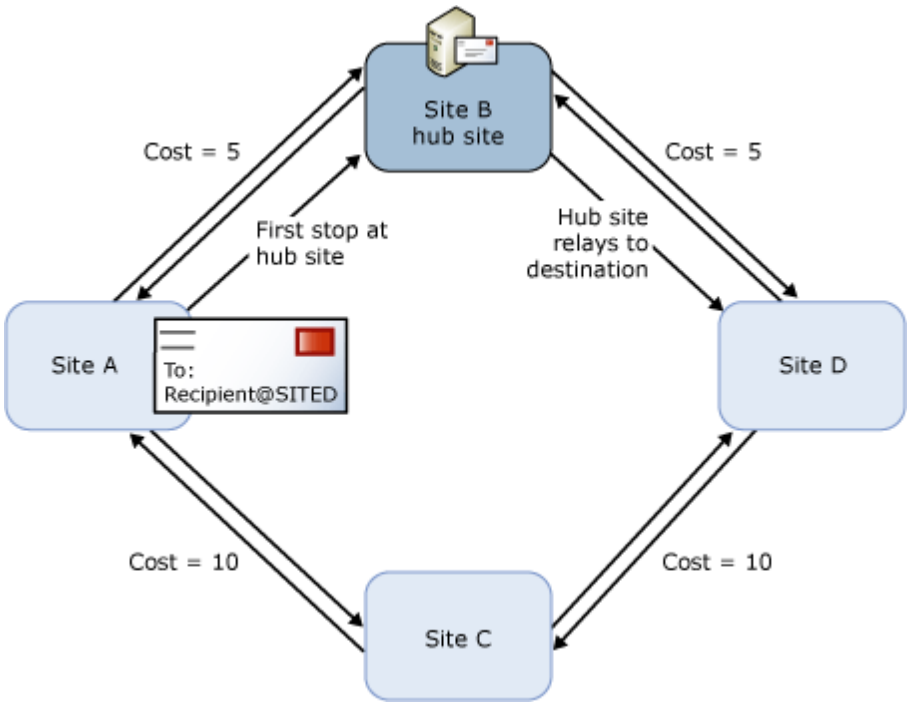
you do this, you cause additional overall overhead because more servers are involved in message delivery. For example, consider a message that's sent from Site A to Site E. If the least-cost routing path is Site A-Site B-Site C-Site D-Site E, and you designate Site C as a hub site, the message is relayed from Site A to Site C and then relayed from Site C to Site E.

You use the **Set-AdSite** cmdlet to specify an Active Directory site as a hub site. Whenever a hub site exists along the least-cost routing path for message delivery, the messages are queued and are processed by the Transport service on Mailbox servers in the hub site before they're relayed to their ultimate destination.

After the least-cost routing path is chosen, routing determines whether there's a hub site along that routing path. If a hub site is configured, messages stop at a Mailbox server in the hub site before they're relayed to the target destination. If there's more than one hub site along the least-cost routing path, messages stop at each hub site along the routing path.

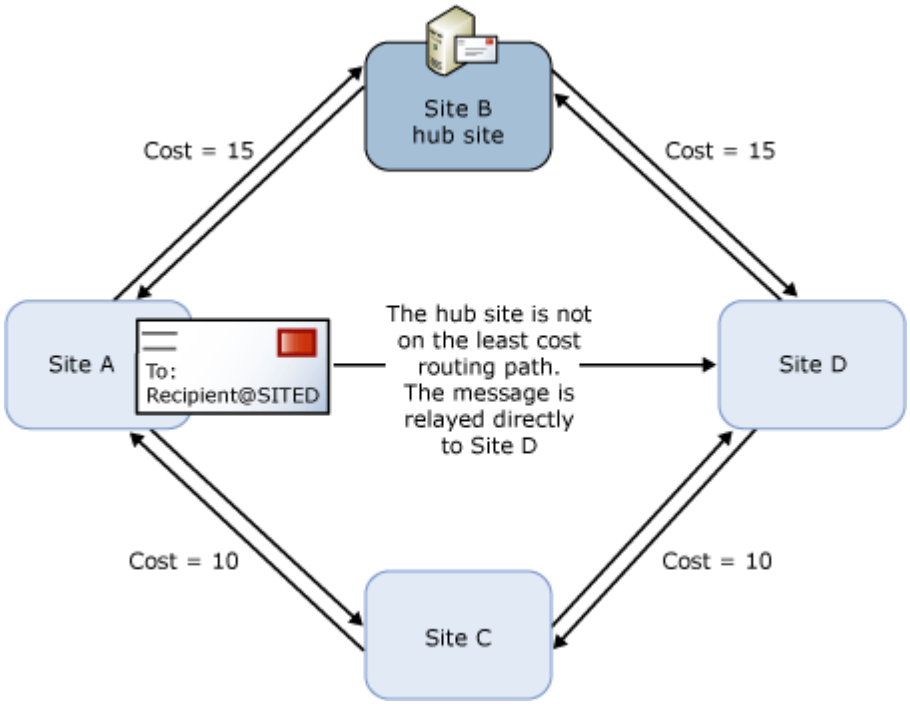
This variation to direct relay routing only is in effect when the hub site is located along the least-cost routing path. The following figure shows the correct use of a hub site. In this diagram, Site B is configured as a hub site. Messages that are relayed from Site A to Site D are relayed through Site B before they're delivered to Site D.

Message delivery with a hub site



The following figure shows how IP site link costs affect routing to a hub site. In this scenario, Site B has been designated as a hub site. However, Site B doesn't exist along the least-cost routing path between any other sites. Therefore, messages that are relayed from Site A to Site D are never relayed through Site B. An Active Directory site is never used as a hub site if it isn't on the least-cost routing path between two other sites.

Misconfigured hub site



You can configure any Active Directory site as a hub site. However, for this configuration to work correctly, you must have at least one Mailbox server in the hub site.

Return to top

Topology discovery

The Active Directory topology is made available to Exchange by the following required elements:

- The Microsoft Exchange Active Directory Topology service.
- The topology discovery module inside the Microsoft Exchange Transport service.

The Microsoft Exchange Active Directory Topology service runs on all Exchange 2013 Client Access servers and Mailbox servers. These servers use the Microsoft Exchange Active Directory Topology service to discover the domain controllers and global catalog servers that can be used by the Exchange servers to read and write Active Directory data. Exchange 2013 binds to the identified directory servers whenever Exchange has to read from or write to Active Directory.

The topology discovery module is part of the Microsoft Exchange Transport service and provides information about the Active Directory topology to Exchange servers. This API discovers the Exchange servers and roles in the organization and determines their relationship to the Active Directory configuration objects. Configuration data is retrieved from Active Directory and then cached so that it can be accessed by the Exchange services that are running on that computer.

The topology discovery module performs the following steps to generate an Exchange routing topology:

1. Data is read from Active Directory. All the following objects are retrieved:
   - Active Directory sites.
   - IP site links.
   - All Exchange servers.
2. The data that's retrieved in step 1 is used to create the initial topology and to begin linking and

mapping the related configuration objects.
3. Exchange servers are matched to Active Directory sites by retrieving the site attribute value from the Exchange server object that's stored in Active Directory.
4. Routing tables are updated with the collection of information retrieved.

This process makes every Exchange 2013 server aware of the other Exchange servers in the organization and of how close the Exchange servers are to one another.

Return to top

Recipient resolution

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-22*

Recipient resolution is the process of expanding and resolving all the recipients in a message. The act of resolving the recipients matches a recipient to the corresponding Active Directory object in the Microsoft Exchange organization. The act of expanding the recipients expands all distribution groups into a list of individual recipients. Recipient resolution allows message limits and alternative recipients to be applied correctly to each recipient.

In a Microsoft Exchange Server 2013 organization, recipient resolution is performed by the categorizer in the Transport service on Mailbox servers. Categorization on each message happens after a newly arrived message is put in the Submission queue. Recipient resolution, in addition to content conversion and routing, is performed on the message before the message is put in a delivery queue. The categorizer performs recipient resolution before routing. The component of the categorizer that's responsible for recipient resolution is frequently called the *resolver*.

Contents

Top-level resolution

Top-level resolution is the first stage of recipient resolution. Top-level resolution associates each recipient in an incoming message to a matching recipient object in Active Directory. During top-level resolution, the categorizer creates a list that contains the sender and the initial, unexpanded

recipient email addresses that exist within the message. The categorizer then uses that list of email addresses to query Active Directory to find any mail-enabled objects that have matching email address attributes. When a match is found, the properties of matching Active Directory objects are cached for later use. Any sender message restrictions are also enforced.

Recipient email addresses

Top-level resolution begins with a message and the initial, unexpanded list of recipients from the *message envelope*. The message envelope contains the commands that are used to transmit messages among SMTP messaging servers. The sender's email address is contained in the **MAIL FROM:** command. Each recipient's email address is contained in a separate **RCPT TO:** command. The envelope sender and envelope recipients are typically created from the sender and recipients in the `To:`, `From:`, `Cc:`, and `Bcc:` header fields in the message header. However, this isn't always true. The `To:`, `From:`, `Cc:`, and `Bcc:` header fields in a message are easily forged and may not match the actual sender or recipient email addresses that were used to transmit the message.

Encapsulated email addresses

Standard SMTP email addresses follow the specifications of RFC 2821 and RFC 2822, such as chris@contoso.com, for example. However, an email address can also be a non-SMTP email address that's encapsulated inside a valid SMTP address. Exchange supports encapsulated addresses that use the Internet Mail Connector Encapsulated Address (IMCEA) encapsulation method.

This encapsulation method requires the encoding of any characters that are invalid in SMTP email addresses. Alphanumeric characters, the equal sign (=) and the hyphen (-) don't require encoding. Other characters use the following encoding syntax:

- A forward slash (/) is replaced by an underscore (_).
- Other US-ASCII characters are replaced by a plus sign (+) and the two digits of its ASCII value are written in hexadecimal. For example, the space character has the encoded value `+20`.

The IMCEA encapsulation method uses the following syntax: `IMCEA<Type>-<address>@<domain>`

The placeholder <*Type*> identifies the type of non-SMTP address, for example `EX`, `X400`, or `FAX`.

Note:

Although `SMTP` and `X500` are theoretically valid values for <*Type*>, Exchange recipient resolution rejects any IMCEA-encoded addresses that use either of these types.

The placeholder <*address*> is the encoded original address. The placeholder <*domain*> represents the SMTP domain that's used to encapsulate the non-SMTP address, for example, contoso.com

With the IMCEA encapsulation method, addresses are unencapsulated only when the domain matches the default authoritative domain in the Exchange organization. For more information about accepted domains, see Accepted domains.

The maximum length for an SMTP email address in Exchange is 571 characters. This limit includes the following:

- 315 characters for the name part of the address
- 255 characters for the domain name
- The at sign (@) character that separates the name part of the address from the domain name

Note that Exchange doesn't support messages that are encoded with the IMCEA encapsulation method when the name part of the address exceeds 315 characters, even if the complete email address is less than 571 characters.

Address resolution

For each message, the sender email address and all recipient email addresses are added to a list that's used to query Active Directory. Any encapsulated addresses are unencapsulated before they're added to the list of email addresses. The Active Directory query is performed on up to 20 email addresses at a time. If the Active Directory query encounters any transient errors, the message is returned to the Submission queue and deferred for the time that's specified by the *ResolverRetryInterval* key in the `%ExchangeInstallPath%Bin\EdgeTransport.exe.config` XML application configuration file that's associated with the Microsoft Exchange Transport service. The default value is 30 minutes.

The following table describes the recipient objects that are found in Active Directory. For more information about Exchange recipient types, see Recipients.

Recipient objects in Active Directory

Active Directory recipient type	Description
DistributionGroup	Any mail-enabled group object. The distribution group object types are as follows: • **MailUniversalDistributionGroup** A universal distribution group object • **MailUniversalSecurityGroup** A universal security group (USG) object that has an email address • **MailNonUniversalGroup** A local security group object or global security group object that has an email address
DynamicDistributionGroup	An object that has the Active Directory class **msExchDynamicDistributionList**. For more information, see Manage dynamic distribution groups.

Mailbox	A user object that has an email address and a defined *Database* parameter
MailUser	A user object that has an email address without a defined *Database* parameter. For more information, see Manage mail users.
MailContact	A contact object that has an email address. Typically, a mail contact is used for recipients outside the Exchange organization. A mail contact is also used in cross-forest Exchange environments. For more information, see Manage mail contacts.
MailPublicFolder	A public folder object that has an email address.
MicrosoftExchangeRecipient	An object that has the Active Directory class **msExchExchangeServerRecipient**. For more information about the Microsoft Exchange recipient object, see Recipients.
SystemAttendantMailbox	An object that has the Active Directory class **exchangeAdminService**. There should be only one system attendant mailbox in the Exchange organization.
SystemMailbox	A user object that has an email address and that's located in the Microsoft Exchange System Objects container. There should be one system mailbox for each mailbox database in the Exchange organization.

An object that contains missing or malformed critical properties is classified by the Active Directory query as an invalid object. For example, a dynamic distribution group object without an email address is considered invalid. Messages that are sent to recipients that are classified as invalid objects generate a non-delivery report (NDR).

For each email address, a single initial query is performed for all possible recipient properties, such

as the recipient identifiers, recipient type, message limits, email addresses, and alternative recipients. The applicable properties for the recipient are cached for later use. Recipient resolution classifies the recipients based on similarities in how the recipients are resolved, and the similarity of the applicable recipient properties.

The LDAP filter that's used for address resolution is described as follows:

- For the **EX** email address type, the LDAP filter is based on the recipient **legacyExchangeDN** Active Directory attribute or the recipient **proxyAddresses** Active Directory attribute. The **legacyExchangeDN** Active Directory attribute takes precedence.
- For all other email addresses types, the recipient **proxyAddresses** Active Directory attribute is used as the LDAP filter.

If the email address that's used in the message doesn't match the primary SMTP address of the corresponding Active Directory object, the categorizer rewrites the email address in the message to match the primary SMTP address. The original email address is saved in the ORCPT= entry in the **RCPT TO:** command in the message envelope.

Sender message restrictions

The size that's used for the sender message size restriction is the value of the **X-MS-Exchange-Organization-OriginalSize:** header field in the message header. Exchange uses this header field to record the original message size of the message when it entered the Exchange organization. Whenever the message is checked against the specified message size limits, the lower value of the current message size or the original message size header is used. The size of the message can change because of content conversion, encoding, and agent processing. If this header field doesn't exist, it's created by using the current message size value. If the message is too large, an NDR is generated and additional message processing is stopped.

The sender recipient limit is only enforced in the Transport service on the first Mailbox server that processes the message. The original, unexpanded message envelope recipient count is compared to the sender recipient limit. The original, unexpanded message envelope recipient count is used to avoid the partial message delivery problems that occured in Microsoft Exchange Server 2003 when nested distribution lists used remote expansion servers.

The message sender and all recipients are marked as resolved by stamping an extended property in the message. This extended property allows the message to bypass top-level resolution if the message must go through recipient resolution again. A message may have to go through recipient resolution again because the Microsoft Exchange Transport service restarted.

Return to top

Expansion

Expansion occurs after top-level resolution. Expansion completely expands nested levels of recipients into individual recipients. Expansion may require multiple trips through the expansion

process to expand all recipients. Not all recipients have to be expanded. However, all recipients must go through the expansion process. The expansion process also enforces recipient message restrictions for all kinds of recipients.

The following list describes the kinds of recipients that require expansion:

- **Distribution groups and dynamic distribution groups** Distribution groups are expanded based on the **memberOf** Active Directory property. Dynamic distribution groups are expanded by using the Active Directory query definition. If the *ExpansionServer* parameter is set on the group, the group isn't expanded by the current server. The distribution group is routed to the specified server for expansion.

Note:

If you select a specific transport server in your organization as the expansion server, the distribution group usage becomes dependent on the availability of the expansion server. If the expansion server is unavailable, any messages that are sent to the distribution group can't be delivered. If you plan to use specific expansion servers for your distribution groups, to reduce the risk of service interruption, you should consider implementing high availability solutions for these servers.

- **Alternative recipients** The *ForwardingAddress* parameter may be set on mailboxes and mail-enabled public folders. The *ForwardingAddress* parameter redirects all messages to the specified alternative recipient. This is known as a *forwarded recipient*. When an alternative delivery address is specified in the *ForwardingAddress* parameter and the *DeliverToMailboxAndForward* parameter is set to `$true`, the message is delivered to the original recipient and the alternative recipient. This is known as *delivered and forwarded recipient*.
- **Contact chains** A *contact chain* is a mail user or mail contact that has the *ExternalEmailAddress* parameter set to the email address of another recipient in the Exchange organization.

Detection of recipient loops

As the distribution groups, alternative recipients, and contacts chains are expanded, the categorizer checks for *recipient loops*. A recipient loop is a recipient configuration problem that causes message delivery to the same recipients in an endless circle. The following list describes the different types of recipient loops:

- **Harmless recipient loop** A harmless recipient loop results in successful message delivery. The following list describes two scenarios when harmless recipient loops occur:
  - When two distribution groups contain one another as members.
  - When mailboxes or mail-enabled public folders are set to deliver and forward to one another. This happens when the *DeliverToMailboxAndForward* parameter of both recipients is set to `$true` and the *ForwardingAddress* parameter is set to one another.

When a harmless recipient loop is detected, the message is delivered to the recipient, but no additional attempts are made to deliver the message to the same recipient.

- **Broken recipient loop** A broken recipient loop can't result in successful message delivery. An example of a broken recipient loop is when mailboxes or mail-enabled public folders have the *ForwardingAddress* parameter set to one another. When the categorizer detects a broken recipient

loop, expansion activity for the current recipient stops, and an NDR is generated for the recipient.

Detection of recipient loops doesn't prevent duplicate message delivery. For example, Distribution Group C will experience duplicate message delivery if the following conditions are true:

- Distribution Group B and Distribution Group C are members of Distribution Group A.
- Distribution Group C is also a member of Distribution Group B.

Delivery report redirection for distribution groups

When a distribution group is expanded, the message type is checked to determine whether it's a delivery report message. If the message is a delivery report, the redirection settings of the distribution group are checked to determine whether redirection of the delivery report is required. You may want to suppress the delivery reports because the delivery reports may disclose unwanted information about the distribution group and its membership.

The following list describes the delivery report redirection settings that are available for distribution groups and dynamic distribution groups:

- **ReportToManagerEnabled** This parameter enables delivery reports to be sent to the distribution group manager. Valid values are `$true` or `$false`. The default value is `$false`. For a distribution group, the manager is controlled by the *ManagedBy* parameter in the **Set-Group** cmdlet. For a dynamic distribution group, the manager is controlled by the *ManagedBy* parameter in the **Set-DynamicDistributionGroup** cmdlet.
- **ReportToOriginatorEnabled** This parameter enables delivery reports to be sent to the sender of email messages that are sent to this distribution group. Valid values are `$true` or `$false`. The default value is `$true`.

Note:

The values of the *ReportToManagerEnabled* parameter and *ReportToOriginatorEnabled* parameter can't both be `$true`. If one parameter is set to `$true`, the other must be set to `$false`. The values of both parameters can be `$false`. This suppresses all redirection of all delivery report messages.

The following list describes the available delivery report messages:

- **Delivery receipt (DR)** This report confirms that a message was delivered to its intended recipient.
- **Delivery status notification (DSN)** This report describes the result of an attempt to deliver a message. For more information about DSN messages, see DSNs and NDRs.
- **Message disposition notification (MDN)** This report describes the status of a message after it has been successfully delivered to a recipient. A read notification (RN) and a non-read notification (NRN) are both examples of an MDN message. MDN messages are defined in RFC 2298 and are controlled by the **Disposition-Notification-To:** header field in the message header. MDN settings that use the `Disposition-Notification-To:` header field are compatible with many different message servers. MDN settings can also be defined by using MAPI properties in Microsoft Outlook and Exchange.
- **Non-delivery report (NDR)** This report indicates to the message sender that the message

couldn't be delivered to the specified recipients.

- **Non-read notification (NRN)** This report indicates that a message was deleted before it was read.
- **Out of office (OOF)** This report indicates that the recipient won't respond to email messages. The acronym OOF dates back to the original Microsoft messaging system where the corresponding notification was named "out of facility."
- **Read notification (RN)** This report indicates that a message was read.
- **Recall Report** This report indicates the status of a recall request for a specific recipient. A recall request is when a sender tries to recall a sent message by using Outlook.

When a delivery report message is sent to a distribution group, the following settings cause the report message to be deleted:

- Report redirection isn't set. Alternatively, report redirection is set to the message sender.
- Report redirection is set to the distribution group manager, and the delivery report message isn't an NDR.

When a delivery report message is sent to a distribution group, the delivery report message may be delivered to the distribution group manager. This happens when report redirection is set to the distribution group manager, and the report message is an NDR.

When a message that isn't a delivery report message is sent to a distribution group, the message is delivered to the distribution group members. The report request settings are summarized in the following list:

- If report redirection is set to the message sender, the report request settings aren't modified.
- If report redirection isn't set, all report request settings are suppressed. The `NOTIFY=NEVER` entry is added to **RCPT TO:** for each recipient in the message envelope.
- If report redirection is set to the distribution group manager, all report request settings are suppressed except NDR messages that are sent to the distribution group manager.

Message restrictions on recipients

The expansion process also enforces any message restrictions that are configured for recipients. These restrictions may be configured individually for each recipient or organizationally for all Hub Transport servers in the Exchange organization. The following table describes the message restrictions that are configured for recipients.

Message restrictions on recipients

Source	Parameter	Description
Set-DistributionGroup **Set-DynamicDistributionGroup** **Set-Mailbox**	*MaxReceiveSize*	The *MaxReceiveSize* parameter specifies the size that's used for message restrictions that are configured for recipients is the value of the **X-MS-Exchange-**

Set-MailContact **Set-MailPublicFolder** **Set-MailUser** **Set-TransportConfig**		**Organization-OriginalSize:** header field in the message header. Exchange uses this header field to record the original message size of the message when it entered the Exchange organization. Whenever the message is checked against the specified message size limits, the lower value of the current message size or the original message size header is used. The size of the message can change because of content conversion, encoding, and agent processing. If this header field doesn't exist, it's created by using the current message size value. If the message is too large, an NDR is generated and additional message processing is stopped.
Set-DistributionGroup **Set-DynamicDistributionGroup** **Set-Mailbox** **Set-MailContact** **Set-MailPublicFolder** **Set-MailUser**	*RequireSenderAuthenticationEnabled*	The *RequireSenderAuthenticationEnabled* parameter requires that all messages that are sent to the recipient come from authenticated senders. When the value of this parameter is set to `$true`, messages from unauthenticated senders are

		rejected. All senders who send messages to the System and System Attendant mailboxes must be authenticated.
Set-DistributionGroup **Set-DynamicDistributionGroup** **Set-Mailbox** **Set-MailContact** **Set-MailPublicFolder** **Set-MailUser**	*AcceptMessagesOnlyFromSendersOrMembers* *RejectMessagesFromSendersOrMembers*	The *AcceptMessagesOnlyFromSendersOrMembers* parameter specifies the senders or distribution groups whose members are allowed to send messages to the recipient. Note that this parameter combines the functionality of the older *AcceptMessagesOnlyFrom* and *AcceptMessagesOnlyFromDLMembers* parameters. The *RejectMessagesFromSendersOrMembers* parameter specifies the senders or distribution groups whose members aren't allowed to send messages to the recipient. Note that this parameter combines the functionality of the older *RejectMessagesFrom* and *RejectMessagesFromDLMembers* parameters. The categorizer checks the recipient permission in two passes. The first pass determines whether the sender

		is present in the *AcceptOnlyMessagesFromSendersOrMembers* or *RejectMessagesFromSendersOrMembers* parameter. If the sender isn't found in either parameter, the distribution groups in those parameters are fully expanded. This complete expansion of distribution groups may take some time. We recommend that you minimize the depth of nested distribution groups in those parameters.

Certain types of messages that are sent by authenticated senders are exempt from restrictions. The following list describes the messages that are exempt from recipient restrictions:

- **All messages that are sent by the Microsoft Exchange recipient** These messages include DSN messages, journal reports, quota messages, and other system-generated messages that are sent to internal message senders. For more information about the Microsoft recipient, see Recipients.
- **All messages that are sent by the external postmaster address** These messages include DSN messages and other system-generated messages that are sent to external message senders. For more information about the external postmaster address, see Configure the external postmaster address.

Certain types of messages are blocked when they are sent from the Exchange organization to external domains. The settings are controlled by the following parameters in the **Set-RemoteDomain** cmdlet:

- *AllowedOOFType*
- *AutoForwardEnabled*
- *AutoReplyEnabled*
- *DeliveryReportEnabled*
- *NDREnabled*

For more information, see Remote domains.

Return to top

Bifurcation and controlling recipient expansion

Because the complete list of message recipients is expanded and resolved by recipient resolution, there are occasions when different copies of the same message must be created. These occasions are described by the following scenarios:

- **When message recipients require different message settings** Message properties such as read receipts may have to be enabled for some recipients and blocked for other recipients. Creating a new version of the message that has slightly different properties than the original message is called *bifurcation*.
- **To limit the number of envelope recipients in a single message** The recipient expansion process can generate thousands of individual recipients when large distribution groups are expanded. In Exchange, instead of creating a single copy of the message that has thousands of envelope recipients, multiple copies of the same message that have a limited number of envelope recipients are created.

Bifurcation

Recipient resolution bifurcates a message if the following conditions are true:

- When the message sender in **MAIL FROM:**, in the message envelope, is updated. An example is when the *ReportToManagerEnabled* parameter on a distribution group has a value of `$true`.
- When auto-response messages, such as DSNs, OOF messages, and recall reports must be suppressed.
- When alternative recipients are expanded.
- When a **Resent-From:** header field must be added to the message header. Resent header fields are informational header fields that can be used to determine whether a message has been forwarded by a user. Resent header fields are used so that the message appears to the recipient as if it was sent directly by the original sender. The recipient can view the message header to discover who forwarded the message. Resent header fields are defined in section 3.6.6 of RFC 2822.
- When the history of the expansion of the distribution group must be transmitted.

Controlling recipient expansion

When the number of expanded recipients is too large, the categorizer splits the message into multiple copies. This is done to reduce system resource use during message expansion. The maximum number of envelope recipients in a message is controlled by the *ExpansionSizeLimit* key in the `%ExchangeInstallPath%Bin\EdgeTransport.exe.config` application configuration file. The default value is 1000.

Caution:
We recommend that you don't modify the value of the *ExpansionSizeLimit* key on an Exchange transport server in a production environment.

Return to top

Recipient resolution diagnostics

Reporting and diagnostic information for recipient resolution is provided by performance counters, message tracking log entries, and recipient resolution logging. These sources can help you identify and diagnose problems with recipient resolution.

Recipient resolution performance counters

The following table describes the performance counters that are available for recipient resolution.

Recipient resolution performance counters

Counter name	Display name	Description
AmbiguousRecipientsTotal	Ambiguous Recipients	This is the total number of ambiguous recipients that were detected during recipient resolution. Ambiguous recipients are different recipients that have matching **legacyExchangeDN** Active Directory attributes or matching **proxyAddresses** Active Directory attributes.
AmbiguousSendersTotal	Ambiguous Senders	This is the number of ambiguous senders that were detected during recipient resolution. Ambiguous senders are different senders that have matching **legacyExchangeDN** Active Directory attributes or matching **proxyAddresses** Active Directory attributes.
FailedRecipientsTotal	Failed Recipients	This is the number of failed recipients that were detected

		during recipient resolution.
LoopRecipientsTotal	Loop Recipients	This is the number of recipients that failed recipient resolution because of recipient loops.
MessagesChippedTotal	Messages Chipped	This is the total number of copies of the same message that were created during recipient resolution to control the number of envelope recipients in a single message. In Exchange, this process is referred to as *chipping*.
MessagesCreatedTotal	Messages Created	This is the number of messages that were created during recipient resolution.
MessagesRetriedTotal	Messages Retried	This is the number of messages that were scheduled for retry during recipient resolution.
UnresolvedOrgRecipientsTotal	Unresolved Org Recipients	This is the number of unresolved recipients from an authoritative domain that were detected during recipient resolution.
UnresolvedOrgSendersTotal	Unresolved Org Senders	This is the number of unresolved senders from an authoritative domain that were detected during recipient resolution.

Recipient resolution events in the message tracking log

The following table describes the recipient resolution events that are written in the message tracking log.

Recipient resolution events in the message tracking log

Message tracking event	Description
EXPAND	This event indicates that a distribution group was expanded.
REDIRECT	This event indicates that a message sent to a mailbox recipient or a mail-enabled public folder recipient was redirected to an alternative recipient as specified by the *ForwardingAddress* parameter.
RESOLVE	This event indicates that a recipient email address was changed to the primary SMTP email address of the corresponding Active Directory recipient object.
TRANSFER	This event indicates that message bifurcation or chipping occurred.

For more information about message tracking, see Message tracking.

Recipient resolution logging

Recipient resolution logging is controlled by the *ResolverLogLevel* key in the `%ExchangeInstallPath%Bin\EdgeTransport.exe.config` application configuration file. The valid values for this key are `Disabled`, `Enabled`, and `FullContent`. The default value is `Disabled`. When the value is set to `Enabled`, only message envelope data is logged. When the value is set to `FullContent`, message envelope data and message header data is logged. The log files are stored at `%ExchangeInstallPath%Logging\Resolver`.

Note:

Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.

Return to top

DNS query failure sensitivity

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-22*

In Microsoft Exchange Server 2013, you can adjust the DNS query sensitivity for slightly faster message delivery when DNS errors are encountered for the destination domain. However, depending on the DNS errors, this adjustment may cause delivery failures in certain circumstances.

DNS queries and remote message delivery

The Exchange server that's responsible for delivering messages to external recipients must be able to find a destination messaging server that accepts mail for the external recipients. Depending on the destination, the messages are put in one or more remote delivery queues as they await delivery to the remote recipients. For more information about delivery queues, see Queues.

The Exchange server queries the configured DNS servers to find the DNS records that are required to deliver the message. The DNS servers are queried in the order in which they're listed. If one of the DNS servers is unavailable, the query goes to the next DNS server on the list. The DNS servers are queried for the following information:

- **Mail exchange (MX) records for the domain part of the external recipient** The MX record contains the fully qualified domain name (FQDN) of the messaging server that's responsible for accepting messages for the domain, and a preference value for that messaging server. A lower preference value indicates a preferred messaging server. The preference value is important if the domain has more than one MX record. To optimize fault tolerance, most organizations use multiple messaging servers and multiple MX records that have different preference values.
- **Address (A) records for the destination messaging servers** Every messaging server that's used in an MX record should have a corresponding A record. The A record is used to find the IP address of the destination messaging server. The subscribed Edge Transport server uses the IP address to open an SMTP connection with the destination messaging server. Although it's technically possible to use the FQDN of a canonical name (CNAME) record in an MX record, this practice violates RFC 974, RFC 1034, RFC 1912, and RFC 2181, and is therefore not supported by most messaging servers.

The required combination of iterative DNS queries and recursive DNS queries that start with a root DNS server is used to resolve the FQDN of the messaging server that's found in the MX record into an IP address.

In Exchange 2013, there's a DNS query limit that's not configurable of 5 seconds for each DNS server, and a one-minute limit for the entire DNS query.

Potential DNS problems

Even when the DNS settings on the Exchange server are configured correctly, problems with the DNS records for a specific domain or problems with any of the DNS servers that are used to find the authoritative DNS server for a specific domain may still occur. Generally, these problems are beyond your control and need to be resolved by the parties that own those DNS servers. These DNS-related errors may be caused by one or more of the following conditions:

- Invalid DNS records for the destination domain
- Problems with DNS server utilization
- Problems with DNS server replication

In Exchange 2013, when a DNS query results in errors, the query continues to the next DNS server only if that DNS server hasn't already returned an error for the current query.

You can control the DNS query failure sensitivity by modifying the `%ExchangeInstallPath%bin\EdgeTransport.exe.config` XML application configuration file. This file is associated with the Microsoft Exchange Transport service. Changes you save to this file are applied after you restart the Microsoft Exchange Transport service. When you restart this service, mail flow on the server is temporarily interrupted. The DNS query failure sensitivity is controlled by the *DnsFaultTolerance* key in the EdgeTransport.exe.config file. This key uses the following values:

- **Lenient** When the DNS query encounters a combination of valid MX records and invalid MX records, the DNS query continues until the DNS query time-out value of one minute is reached. The invalid MX records are discarded, and the valid MX record that has the lowest preference value is used to deliver the message to the destination messaging server. This is the default value.
- **Normal** When the DNS query first encounters an invalid MX record, any resolved MX records that have a preference value that's greater than or equal to the invalid MX records are immediately discarded. The remaining MX record that has the lowest preference value is used to deliver the message to the destination messaging server without waiting for the whole DNS query to time out. Although this behavior may result in faster message delivery, the potential drawback of this behavior is the DNS query may have no valid MX records if the following conditions are true:
  - The invalid MX record is the first MX record for the destination domain.
  - The valid MX records have the same precedence value as the invalid MX records.

In both `Normal` mode and `Lenient` mode, the results of the DNS query for an invalid MX record are never cached. The next time that a DNS query is executed, it will try to resolve the MX records for the destination domain.

Note:

Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.

Use Telnet to test SMTP communication

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-10*

This topic explains how to use Telnet to test Simple Mail Transfer Protocol (SMTP) communication between messaging servers. By default, SMTP listens on port 25. If you use Telnet on port 25, you can enter the SMTP commands that are used to connect to an SMTP server and send a message exactly as if your Telnet session was an SMTP messaging server. You can see the success or failure of each step in the connection and message submission process.

Here are the scenarios where you may want to use Telnet to test SMTP communication to or from the transport servers that exist in your Microsoft Exchange organization:

- Connect to your organization's Internet-facing Exchange server from a host that is located outside your perimeter network and send a test message.
- Connect to a remote messaging server from your organization's Internet-facing Exchange server and send a test message.

The procedure in this topic shows you how to use Telnet Client, which is a component that is included with Microsoft Windows. Third-party Telnet clients may require a syntax that is different from that of the Windows Telnet component.

What do you need to know before you begin?

- Estimated time to complete: 30 minutes
- Exchange permissions don't apply to the procedures in this topic. These procedures are performed in the operating system of the Exchange Server or a client computer.
- The procedures in this topic are best used to connect to and from Internet-facing servers that allow anonymous connections. Message transmission between internal Exchange servers is encrypted and authenticated. To use Telnet to connect to the Hub Transport service on a Mailbox server, you'll need to create a Receive connector that's configured to allow anonymous access or Basic authentication to receive messages. If the connector allows Basic authentication, you need a utility to convert the text strings that are used for the username and password into the Base64 format. Because the user name and password are easily discernible when Basic authentication is used, we don't recommend Basic authentication without encryption.
- If you connect to a remote messaging server, consider performing the procedures in this topic on your Internet-facing Exchange server. This will help to avoid rejection of the test message by remote messaging servers that are configured to validate the source IP address, the

corresponding domain name system (DNS) domain name, and the reverse lookup IP address of any Internet host that tries to send a message to the server.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Install the Telnet Client in Windows

By default, the Telnet Client isn't installed in most client or server versions of the Microsoft Windows operating systems. To install it, see Install Telnet Client.

Step 2: Use Nslookup to find the FQDN or IP address in the MX record of the remote SMTP server

To connect to a destination SMTP server by using Telnet on port 25, you must use the fully qualified domain name (FQDN) or the IP address of the SMTP server. If the FQDN or IP address is unknown, the easiest way to find this information is to use the Nslookup command-line tool to find the MX record for the destination domain.

1. At a command prompt, type **nslookup**, and then press ENTER. This command opens the Nslookup session.
2. Type **set type=mx** and then press ENTER.
3. Type **set timeout=20** and then press ENTER. By default, Windows DNS servers have a 15-second recursive DNS query time-out limit.
4. Type the name of the domain for which you want to find the MX record. For example, to find the MX record for the fabrikam.com domain, type **fabrikam.com.**, and then press ENTER.

Note:

The trailing period (**.**) indicates a FQDN. The use of the trailing period prevents any default DNS suffixes that are configured for your network from being unintentionally added to the domain name.

The output of the command will resemble the following:

```
fabrikam.com mx preference=10, mail exchanger =
mail1.fabrikam.com
fabrikam.com mx preference=20, mail exchanger =
mail2.fabrikam.com
```

```
mail1.fabrikam.com internet address = 192.168.1.10
mail2 fabrikam.com internet address = 192.168.1.20
```

You can use any of the host names or IP addresses that are associated with the MX records as the destination SMTP server. A lower value of preference indicates a preferred SMTP server. You can use multiple MX records and different values of preference for load balancing and fault tolerance.

5. When you're ready to end the Nslookup session, type **exit**, and then press ENTER.

Note:

Firewall or Internet proxy restrictions that are imposed on your organization's internal network may prevent you from using the Nslookup tool to query public DNS servers on the Internet.

Step 3: Use Telnet on Port 25 to test SMTP communication

In this example, the following values are used:

- **Destination SMTP server** mail1.fabrikam.com
- **Source domain** contoso.com
- **Sender's e-mail address** chris@contoso.com
- **Recipient's e-mail address** kate@fabrikam.com
- **Message subject** Test from Contoso
- **Message body** This is a test message

Note:

- The commands in Telnet Client are not case-sensitive. The SMTP command verbs are capitalized for clarity.
- You can't use the backspace key after you have connected to the destination SMTP server within the Telnet session. If you make a mistake as you type an SMTP command, you must press ENTER and then type the command again. Unrecognized SMTP commands or syntax errors result in an error message that resembles the following:

```
500 5.3.3 Unrecognized command
```

1. At a command prompt, type **telnet**, and then press ENTER. This command opens the Telnet session.
2. Type **set localecho** and then press ENTER. This optional command lets you view the characters as you type them. This setting may be required for some SMTP servers.
3. Type **set logfile** <*filename*>. This optional command enables logging of the Telnet session to the specified log file. If you only specify a file name, the location of the log file is the current working directory. If you specify a path and a file name, the path must be local to the computer. Both the path and the file name that you specify must be entered in the Microsoft DOS 8.3 format. The path that you specify must already exist. If you specify a log file that doesn't exist, it will be created for you.
4. Type **open mail1.fabrikam.com 25** and then press ENTER.

5. Type **EHLO contoso.com** and then press ENTER.
6. Type **MAIL FROM:chris@contoso.com** and then press ENTER.
7. Type **RCPT TO:kate@fabrikam.com NOTIFY=success,failure** and then press ENTER. The optional NOTIFY command defines the particular delivery status notification (DSN) messages that the destination SMTP server must provide to the sender. DSN messages are defined in RFC 1891. In this case, you're requesting a DSN message for successful or failed message delivery.
8. Type **DATA** and then press ENTER. You will receive a response that resembles the following:

```
354 Start mail input; end with <CLRF>.<CLRF>
```

9. Type **Subject: Test from Contoso** and then press ENTER.
10. Press ENTER. RFC 2822 requires a blank line between the `Subject:` header field and the message body.
11. Type **This is a test message** and then press ENTER.
12. Press ENTER, type a period (`.`) and then press ENTER. You will receive a response that resembles the following:

```
250 2.6.0 <GUID> Queued mail for delivery
```

13. To disconnect from the destination SMTP server, type **QUIT** and then press ENTER. You will receive a response that resembles the following:

```
221 2.0.0 Service closing transmission channel
```

14. To close the Telnet session, type **quit** and then press ENTER.

Step 4: Evaluate the Results of the Telnet Session

This section provides information about responses that may be provided to the following commands, which were used in the previous example:

- Open mail1.fabrikam.com 25
- EHLO contoso.com
- MAIL FROM:chris@contoso.com
- RCPT TO:kate@fabrikam.com NOTIFY=success,failure

Note:
The 3-digit SMTP response codes that are defined in RFC 2821 are the same for all SMTP messaging servers. The text descriptions may differ slightly for some SMTP messaging servers.

Open mail1.fabrikam.com 25

Successful Response `220 mail1.fabrikam.com Microsoft ESMTP MAIL Service ready at <day-date-time>`

Failure Response `Connecting to mail1.fabrikam.com...Could not open connection to the host, on port 25: Connect failed`

Possible Reasons for Failure

- The destination SMTP service is unavailable.

- There are restrictions on the destination firewall.
- There are restrictions on the source firewall.
- An incorrect FQDN or IP address for the destination SMTP server was specified.
- An incorrect port number was specified.

EHLO contoso.com

Successful Response `250 mail1.fabrikam.com Hello [<sourceIPaddress>]`

Failure Response `501 5.5.4 Invalid domain name`

Possible Reasons for Failure There are invalid characters in the domain name. Alternatively, there are connection restrictions on the destination SMTP server.

Note:

EHLO is the Extended Simple Message Transfer Protocol (ESMTP) verb that is defined in RFC 2821. ESMTP servers can advertise their capabilities during the initial connection. These capabilities include their maximum accepted message size and their supported authentication methods. HELO is the older SMTP verb that is defined in RFC 821. Most SMTP messaging servers support ESMTP and EHLO.

MAIL FROM:chris@contoso.com

Successful Response `250 2.1.0 Sender OK`

Failure Response `550 5.1.7 Invalid address`

Possible Reasons for Failure There is a syntax error in the sender's e-mail address.

Failure Response `530 5.7.1 Client was not authenticated`

Possible Reasons for Failure The destination server does not accept anonymous message submissions. You receive this error if you try to use Telnet to submit a message directly to a Hub Transport server.

RCPT TO:kate@fabrikam.com NOTIFY=success,failure

Successful Response `250 2.1.5 Recipient OK`

Failure Response `550 5.1.1 User unknown`

Possible Reasons for Failure The specified recipient does not exist in the organization.

Configure the external postmaster address

Exchange Server 2013 > Mail flow > Mail routing >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-30*

The external postmaster address is used as the sender for system-generated messages and notifications sent to message senders that exist outside of your Microsoft Exchange Server 2013 organization. An external sender is any sender that has an email address in a domain that isn't configured as an accepted domain in your organization.

By default, the value of the external postmaster address setting is blank. This default value causes the following behavior in your Exchange organization:

- The external postmaster address is postmaster@<*Default accepted domain*> for all Mailbox servers and subscribed Edge Transport servers.
- The external postmaster address is postmaster@<*Edge Transport server FQDN*> for all unsubscribed Edge Transport servers.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport configuration" entry in the Mail flow permissions topic.
- When you configure a custom external postmaster address, that value applies to all Exchange 2013 Mailbox servers and Exchange 2010 Hub Transport servers in your Exchange organization. However, that value isn't replicated to Edge Transport servers. If you specify a custom value for the external postmaster address, you need to manually configure the external postmaster address value on any Edge Transport servers.
- If you have any Exchange 2007 Hub Transport servers or Edge Transport servers in your organization, you need to configure the custom external postmaster address on each one of those servers using the **Set-TransportServer** cmdlet. For more information, see Managing the External Postmaster Address.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to configure the external postmaster address

1. In the EAC, navigate to **Mail flow** > **Receive connectors** > **More options** ··· > **Organization transport settings** > **Delivery** tab.
2. In the **External postmaster address** field, enter the SMTP email address, for example, `postmaster@contoso.com`. If you want to return the external postmaster address to the default value, delete any existing value so the field is blank.
3. When you're finished, click **Save**.

Use the Shell to configure the external postmaster address

To configure the external postmaster address, use the following syntax.

```
Set-TransportConfig -ExternalPostmasterAddress <postmaster address>
```

For example, to set the external postmaster address to the value `postmaster@contoso.com`, run the following command

```
Set-TransportConfig -ExternalPostmasterAddress postmaster@contoso.com
```

To return the external postmaster address to the default value, run the following command:

```
Set-TransportConfig -ExternalPostmasterAddress $null
```

How do you know this worked?

To verify that you have successfully configured the external postmaster address, do the following:

1. Run the following command on a Mailbox server to verify the external postmaster address value:

```
Get-TransportConfig | Format-List ExternalPostmasterAddress
```

2. From an external email account, send a message to your Exchange organization that will generate a delivery status notification (DSN). For example, you can configure a transport rule to send a non-delivery report (NDR) for a message from that sender that contains specific keywords. Verify the sender's email address in the DSN matches the value you specified.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Connectors

Exchange Server 2013 > Mail flow >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-12*

Connectors are used to control inbound and outbound mail flow in Microsoft Exchange Server 2013. With connectors, you can route mail to and receive mail from recipients outside of your organization, a partner through a secure channel, or a message-processing appliance.

The most commonly used connector types are Send connectors, which control outbound messages, and Receive connectors, which control inbound messages.

For more information

Send connectors

Receive connectors

Create a Send connector for email sent to the Internet

Create a secure Receive connector to receive email from a partner

Send connectors

Exchange Server 2013 > Mail flow > Connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-15*

In Microsoft Exchange Server 2013, a Send connector controls the flow of outbound messages to the receiving server. They are configured on Mailbox servers running the Transport service. Most commonly, you configure a Send connector to send outbound email messages to a smart host or directly to their recipient, using DNS.

Exchange 2013 Mailbox servers running the Transport service require Send connectors to deliver messages to the next hop on the way to their destination. Send connectors that are created on Mailbox servers are stored in Active Directory and are available to all Mailbox servers running the Transport service in the organization.

Important:
When you deploy Exchange 2013, outbound mail flow cannot occur until you configure a Send connector to route outbound mail to the Internet. For more information, see Create a Send connector for email sent to the Internet.

Selecting the Type for a Send connector

Typically you create a Send connector in the **Mail flow** section of the Exchange Administration Center (EAC). When you create a new Send connector, you choose an available **Type** appropriate to your connection scenario. The type determines the default permission sets that are assigned on the connector and grants those permissions to trusted security principals. Security principals include users, computers, and security groups.

Procedures that explain specific **Type** selections include Create a Send connector to route

outbound email through a smart host and Create a Send connector to send email to a partner, with Transport Layer Security (TLS) applied.

In you prefer using the Exchange Management Shell to the EAC, you can create a Send connector and specify a type by using the New-SendConnector cmdlet.

New Send connector features in Exchange 2013

With the relationship between the Front End Transport service on Client Access servers and the Transport service on Mailbox servers in Exchange 2013 comes new behavior for Send connectors. Firstly, you can set a Send connector in the Transport service of a Mailbox server to route outbound mail through a Front End transport server in the local Active Directory site, by means of the *FrontEndProxyEnabled* parameter of the Set-SendConnector cmdlet, thus consolidating how email is routed from the Transport service. Mail routing provides more information about transport services, routing behavior, and destinations in Exchange 2013. Mail flow provides an overview of the transport pipeline and descriptions of each transport service.

The *IsCoexistenceConnector* parameter has been deprecated. In most cases, when you want to configure a hybrid environment, where a portion of your mailboxes are hosted on-premises and a portion are hosted in the cloud, we recommend that you use the Hybrid Configuration Wizard.

LinkedReceiveConnector has been deprecated. This parameter was used to create connectors that could route messages to a third-party anti-spam service, for instance. Now, in most cases, you route mail to your anti-spam service by means of your MX record, and the linked-connector behavior is not necessary.

The default maximum message size, specified by the *MaxMessageSize* parameter, has been increased from 10MB to 25MB. Set-SendConnector provides more information on how to set parameters on a Send connector.

TlsCertificateName has been added. It is used to authenticate the local certificate to be used for outbound connections and minimize the risk of fraudulent certificates.

Create a Send connector for email sent to the Internet

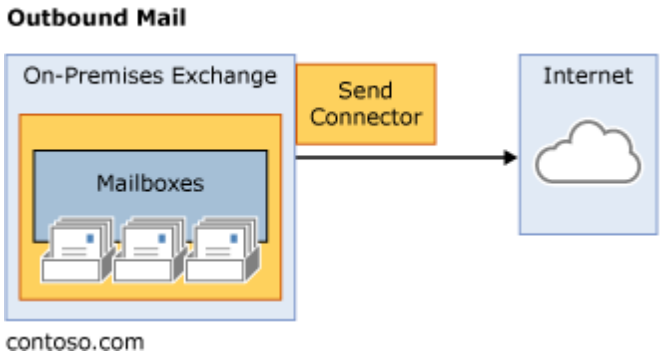
Mail flow > Connectors > Send connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-29*

By default, Microsoft Exchange Server 2013 doesn't allow you to send mail outside of your domain.

To send mail outside your domain, you need to create a Send connector. The following graphic illustrates mail flow when you create a Send connector to send mail to the Internet.



Interested in scenarios where this procedure is used? See the following topics:

- Configure mail flow and client access

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Send connectors" entry in the Mail flow permissions topic.
- See Deploy a new installation of Exchange 2013 if you are beginning your installation. After the installation you can use the steps in this topic to create your outbound connector.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to create a send connector for email sent to the Internet

1. In the EAC, navigate to **Mail flow** > **Send connectors**, and then click **Add +**.
2. In the **New send connector** wizard, specify a name for the send connector and then select **Internet** for the **Type**. Click **Next**.
3. Verify that **MX record associated with recipient domain** is selected, which specifies that the connector uses the domain name system (DNS) to route mail. Click **Next**.
4. Under **Address space**, click **Add +**. In the **Add domain** window, make sure SMTP is listed as the **Type**. For **Fully Qualified Domain Name (FQDN)**, enter *, which indicates that this send connector applies to messages addressed to any domain. Click **Save**.
5. Make sure **Scoped send connector** is not selected and then click **Next**.

6. For **Source server**, click **Add +**. In the **Select a server** window, select a Mailbox server that will be used to send mail to the Internet via the Client Access server and click **Add +**. After you've selected the server, click **Add +**. Click **OK**.
7. Click **Finish**.

Once you have created the Send connector, it appears in the Send connector list.

Use the Shell to route mail through the Client Access server

In Exchange 2013 you can use the *FrontendProxyEnabled* parameter of the **Set-SendConnector** cmdlet to route outbound messages through the Client Access server. This parameter is not set to `$true` by default, but in many cases it can consolidate and simplify mail flow, especially if you are working with an environment with a large number of messaging servers.

This example sets the *FrontendProxyEnabled* parameter to `$true` on a Send connector.

```
Set-SendConnector "Contoso.com Send Connector" -
FrontendProxyEnabled $true
```

How do you know this worked?

To verify that you have successfully created a Send Connector for email sent to the Internet, send mail from one of your users to an outside recipient and verify that the message arrives successfully.

For more information

Send connectors

Create a Send connector to route outbound email through a smart host

Create a Send connector to route outbound email through a smart host

Mail flow > Connectors > Send connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-07*

In some situations you may want to route email through a third-party smart host, such as in an

instance where you have a network appliance that you want to perform policy checks on outbound messages.

Note:
The third-party smart host must use SMTP for transport. If it does not, you should use a Foreign connector or Delivery Agent connector.

Interested in scenarios where this procedure is used? See the following topics:

- Configure mail flow and client access

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Send connectors" entry in the Mail flow permissions topic.
- See Deploy a new installation of Exchange 2013 if you are beginning your installation. After the installation you can use the steps in this topic to create your outbound connector.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to create a Send connector to route outbound email through a smart host

1. In the EAC, navigate to **Mail flow** > **Send connectors**, and then click **Add +**.
2. In the **New send connector** wizard, specify a name for the send connector and then select **Custom** for the **Type**. You typically choose this selection when you want to route messages to computers not running Microsoft Exchange Server 2013. Click **Next**.
3. Choose **Route mail through smart hosts**, and then click **Add +**. In the **Add smart host** window, specify the IP address, such as 192.168.100.1, or the fully qualified domain name (FQDN), such as contoso.com. Click **Save**.

For **Smart host authentication**, choose the type of authentication required by the smart host. If you choose **Basic authentication**, you must provide a user name and password.

Note:
If you choose Basic authentication, we recommend that you use an encrypted connection because the user name and password are sent in clear text.

4. Under **Address space**, click **Add +**. In the **Add domain** window, make sure SMTP is listed as the **Type**. For **Fully Qualified Domain Name (FQDN)**, enter * to specify that this send connector applies to messages sent to any domain. Click **Save**.

5. For **Source server**, click **Add +**. In the **Select a server** window, choose a server and click **Add +**. Click **OK**.
6. Click **Finish**.

Once you have created the send connector, it appears in the Send connector list.

How do you know this worked?

To verify that you have successfully created a Send connector to route outbound email through a smart host, send a message from a user in your organization (you can use the Outlook Web App) to the domain you specified for the **Address space**. If the recipient receives the message, you've successfully configured the send connector.

For more information

Create a Send connector for email sent to the Internet

Send connectors

Create a Send connector to send email to a partner, with Transport Layer Security (TLS) applied

Mail flow > Connectors > Send connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-15*

If you want to ensure secure, encrypted communication with a partner, you can create a Send connector that is configured to enforce Transport Layer Security (TLS) for messages sent to a partner domain. TLS provides secure communication over the Internet.

Interested in scenarios where this procedure is used? See the following topics:

- Configure mail flow and client access

What do you need to know before you begin?

- Estimated time to complete: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Send connectors" entry in the Mail flow permissions

topic.

- See Deploy a new installation of Exchange 2013 if you are beginning your installation. After the installation you can use the steps in this topic to create your outbound connector.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to create a Send connector to send email to a partner, with TLS applied

To create a Send connector for this scenario, log in to the EAC and perform the following steps:

1. In the EAC, navigate to **Mail flow** > **Send connectors**, and then click **Add +**.
2. In the **New send connector** wizard, specify a name for the send connector and then select **Partner** for the **Type**. When you select **Partner**, the connector is configured to allow connections only to servers that authenticate with TLS certificates. Click **Next**.
3. Verify that **MX record associated with recipient domain** is selected, which specifies that the connector uses the domain name system (DNS) to route mail. Click **Next**.
4. Under **Address space**, click **Add +**. In the **Add domain** window, make sure SMTP is listed as the **Type**. For **Fully Qualified Domain Name (FQDN)**, enter the name of your partner domain. Click **Save**.
5. For **Source server**, click **Add +**. In the **Select a server** window, select a Mailbox server that will be used to send mail to the Internet via the Client Access server and click **Add +**. After you've selected the server, click **Add +**. Click **OK**.
6. Click **Finish**.

Once you have created the Send connector, it appears in the Send connector list.

How do you know this worked?

To verify that you have successfully created a Send connector to send email to a partner, with TLS applied, send a message from a user in your organization to a recipient at the partner organization. If the recipient receives the message, the connector was created successfully.

For more information

Create a Send connector for email sent to the Internet

Create a Send connector to route outbound email through a smart host

Configure a cross-forest Send connector

Mail flow > Connectors > Send connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-21*

In Active Directory, the *forest* represents the outer boundary of your directory service. You can create Send connectors to enable communication between forests. In this example, the connectors use Basic authentication.

For additional management tasks related to configuring connectors, see Connectors.

Interested in scenarios where this procedure is used? See the following topics:

- Deploy Exchange 2013 in a cross-forest topology

What do you need to know before you begin?

- Estimated time to complete: 20 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Send connectors" entry and “Receive connectors” entry in the Mail flow permissions topic.
- See Deploy a new installation of Exchange 2013 if you are beginning your installation. After the installation you can use the steps in this topic to create connectors to configure a cross-forest topology.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a user account in each forest

You must create a user account in each forest to use for Basic authentication. Create an account in each forest and add each to the universal security group of the Exchange Server used for communication. This account is used by the Send connector to authenticate to the server receiving mail in the other forest. For example, provide a user account that has the user principal name (UPN)

FourthCoffee@Contoso.com as the credentials that must be used for authentication by the Exchange server in the Fourth Coffee domain when mail is sent to the Exchange server in the Contoso domain.

Use the EAC to create a send connector to route email to another Exchange 2013 forest

Establish cross-forest mail flow using Basic authentication.

1. In the EAC, navigate to **mail flow** > **send connectors**. Click **Add +**.
2. In the **new send connector** wizard, specify a name for the send connector and then select **Internal** for the **Type**. Click **next**.
3. Choose **Route mail through smart hosts**, and then click **Add +**. In the **add smart host** window, specify the IP address of the target server in the second forest, such as 64.4.6.100. Click **save** and then **next**.

For **Smart host authentication**, choose **Basic authentication** and provide a user name and password. Here you can choose **Offer basic authentication only after starting TLS** for secure communication over TLS.

Note:

If you use Basic authentication over TLS, the target server must be configured to use an X.509 certificate.

4. Under **Address space**, click **Add +**. In the **add domain** window, make sure SMTP is listed as the **Type**. For **Fully Qualified Domain Name (FQDN)**, enter the receiving domain, such as fourthcoffee.com. Click **save** and then **next**.
5. For **Source server**, click **Add +**. In the **Select a server** window, choose the server to use and click **add +**. Click **ok**.
6. Click **finish**. The connector appears in the list of Send connectors.

After you create your Send connector, create a Send connector in the second forest that sends mail to the original forest. In this case, the Fully Qualified Domain Name (FQDN) you specify will be the domain name of the first forest. For example, contoso.com.

Use the Shell to set permissions on the Send connector

This example uses the Enable-CrossForestConnector.ps1 script in the Shell to set permissions on the Send connector for use in a cross-forest topology.

```
.\Enable-CrossForestConnector.ps1 -Connector "Cross-Forest"
-user "ANONYMOUS LOGON"
```

How do you know this worked?

To verify that you have successfully created Send connectors to route email to a second forest, send a message from a user in your organization (you can use the Outlook Web App) to the domain you specified for the **Address space**. If the recipient receives the message, you've successfully configured the send connector.

For more information

Create a Send connector for email sent to the Internet

Send connectors

Receive connectors

Exchange Server 2013 > Mail flow > Connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-19*

Receive connectors control the flow of inbound messages to your Exchange organization. They are configured on computers running Microsoft Exchange Server 2013 with the Transport service, or in the Front End service on a Client Access server. They can be created in the Exchange Administration Center (EAC), or in the Exchange Management Shell.

By default, the Receive connectors that are required for internal mail flow are automatically created when a Client Access server or Mailbox server is installed.

Exchange 2013 servers running the Transport service require Receive connectors to receive messages from the Internet, from email clients, and from other email servers. A Receive connector controls inbound connections to the Exchange organization.

Each Receive connector listens for inbound connections that match the settings of the Receive connector. A Receive connector listens for connections that are received through a particular local IP address and port, and from a specified IP address range. You can create a Receive connector when you want to control which servers receive messages from a particular IP address or IP address range, and when you want to configure special connector properties for messages that are received from a particular IP address, such as allowing larger messages or more recipients per message. You can also scope the Receive connector using the *TlsCertificateName* parameter of the **Set-ReceiveConnector** cmdlet, which allows you to specify the certificate to use for the connector.

Receive connectors are scoped to a single server and determine how that specific server listens for connections. When you create a Receive connector on a Mailbox server running the Transport service, the Receive connector is stored in Active Directory as a child object of the server on which it's created.

If you need additional Receive connectors for specific scenarios, you can create them by using the Exchange Administration Center (EAC) or the Exchange Management Shell. Each Receive connector must use a unique combination of IP address bindings, port number assignments, and remote IP address ranges from which mail is accepted.

For more information about how to create a Receive Connector, see Receive connector procedures.

Contents

Default Receive connectors created during setup

Certain Receive connectors are created by default when you install the Mailbox server role.

Default Receive connectors created on a Mailbox server running the Transport service

When you install a Mailbox server running the Transport service, two Receive connectors are created. No additional Receive connectors are needed for typical operation, and in most cases the default Receive connectors don't require a configuration change. These connectors are the following:

- **Default <server name>** Accepts connections from Mailbox servers running the Transport service and from Edge servers.
- **Client Proxy <server name>** Accepts connections from front-end servers. Typically, messages are sent to a front-end server over SMTP.

Each connector is assigned a *TransportRole* value. You can use it to determine the role the connector is running in. This can be helpful in cases where you are running multiple roles on a single server. In the case of each Receive connector previously mentioned, their *TransportRole* value is **HubTransport**.

To view the default Receive connectors and their parameter values, you can use the Get-

ReceiveConnector cmdlet.

Default Receive connectors created on a Front End Transport server

During installation, three Receive connectors are created on the Front End transport, or Client Access server. The default Front End Receive connector is configured to accept SMTP communications from all IP address ranges. Additionally, there is a Receive connector that can act as an outbound proxy for messages sent to the front-end server from Mailbox servers. Finally, there is a secure Receive connector configured to accept messages encrypted with Transport Layer Security (TLS). These connectors are the following:

- **Default FrontEnd <server name>** Accepts connections from SMTP senders over port 25. This is the common messaging entry point into your organization.
- **Outbound Proxy Frontend <server name>** Accepts messages from a Send Connector on a back-end server, with front-end proxy enabled.
- **Client Frontend <server name>** Accepts secure connections, with Transport Layer Security (TLS) applied.

In a typical installation, no additional Receive connectors are required.

Receive connector types

The type determines the default security settings for each Receive connector.

The security settings for a Receive connector specify the permissions that are granted to sessions that connect to the Receive connector and the supported authentication mechanisms.

When you use the EAC to configure a Receive connector, the new receive connector page prompts you to select the type for the connector. Based on your selection, parameters are set to conform to the configuration you have chosen. Specific procedures contain more information about Receive connector type settings. Examples of these procedures are Create a Receive connector to receive email from the Internet and Create a secure Receive connector to receive email from a partner.

Receive connector permission groups

A permission group is a predefined set of permissions that's granted to well-known security principals and assigned to a Receive connector. Security principals include users, computers, and security groups. The use of permission groups simplifies the configuration of permissions on Receive connectors. The **PermissionGroups** property defines the groups or roles that can submit messages to the Receive connector and the permissions that are assigned to those groups.

Permission groups include *Anonymous*, *ExchangeUsers*, *ExchangeServers*, *ExchangeLegacyServers*, and *Partner*.

Receive connector type specifications

The type determines the default permission groups that are assigned to the Receive connector and the default authentication mechanisms that are available for session authentication. The following list describes the available types:

1. **Client** Typically used to connect to clients not using Microsoft Office Outlook. It can use TLS authentication.
2. **Custom** Typically used in a cross-forest scenario, or in a scenario where your organization receives messages from an SMTP message transfer agent.
3. **Internal** Used for communication between servers running the Transport service, or between Mailbox servers running the Transport service and third-party transfer agents.
4. **Internet** Used to receive SMTP mail from the Internet.
5. **Partner** Use this type when you want to configure secure communication with a partner.

Each type is appropriate for a specific connection scenario. Select the type that has the default settings most applicable to the configuration that you want. You can modify permissions by using the **Add-ADPermission** and **Remove-ADPermission** cmdlets. For more information, see the following topics:

- Add-ADPermission
- Remove-ADPermission

Receive connector permissions

Receive connector permissions are assigned to security principals when you specify the permission groups for the connector. When a security principal establishes a session with a Receive connector, the Receive connector permissions determine whether the session is accepted and how the received messages are processed. You can set Receive connector permissions by using the EAC or by using the *PermissionGroups* parameter with the **Set-ReceiveConnector** cmdlet in the Shell. To modify the default permissions for a Receive connector, you can also use the **Add-ADPermission** cmdlet.

Receive connector permissions contains a table that lists security principals and permissions types in detail.

Receive connector authentication settings

In the EAC, you use the authentication settings for a Receive connector to specify the authentication mechanisms that are supported by the Exchange 2010 transport server. In the Shell, you use the *AuthMechanisms* parameter to specify the supported authentication mechanisms. You can configure more than one authentication mechanism for a Receive connector. The following table lists the available authentication mechanisms for a Receive connector.

New Receive connector features in Exchange 2013

The following features were added in Exchange 2013:

- With the *TlsCertificateName* parameter you can specify the local Certificate Authority (CA) issued certificate to use for secure mail. It helps minimize the risk of fraudulent certificates.
- The *TransportRole* parameter designates the server role associated with this connector. It is typically used to specify the server role when you host multiple server roles on a single computer.

See New-ReceiveConnector for more information about these parameters and other parameters for Receive connectors.

Receive connector permissions

Mail flow > Connectors > Receive connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-11*

The following table lists permission types and a description for each.

Receive connector permission	Description
ms-Exch-SMTP-Submit	The session must be granted this permission or it will be unable to submit messages to this Receive connector. If a session doesn't have this permission, the MAIL FROM and AUTH commands will fail.
ms-Exch-SMTP-Accept-Any-Recipient	This permission allows the session to relay messages through this connector. If this permission isn't granted, only messages that are addressed to recipients in accepted domains are accepted by this connector.
ms-Exch-SMTP-Accept-Any-Sender	This permission allows the session to bypass the sender address spoofing check.
ms-Exch-SMTP-Accept-Authoritative-Domain-Sender	This permission allows senders that have e-mail addresses in authoritative domains to

	establish a session to this Receive connector.
ms-Exch-SMTP-Accept-Authentication-Flag	This permission allows Exchange 2003 servers to submit messages from internal senders. Exchange 2010 will recognize the messages as being internal. The sender can declare the message as trusted. Messages that enter your Exchange system through anonymous submissions will be relayed through your Exchange organization with this flag in an untrusted state.
ms-Exch-Accept-Headers-Routing	This permission allows the session to submit a message that has all received headers intact. If this permission isn't granted, the server will strip all received headers.
ms-Exch-Accept-Headers-Organization	This permission allows the session to submit a message that has all organization headers intact. Organization headers all start with **X-MS-Exchange-Organization-**. If this permission isn't granted, the receiving server will strip all organization headers.
ms-Exch-Accept-Headers-Forest	This permission allows the session to submit a message that has all forest headers intact. Forest headers all start with **X-MS-Exchange-Forest-**. If this permission isn't granted, the receiving server will strip all forest headers.
ms-Exch-Accept-Exch50	This permission allows the session to submit a message that contains the XEXCH50 command. This command is needed for interoperability with Exchange 2003. The XEXCH50 command provides data such as the spam confidence

	level (SCL) for the message.
ms-Exch-Bypass-Message-Size-Limit	This permission allows the session to submit a message that exceeds the message size restriction configured for the connector.
Ms-Exch-Bypass-Anti-Spam	This permission allows the session to bypass anti-spam filtering.

Receive connector authentication mechanisms

Mail flow > Connectors > Receive connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-19*

The Receive connector authentication mechanisms are the following:

Authentication mechanism	Description
None	No authentication.
TLS	Advertise STARTTLS. Requires availability of a server certificate to offer TLS.
Integrated	NTLM and Kerberos (Integrated Windows authentication).
BasicAuth	Basic authentication. Requires an authenticated logon.
BasicAuthRequireTLS	Basic authentication over TLS. Requires a server certificate.
ExchangeServer	Exchange Server authentication (Generic Security Services application programming interface (GSSAPI) and Mutual GSSAPI).

ExternalAuthoritative	The connection is considered externally secured by using a security mechanism that's external to Exchange. The connection may be an Internet Protocol security (IPsec) association or a virtual private network (VPN). Alternatively, the servers may reside in a trusted physically controlled network. The `ExternalAuthoritative` authentication method requires the `ExchangeServers` permission group. This combination of authentication method and security group permits the resolution of anonymous sender email addresses for messages that are received through this connector.

For more information about Receive Connector authentication mechanisms, see New-ReceiveConnector.

Receive connector procedures

Mail flow > Connectors > Receive connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-23*

Create a Receive connector to receive email from the Internet

Create a secure Receive connector to receive email from a partner

Create a Receive connector to receive email from a system not running Exchange

Create a Receive connector to receive messages from an internal Exchange server

Create a Receive connector to receive email from the Internet

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-15*

This procedure shows you how to configure a Receive connector to receive email from the Internet.

Note:

In most cases, you won't need to explicitly set up a Receive connector to receive mail from the Internet, because a Receive connector to accept mail from the Internet is implicitly created upon installation of Exchange. See Receive connectors for more information.

Interested in scenarios where this procedure is used? See the following topics:

- Configure mail flow and client access

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Receive connectors" entry in the Mail flow permissions topic.
- See Deploy a new installation of Exchange 2013 if you are beginning your installation. After the installation you can use the steps in this topic to create your receive connector.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to Create a Receive Connector to Receive Messages from the Internet

1. In the EAC, navigate to **Mail flow** > **Receive connectors**. Click **Add +** to create a Receive connector.
2. On the **New receive connector** page, specify a name for the Receive connector and then select **Frontend transport** for the **Role**. Since you are receiving mail from the Internet in this case, we recommend that you initially route mail to your Front End server or servers, to simplify and consolidate your mail flow.
3. Choose **Internet** for the type. The Receive connector will receive mail from Internet senders.
4. For the **Network adapter bindings**, observe that **All available IPV4** is listed in the **IP addresses** list and the **Port** is 25. (Simple Mail Transer Protocol (SMTP) uses port 25.) This indicates that the connector listens for connections on all IP addresses assigned to network adapters on the local server.

Note:

If you have multiple network adapters, on this page you can add an IP address that is assigned to a specific network adapter on the local server, but this isn't required.

5. Click the **Finish** button to create your connector.

Once you have created the Receive connector, it appears in the Receive connector list. If you would like to see an example of how to create a Receive connector with a cmdlet, see New-ReceiveConnector.

How do you know this worked?

To verify that you have successfully created a Receive connector to receive messages from the Internet, test that you can send mail from an outside source and one of your users can receive it. If you can receive mail, you know that the configuration worked successfully.

For more information

Create a secure Receive connector to receive email from a partner

Create a Receive connector to receive email from a system not running Exchange

Create a secure Receive connector to receive email from a partner

Connectors > Receive connectors > Receive connector procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

This procedure shows you how to configure a Receive connector to receive secure email from a partner. Use this procedure when you are required to encrypt communication between you and a trusted partner. The connector is configured to accept connections only from servers that authenticate with Transport Layer Security (TLS).

Interested in scenarios where this procedure is used? See the following topics:

- Configure mail flow and client access

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "Receive connectors" entry in the Mail flow permissions topic.

- See Deploy a new installation of Exchange 2013 if you are beginning your installation. After the installation you can use the steps in this topic to create your receive connector.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to Create a Receive Connector to Receive Secure Messages from a Partner

1. In the EAC, navigate to **Mail flow** > **Receive connectors**. Click **Add +** to create a new Receive connector.
2. On the **New receive connector** page, specify a name for the Receive connector and then select **Frontend Transport** for the **Role**. Since you are receiving mail from a partner in this case, we recommend that you initially route mail to your front end server to simplify and consolidate your mail flow.
3. Choose **Partner** for the type. The Receive connector will receive mail from a trusted third party.
4. For the **Network adapter bindings**, observe that **All available IPV4** is listed in the **IP addresses** list and the **Port** is 25. (Simple Mail Transfer Protocol uses port 25.) This indicates that the connector listens for connections on all IP addresses assigned to network adapters on the local server. Click **Next**.
5. If the Remote network settings page lists 0.0.0.0-255.255.255.255, which means that the Receive connector receives connections from all IP addresses, click **Remove —** to remove it. Click **Add +**, add the IP address for your partner's server, and click **Save**.

Note:

You can also specify an IP address range with CIDR notation, such as 64.4.6.100/24.

6. Click **Finish** to create the connector.

Once you have created the Receive connector, it appears in the Receive connector list. If you would like to see an example of how to create a Receive connector with a cmdlet, see New-ReceiveConnector.

How do you know this worked?

To verify that you have successfully created a Receive connector to receive messages from a partner, test that the partner can send mail to one of your users and that the user successfully receives it. If you can receive encrypted mail (you can verify that TLS is used by checking the message header), you know that the configuration worked successfully.

For more information

Receive connectors

Create a Receive connector to receive email from a system not running Exchange

Connectors > Receive connectors > Receive connector procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

You may have a situation where you want to receive messages from a system not running Exchange. For instance, if you have a network appliance that performs policy checks and then routes messages to your Exchange server. We assume in this case that the appliance uses SMTP. If not, you should use a Foreign connector or a Delivery Agent connector.

Interested in scenarios where this procedure is used? See the following topics:

- Configure mail flow and client access

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Receive connectors" entry in the Mail flow permissions topic.
- See Deploy a new installation of Exchange 2013 if you are beginning your installation. After the installation you can use the steps in this topic to create your receive connector.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to Create a Receive Connector to Receive

Messages from a Messaging Appliance

1. In the EAC, navigate to **Mail flow** > **Receive connectors**. Click **Add +** to create a Receive connector.
2. On the **New receive connector** page, specify a name for the Receive connector and then select **Hub Transport** for the **Role**. In this case, you want your Mailbox server running the Transport service to receive messages from the appliance.
3. Choose **Custom** for the type, since the Receive connector will receive mail from an appliance not running Microsoft Exchange Server 2013.
4. For the **Network adapter bindings**, observe that **All available IPV4** is listed in the **IP addresses** list. Click **Next**.
5. For **Remote network settings**, click **Remove —** to remove **0.0.0.0-255.255.255.255** from the **IP addresses** list, since you want to specify that the connector accepts mail from a specific appliance. Click **Add +** to add a new IP address, and in the **Add IP address** window, add the IP address of your appliance. Click **Save**.
6. Click the **Finish** button to create your connector.

Once you have created the Receive connector, it appears in the Receive connector list. If you would like to see an example of how to create a Receive connector with a cmdlet, see New-ReceiveConnector.

How do you know this worked?

To verify that you have successfully created a Receive connector to receive messages from a messaging appliance, test that you can receive mail from the appliance. If you can receive mail, you know that the configuration worked successfully.

For more information

Create a Receive connector to receive email from the Internet

Create a Receive connector to receive messages from an internal Exchange server

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

You create a Receive connector of the **Internal** type when you want to receive mail from an Exchange server. Use this type of connector to control mail routing within your organization: for example, when you want to route mail from the Transport service on a Mailbox server to a specific Edge Transport server, or from one Mailbox server to another.

Interested in scenarios where this procedure is used? See the following topics:

- Configure mail flow and client access

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Receive connectors" entry in the Mail flow permissions topic.
- See Deploy a new installation of Exchange 2013 if you are beginning your installation. After the installation you can use the steps in this topic to create your receive connector.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Create a Receive Connector to Receive Messages from an Internal Exchange Server

1. In the EAC, navigate to **Mail flow** > **Receive connectors**. Click **Add +** to create a new Receive connector.
2. On the **New receive connector** page, specify a name for the Receive connector and then select **Hub transport** for the **Role**. In this case we assume you want to route mail within your network, not into and out of the organization.
3. Choose **Internal** for the type. The connector is configured with Exchange server authentication.
4. If the Remote network settings page lists 0.0.0.0-255.255.255.255, which means that the Receive connector receives connections from all IP addresses, click **Remove −** to remove it. Click **Add +**, add the IP address for the server you want to receive mail from, such as 192.168.1.1, and click **Save**.
5. Click **Finish** to create the connector.

Once you have created the Receive connector, it appears in the Receive connector list. If you would like to see an example of how to create a Receive connector with a cmdlet, see New-

ReceiveConnector.

How do you know this worked?

To verify that you have successfully created a Receive connector to receive messages from an internal server, test that messages from the sending server travel successfully to the recipient server. One way to do this is to use the Exchange Management Shell to set the *ProtocolLoggingLevel* for the Receive connector you created to `verbose`, using the Set-ReceiveConnector cmdlet, and check the logs to ensure message delivery.

For more information

Create a Receive connector to receive email from the Internet

Create a secure Receive connector to receive email from a partner

Modify the SMTP banner on a Receive connector

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-04*

The *SMTP banner* is the SMTP connection response that a remote SMTP messaging server receives after it connects to a Receive connector that's configured on a computer running Microsoft Exchange Server 2013.

This is the default response received by a remote SMTP messaging server after it connects to the Receive connector:

```
220 <Servername> Microsoft ESMTP MAIL service ready at
<RegionalDay-Date-24HourTimeFormat>
<RegionalTimeZoneOffset>
```

When you specify a custom value for SMTP banner on a Receive connector, a remote SMTP messaging server that connects to that SMTP Receive connector receives the following response.

```
220 <Banner Text>
```

You may want to modify the SMTP banner for Internet-facing SMTP Receive connectors so the

server name and messaging server software aren't disclosed by the SMTP banner.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Receive connectors" entry in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.
- The replacement SMTP banner text string must always start with `220`. As defined in RFC 2821, the default service ready SMTP response code is 220.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to modify the SMTP banner on a Receive connector

Run the following command:

```
Set-ReceiveConnector <ConnectorIdentity> -Banner "220
<Banner Text>"
```

This example modifies the SMTP banner on the existing Receive connector named From the Internet so the SMTP banner displays `220 Contoso Corporation`.

```
Set-ReceiveConnector "From the Internet" -Banner "220
Contoso Corporation"
```

This example removes the custom SMTP banner on the Receive connector named From the Internet, which returns the SMTP banner to the default value.

```
Set-ReceiveConnector "From the Internet" -Banner $null
```

How do you know this worked?

To verify that you have successfully modified the SMTP banner on a Receive connector, do the following:

1. Open a telnet client on a computer that can access the Receive connector, and run the following

command:

```
open <Connector FQDN or IP address> <Port>
```

2. Verify the response from the Receive connector contains the SMTP banner you configured.

Note that this procedure only works on Receive connectors that allow anonymous or Basic authentication. For more information, see Use Telnet to test SMTP communication.

Foreign connectors

Exchange Server 2013 > Mail flow > Connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-25*

A Foreign connector delivers messages to a server or foreign system that does not use SMTP as its primary transport mechanism. An example of this can be a fax-gateway server. A Foreign connector uses a local or shared Drop directory to send outbound messages, by means of a file transfer, to the foreign system. Foreign connectors are created in the Transport service of a Microsoft Exchange Server 2013 Mailbox server.

Foreign gateway servers can send messages into the Exchange 2013 organization by using the Pickup or Replay directories that exist in the Transport service of a Mailbox server. Correctly formatted email message files that you copy to each directory are submitted for delivery to an Exchange mailbox.

Tip:

In most cases where you must deliver outbound messages to a non-SMTP system, we recommend Delivery Agent connectors, because they allow for queue management of messages, messages do not have to be written to the file system, and other benefits. The Delivery agents and Delivery Agent connectors topic provides more details.

For more information

Create a Foreign connector to deliver messages to a non-SMTP fax gateway

Delivery agents and Delivery Agent connectors

New-ForeignConnector

Set-ForeignConnector

Create a Foreign connector to deliver messages to a non-SMTP fax gateway

Mail flow > Connectors > Foreign connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

You may have a scenario where you want to send messages to and receive messages from a fax-gateway server that doesn't use SMTP as its primary transport mechanism. Follow the steps outlined in this procedure to create a Foreign connector that delivers messages to and receives messages from the foreign system.

Tip:

In most cases where you must deliver outbound messages to a non-SMTP system, we recommend Delivery Agent connectors, because they allow for queue management of messages, messages do not have to be written to the file system, and other benefits. The Delivery agents and Delivery Agent connectors topic provides more details.

Interested in scenarios where this procedure is used? See the following topics:

- Planning and deployment

What do you need to know before you begin?

- Estimated time to complete this task: 30 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Foreign connectors" entry in the Mail flow permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Use the Shell to create a Foreign connector that sends messages to a non-SMTP gateway server

1. Run the following command to create the Foreign connector:

```
New-ForeignConnector -Name "Contoso Foreign Connector" -
AddressSpaces "X400:c=US;a=Fabrikam;P=Contoso;5" -
SourceTransportServers Hub01,Hub02
```

In this example, Hub01 and Hub02 are source servers in your organization that you designate to deliver messages to the foreign system. Using more than one source server provides fault tolerance.

Once you have created the Foreign Connector, you can configure the Drop Pickup, and Replay directories, depending on the requirements for your organization.

How do you know this step worked?

To verify that the Foreign connector was created successfully, run the following command:

```
Get-ForeignConnector | Format-List Name
```

Verify that the name for the Foreign connector you created appears.

Step 2: Use the Shell to configure the Drop directory for a Mailbox server running the Transport service

The Drop directory for a Mailbox server running the Transport service is used to deliver outbound messages from your Foreign connector.

You create a directory to use as the Drop directory on your local file system. You can also use a directory on a network file share.

1. Run the following script to specify the Drop directory for your Foreign connector (change the value for the *DropDirectory* parameter to a path appropriate for your environment):

```
Set-ForeignConnector "Contoso Foreign Connector" -
DropDirectory "C:\Drop Directory"
```

How do you know this step worked?

To verify that you have set the Drop Directory correctly, you can run the following cmdlet script and verify the value for the *DropDirectory* parameter:

```
Get-ForeignConnector "Contoso Foreign Connector" | Format-
List
```

Once you have created your Foreign connector and specified your Drop directory, you can send a message using the Mailbox server where you created your Foreign connector and verify that a file is delivered to the Drop directory.

Step 3: Use the Shell to configure the Pickup directory for the Transport service on a Mailbox server

The Pickup directory for the Transport service on a Mailbox server is used to collect messages generated by non-SMTP systems. Use this procedure in cases where you want to gather new messages generated by a non-SMTP system, such as a fax gateway server, by means of file transfer.

For detailed instructions for configuring your Pickup directory, see Configure the Pickup directory and the Replay directory.

How do you know this step worked?

To verify that you have set the Pickup directory correctly, you can run the following command and verify the value for the *PickupDirectoryPath* parameter:

```
Get-TransportService | Format-List PickupDirectoryPath
```

Step 4: Use the Shell to configure the Replay directory for the Transport service on a Mailbox server

The Replay directory for the Transport service on a Mailbox server is used to collect messages generated by non-SMTP systems. Use this procedure to configure the Replay directory in cases where you want to resubmit email messages, typically from a non-SMTP foreign gateway server, that were generated in your Exchange environment and exported from Exchange transport.

For detailed instructions for configuring your Pickup directory, see Configure the Pickup directory and the Replay directory.

How do you know this step worked?

To verify that you have set the Replay directory correctly, you can run the following command and verify the value for the *ReplayDirectoryPath* parameter:

```
Get-TransportService | Format-List ReplayDirectoryPath
```

For more information

Foreign connectors

Delivery agents and Delivery Agent connectors

Delivery agents and Delivery Agent connectors

Exchange Server 2013 > Mail flow > Connectors >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-09*

A delivery agent can deliver messages from your SMTP Exchange Server environment to a system that doesn't use the SMTP protocol. Each delivery agent is associated with a Delivery Agent connector, which queues messages routed to the delivery agent for processing and delivery to the non-SMTP device or system.

Tip:

While the Foreign connector architecture remains in Microsoft Exchange 2013, we recommend using delivery agents for routing messages to non-SMTP systems whenever possible. The primary reasons for this are that you can use queue management for messages, there is no need to manage file transfer to a Drop directory, and you can verify message delivery.

Contents

Function and benefits of Delivery Agents

A delivery agent is a component installed in the Transport service of a Mailbox server that can perform the following tasks:

- Establish a connection to the foreign system for message delivery.
- Retrieve messages from the delivery queues on Mailbox servers.
- Deliver messages to the foreign system.
- Provide acknowledgement for each successful message delivery.

While the Foreign connector architecture remains in Microsoft Exchange Server 2013, we recommend using delivery agents for routing messages to non-SMTP systems whenever possible. Delivery agents provide the following benefits:

- They allow queue management of messages routed to foreign systems.
- Because the messages no longer need to be written to and read from the file system, message delivery performance is improved.

- They provide access to message properties with rich events for agent developers.
- Development time for a delivery agent is faster than implementing a Foreign connector because the delivery agent can use the message representation and management features of Exchange.
- You can verify that the messages are delivered to the foreign system, rather than simply written to the Drop directory.
- The use of Delivery Agent connectors allows service level agreement (SLA) analysis because it's possible to track the latency of message delivery to the foreign system.

Adding Delivery Agents to your organization

To use a delivery agent in your organization, you have to complete the following:

1. Acquire the delivery agent. Typically, delivery agents are written by third parties. Exchange 2013 comes with only one Delivery Agent connector by default: the Text Messaging Delivery Agent connector.
2. Install the delivery agent in the Transport service of your Mailbox servers that will act as source servers for the Delivery Agent connectors.
3. Create a Delivery Agent connector for the specific protocol.

When all of these steps are completed, messages to the foreign systems will be routed through the Delivery Agent connectors and processed by the delivery agent.

Microsoft.Exchange.Data.Transport.Delivery Namespace provides more information about developing a delivery agent.

Delivery Agent connectors

A Delivery Agent connector in Exchange 2013 is similar to the Delivery Agent connector introduced in Exchange 2010. They route messages addressed to foreign systems that do not use the SMTP protocol. When a message is routed to a Delivery Agent connector, the associated delivery agent performs the content conversion and message delivery. Typically, delivery agents are created by a third-party and configured to work with a Delivery Agent connector in your organization.

A Delivery Agent connector cannot be created in the Exchange Administration Center. Rather, you create a Delivery Agent connector in the Exchange Management Shell with the New-DeliveryAgentConnector cmdlet and edit the Delivery Agent connector's properties with Set-DeliveryAgentConnector. You can specify one or more host Mailbox servers for the connector, by using the optional *SourceTransportServers* parameter.

Default text messaging Delivery Agent connector

You can use the Text Messaging Delivery Agent connector to route messages to mobile devices. On your Exchange server, run `Get-DeliveryAgentConnector | fl` to view the connector and all of its parameters. Note that the *DeliveryProtocol* is set to `MOBILE`.

Header firewall

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-31*

In Microsoft Exchange Server 2013, *header firewall* is a mechanism that removes specific header fields from inbound and outbound messages. There are two different types of header fields that are affected by header firewall:

- **X-headers** An *X-header* is a user-defined, unofficial header field. X-headers aren't specifically mentioned in RFC 2822, but the use of an undefined header field starting with **X-** has become an accepted way to add unofficial header fields to a message. Messaging applications, such as anti-spam, antivirus, and messaging servers may add their own X-headers to a message. In Exchange, the X-header fields contain details about the actions that are performed on the message by the Transport service, such as the spam confidence level (SCL), content filtering results, and rules processing status. Revealing this information to unauthorized sources could pose a potential security risk.
- **Routing headers** Routing headers are standard SMTP header fields that are defined in RFC 2821 and RFC 2822. Routing headers stamp a message by using information about the different messaging servers that were used to deliver the message to the recipient. Routing headers that are inserted into messages by malicious users can misrepresent the routing path that a message traveled to reach a recipient.

Header firewall prevents the spoofing of these Exchange-related X-headers by removing them from inbound messages that enter the Exchange organization from untrusted sources. Header firewall prevents the disclosure of these Exchange-related X-headers by removing them from outbound messages sent to untrusted destinations outside the Exchange organization. Header firewall also prevents the spoofing of standard routing headers that are used to track the routing history of a message.

Contents

Message header fields affected by header firewall in Exchange

The following types of X-headers and routing headers are affected by header firewall:

- **Organization X-headers** These X-header fields start with **X-MS-Exchange-Organization-**.
- **Forest X-headers** These X-header fields start with **X-MS-Exchange-Forest-**.

For examples of organization X-headers and forest X-headers, see the Organization X-headers and forest X-headers in Exchange section at the end of this topic.

- **Received: routing headers** A different instance of this header field is added to the message header by every messaging server that accepted and forwarded the message to the recipient. The **Received:** header typically includes the name of the messaging server and a date-timestamp.
- **Resent-*: routing headers** Resent header fields are informational header fields that can be used to determine whether a message has been forwarded by a user. The following resent header fields are available: **Resent-Date:**, **Resent-From:**, **Resent-Sender:**, **Resent-To:**, **Resent-Cc:**, **Resent-Bcc:**, and **Resent-Message-ID:**. The **Resent-** fields are used so that the message appears to the recipient as if it was sent directly by the original sender. The recipient can view the message header to discover who forwarded the message.

Exchange uses two different ways to apply header firewall to organization X-headers, forest X-headers, and routing headers that exist in messages:

- Permissions are assigned to Send connectors or Receive connectors that can be used to preserve or remove specific types of headers in messages when the message travels through the connector.
- Header firewall is automatically implemented for specific types of headers in messages during other types of message submission.

Return to top

How header firewall is applied to Receive connectors and Send connectors

Header firewall is applied to messages that travel through Send connectors and Receive connectors based on specific permissions that are assigned to the connectors.

If the permission is assigned to the Receive connector or Send connector, header firewall isn't applied to the messages that travel through the connector. The affected header fields are preserved in the messages.

If the permission isn't assigned to the Receive connector or Send connector, header firewall is applied to the messages that travel through the connector. The affected header fields are removed from the messages.

The following table describes the permissions on Send connectors and Receive connectors that are used to apply header firewall, and the affected header fields.

Connector type	Permission	Description
Receive connector	**Ms-Exch-Accept-Headers-Organization**	This permission affects organization X-header fields that start with **X-MS-Exchange-Organization-** in inbound messages.
Receive connector	**Ms-Exch-Accept-Headers-Forest**	This permission affects forest X-header fields that start with **X-MS-Exchange-Forest-** in inbound messages.
Receive connector	**Ms-Exch-Accept-Headers-Routing**	This permission affects **Received:** and **Resent-*:** routing header fields in inbound messages.
Send connector	**Ms-Exch-Send-Headers-Organization**	This permission affects organization X-header fields that start with **X-MS-Exchange-Organization-** in outbound messages.
Send connector	**Ms-Exch-Send-Headers-Forest**	This permission affects forest X-header fields that start with **X-MS-Exchange-Forest-** in outbound messages.
Send connector	**Ms-Exch-Send-Headers-Routing**	This permission affects **Received:** and **Resent-*:** routing header fields in outbound messages.

Header firewall for inbound messages on Receive connectors

The following table describes the default application of the header firewall permissions on Receive connectors.

Permission	Default Exchange security principals that have the permission assigned	Permission group that has the security principals as members	Default usage type that assigns the permission groups to the Receive connector
Ms-Exch-Accept-Headers-Organization and **Ms-Exch-Accept-Headers-Forest**	• Hub Transport servers • Edge Transport servers • Exchange Servers **Note:** On Hub Transport servers only	**ExchangeServers**	`Internal`
Ms-Exch-Accept-Headers-Routing	Anonymous user account	**Anonymous**	`Internet`
Ms-Exch-Accept-Headers-Routing	Authenticated user accounts	**ExchangeUsers**	`Client` (unavailable on Edge Transport servers)
Ms-Exch-Accept-Headers-Routing	• Hub Transport servers • Edge Transport servers • Exchange Servers **Note:** Hub Transport servers only • Externally Secured servers	**ExchangeServers**	`Internal`
Ms-Exch-Accept-Headers-Routing	Partner Server account	**Partner**	`Internet` and `Partner`

Return to top

Header firewall on custom Receive connectors

If you want to apply header firewall to messages in a custom Receive connector scenario, use any of the following methods:

- Create the Receive connector with a usage type that automatically applies header firewall to messages. Note that you can set the usage type only when you create the Receive connector.
  - To remove the organization X-headers or forest X-headers from messages, create a Receive connector and select a usage type other than `Internal`.
  - To remove the routing headers from messages, do one of the following actions:
    - Create a Receive connector and select the usage type `Custom`. Don't assign any permission groups to the Receive connector.
    - Modify an existing Receive connector, and set the *PermissionGroups* parameter to the value `None`.

Note that if you have a Receive connector that has no permission groups assigned to it, you need to add security principals to the Receive connector as described in the last step.

- Use the **Remove-ADPermission** cmdlet to remove the **Ms-Exch-Accept-Headers-Organization** permission, the **Ms-Exch-Accept-Headers-Forest** permission, and the **Ms-Exch-Accept-Headers-Routing** permission from a security principal that's configured on the Receive connector. This method doesn't work if the permission has been assigned to the security principal using a permission group on the Receive connector, because you can't modify the permissions assignments or the group membership of a permission group.
- Use the **Add-ADPermission** cmdlet to add the appropriate security principals that are required for mail flow on the Receive connector. Make sure that no security principals have the **Ms-Exch-Accept-Headers-Organization** permission, the **Ms-Exch-Accept-Headers-Forest** permission, and the **Ms-Exch-Accept-Headers-Routing** permission assigned to them. If necessary, use the **Add-ADPermission** cmdlet to deny the **Ms-Exch-Accept-Headers-Organization** permission, the **Ms-Exch-Accept-Headers-Forest** permission, and the **Ms-Exch-Accept-Headers-Routing** permission to the security principals that are configured on the Receive connector.

For more information, see the following topics:

- Receive connectors
- Add-ADPermission
- Remove-ADPermission

Return to top

Header firewall for outbound messages on Send connectors

The following table describes the default application of the header firewall permissions on Send connectors.

Permission	Default Exchange security principals that have the permission assigned	Default usage type that assigns the security principals to the Send

		connector
Ms-Exch-Send-Headers-Organization and **Ms-Exch-Send-Headers-Forest**	• Hub Transport servers • Edge Transport servers • Exchange Servers **Note:** On Hub Transport servers only • Externally Secured servers • **ExchangeLegacyInterop** universal security group	`Internal`
Ms-Exch-Send-Headers-Routing	• Hub Transport servers • Edge Transport servers • Exchange Servers **Note:** On Hub Transport servers only • Externally Secured servers • **ExchangeLegacyInterop** universal security group	`Internal`
Ms-Exch-Send-Headers-Routing	Anonymous User Account	`Internet`
Ms-Exch-Send-Headers-Routing	Partner Servers	`Partner`

Return to top

Header firewall on custom Send connectors

If you want to apply header firewall to messages in a custom Send connector scenario, use the any of the following methods:

- Create the Send connector with a usage type that automatically applies header firewall to messages. Note that you can set the usage type only when you create the Send connector.
  - To remove the organization X-headers or forest X-headers from messages, create a Send connector and select a usage type other than `Internal` or `Partner`.
  - To remove the routing headers from messages, create a Send connector and select the usage type `Custom`. The **Ms-Exch-Send-Headers-Routing** permission is assigned to all Send connector usage types except `Custom`.
- Remove a security principal that assigns the **Ms-Exch-Send-Headers-Organization** permission, the **Ms-Exch-Send-Headers-Forest**, and the **Ms-Exch-Send-Headers-Routing** permission from

the connector.

- Use the **Remove-ADPermission** cmdlet to remove the **Ms-Exch-Send-Headers-Organization** permission, the **Ms-Exch-Send-Headers-Forest** permission, and the **Ms-Exch-Send-Headers-Routing** permission from one of the security principals that's configured on the Send connector.
- Use the **Add-ADPermission** cmdlet to deny the **Ms-Exch-Send-Headers-Organization** permission, the **Ms-Exch-Send-Headers-Forest** permission, and the **Ms-Exch-Send-Headers-Routing** permission to one of the security principals that are configured on the Send connector.

For more information, see the following topics:

- Send connectors
- Add-ADPermission
- Remove-ADPermission

Return to top

Header firewall for other message sources

Messages can enter the transport pipeline on a Mailbox server or an Edge Transport server without using Send connectors or Receive connectors. Header firewall is applied to these other message sources as described in the following list:

- **Pickup directory and Replay directory** The Pickup directory is used by administrators or applications to submit message files. The Replay directory is used to resubmit messages that have been exported from Exchange message queues. For more information about the Pickup and Replay directories, see Pickup directory and Replay directory.

Organization X-headers, forest X-headers, and routing headers are removed from the message files in the Pickup directory.

Routing headers are preserved in messages submitted by the Replay directory.

Whether or not organization X-headers and forest X-headers are preserved or removed from messages in the Replay directory is controlled by the **X-CreatedBy:** header field in the message file:

  - If the value of **X-CreatedBy:** is `MSExchange15`, organization X-headers and forest X-headers are preserved in messages.
  - If the value of **X-CreatedBy:** isn't `MSExchange15`, organization X-headers and forest X-headers are removed from messages.
  - If the **X-CreatedBy:** header field doesn't exist in the message file, organization X-headers and forest X-headers are removed from messages.

- **Drop directory** The Drop directory is used by Foreign connectors on Mailbox servers to send messages to messaging servers that don't use SMTP to transfer messages. For more information about Foreign connectors, see Foreign connectors.

Organization X-headers and forest X-headers are removed from message files before they're put in the Drop directory.

Routing headers are preserved in messages submitted by the Drop directory.

- **Mailbox Transport** The Mailbox Transport service exists on Mailbox servers to transmit

messages to and from mailboxes on Mailbox servers. For more information about the Mailbox Transport service, see Mail flow.

Organization X-headers, forest X-headers, and routing headers are removed from outgoing messages that are sent from mailboxes by the Mailbox Transport Submission service.

Routing headers are preserved for inbound messages to mailbox recipients by the Mailbox Transport Delivery service. The following organization X-headers are preserved for inbound messages to mailbox recipients by the Mailbox Transport Delivery service:

- **X-MS-Exchange-Organization-SCL**
- **X-MS-Exchange-Organization-AuthDomain**
- **X-MS-Exchange-Organization-AuthMechanism**
- **X-MS-Exchange-Organization-AuthSource**
- **X-MS-Exchange-Organization-AuthAs**
- **X-MS-Exchange-Organization-OriginalArrivalTime**
- **X-MS-Exchange-Organization-OriginalSize**

- **DSN messages** Organization X-headers, forest X-headers, and routing headers are removed from the original message or the original message header that's attached to the DSN message. For more information about DSN messages, see DSNs and NDRs.
- **Transport agent submission** Organization X-headers, forest X-headers, and routing headers are preserved in messages that are submitted by transport agents.

Return to top

Organization X-headers and forest X-headers in Exchange

The Transport service on Mailbox servers or Edge Transport servers insert custom X-header fields into the message header.

Organization X-headers start with **X-MS-Exchange-Organization-**. Forest X-headers start with **X-MS-Exchange-Forest-**. The following table describes some of the organization X-headers and forest X-headers that are used in messages in an Exchange organization.

X-header	Description
X-MS-Exchange-Forest-RulesExecuted	Transport rules that acted on the message.
X-MS-Exchange-Organization-Antispam-Report	A summary of the anti-spam filter results that have been applied to the message by the Content Filter agent.
X-MS-Exchange-Organization-AuthAs	Specifies the authentication source. This X-header is always present when the security of a message has been evaluated. The possible values are `Anonymous`, `Internal`, `External`, or

	`Partner`.
X-MS-Exchange-Organization-AuthDomain	Populated during Domain Secure authentication. The value is the FQDN of the remote authenticated domain.
X-MS-Exchange-Organization-AuthMechanism	Specifies the authentication mechanism for the submission of the message. The value is a 2-digit hexadecimal number.
X-MS-Exchange-Organization-AuthSource	Specifies the FQDN of the server computer that evaluated the authentication of the message on behalf of the organization.
X-MS-Exchange-Organization-Journal-Report	Identifies journal reports in transport. As soon as the message leaves the transport service, the header becomes **X-MS-Journal-Report**.
X-MS-Exchange-Organization-OriginalArrivalTime	Identifies the time when the message first entered the Exchange organization.
X-MS-Exchange-Organization-Original-Sender	Identifies the original sender of a quarantined message when it first entered the Exchange organization.
X-MS-Exchange-Organization-OriginalSize	Identifies the original size of a quarantined message when it first entered the Exchange organization.
X-MS-Exchange-Organization-Original-Scl	Identifies the original SCL of a quarantined message when it first entered the Exchange organization.
X-MS-Exchange-Organization-PCL	Identifies the phishing confidence level. The possible phishing confidence level values are from 1 through 8. A larger value indicates a suspicious message. For more information, see

	Anti-spam stamps.
X-MS-Exchange-Organization-Quarantine	Indicates the message has been quarantined in the spam quarantine mailbox and a delivery status notification (DSN) has been sent. Alternatively, it indicates that the message was quarantined and released by the administrator. This X-header field prevents the released message from being submitted to the spam quarantine mailbox again. For more information, see Release quarantined messages from the spam quarantine mailbox.
X-MS-Exchange-Organization-SCL	Identifies the SCL of the message. The possible SCL values are from 0 through 9. A larger value indicates a suspicious message. The special value -1 exempts the message from processing by the Content Filter agent. For more information, see Content filtering.
X-MS-Exchange-Organization-SenderIdResult	Contains the results of the Sender ID agent. The Sender ID agent uses the sender policy framework (SPF) to compare the message's source IP address to the domain that's used in the sender's email address. The Sender ID results are used to calculate the SCL of a message. For more information, see Sender ID.

Return to top

Domains

Exchange Server 2013 > Mail flow >

Applies to: *Office 365*

Topic Last Modified: *2012-10-11*

Domains represent SMTP namespaces for which email directories and mailboxes are set up. By configuring the domains that interact with your Microsoft Exchange Server 2013 organization, you can configure how email to and from various domains is processed by Exchange.

Accepted domains

An accepted domain is any SMTP namespace for which your Exchange organization sends or receives email. Accepted domains include those domains for which the Exchange organization is authoritative, as well as internal relay domains and external relay domains. An Exchange organization is authoritative when it handles mail delivery for recipients in the accepted domain. Accepted domains also include domains for which the Exchange organization receives mail and then relays it to an email server that's outside the organization.

For more information about accepted domains, see Accepted domains.

Remote domains

In Exchange 2013, you create remote domain entries to define the settings for message transfer between your Exchange organization and domains outside of your organization. When you create a remote domain entry, you control the types of messages that are sent to that domain. You can also apply message format policies and acceptable character sets for messages that are sent from users in your organization to the remote domain.

Settings for remote domains are global configuration settings for your Exchange organization. Remote domain settings are applied to messages during categorization. When recipient resolution occurs, the recipient domain is matched against the configured remote domains. If a remote domain configuration blocks a specific message type from being sent to recipients in that domain, the message is deleted. If you specify a particular message format for the remote domain, the message headers and content are modified. The settings apply to all messages that are processed by the Exchange organization.

For more information about remote domains, see Remote domains.

Accepted domains

Exchange Server 2013 > Mail flow > Domains >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-17*

An accepted domain is any SMTP namespace for which a Microsoft Exchange Server 2013

organization sends or receives email. Accepted domains include those domains for which the Exchange organization is authoritative. An Exchange organization is authoritative when it handles mail delivery for recipients in the accepted domain. Accepted domains also include domains for which the Exchange organization receives mail and then relays it to an email server that's outside the organization for delivery to the recipient.

Contents

Configuring accepted domains

Accepted domains are configured as global settings for the Exchange organization. You need to configure every domain for which your Exchange organization relays or delivers messages as an accepted domain in your organization.

There are three types of accepted domains: authoritative, internal relay, and external relay. These accepted domain types are described in the following sections.

Note:

If you have a subscribed Edge Transport server in your perimeter network, you configure accepted domains on a Mailbox server in your Exchange organization. The accepted domains configuration is replicated to the Edge Transport server during EdgeSync synchronization. For more information, see Edge Subscriptions.

Authoritative domains

An organization may have more than one SMTP domain. The set of email domains for an organization are the authoritative domains. In Exchange 2013, an accepted domain is considered authoritative when the Exchange organization hosts mailboxes for recipients in this SMTP domain.

By default, when the first Exchange 2013 Mailbox server is installed, one accepted domain is configured as authoritative for the Exchange organization. The default accepted domain is the fully qualified domain name (FQDN) for your forest root domain. Frequently, the internal domain name differs from the external domain name. For example, your internal domain name may be contoso.local, although your external domain name is contoso.com. The DNS mail exchanger (MX) record for your organization references contoso.com. Contoso.com is the SMTP namespace that you assign to users when you create an email address policy. You need to create an accepted

domain to match your external domain name.

To learn more, see:

- Configure an accepted domain within your Exchange organization as authoritative
- Configure Exchange to accept mail for multiple authoritative domains

Relay domains

Typically, most Internet-facing messaging servers are configured to not allow for other domains to be relayed through them. However, there are scenarios where you may want to let partners or subsidiaries relay email through your Exchange servers. In Exchange 2013, you can configure accepted domains as relay domains. Your organization receives the email messages and then relays the messages to another email server.

You can configure a relay domain as an internal relay domain or as an external relay domain. These two relay domain types are described in the following sections.

Internal relay domain

When you configure an internal relay domain, some or all of the recipients in this domain don't have mailboxes in this Exchange organization. Mail from the Internet is relayed for this domain through Transport servers in this Exchange organization. This configuration is used in the scenarios that are described in this section.

An organization may have to share the same SMTP address space between two or more different messaging systems. For example, you may have to share the SMTP address space between Exchange and a third-party messaging system, or between Exchange environments that are configured in different Active Directory forests. In these scenarios, users in each email system have the same domain suffix as part of their email addresses.

To support these scenarios, you need to create an accepted domain that's configured as an internal relay domain. You also need to add a Send connector that's sourced on a Mailbox server and configured to send email to the shared address space. If an accepted domain is configured as authoritative and a recipient isn't found in Active Directory, a non-delivery report (NDR) is returned to the sender. The accepted domain that's configured as an internal relay domain first tries to deliver to a recipient in the Exchange organization. If the recipient isn't found, the message is routed to the Send connector that has the closest address space match.

If an organization contains more than one forest and has configured global address list (GAL) synchronization, the SMTP domain for one forest may be configured as an internal relay domain in a second forest. Messages from the Internet that are addressed to recipients in internal relay domains are relayed to the Mailbox servers in the same organization. The receiving Mailbox servers then route the messages to the Mailbox servers in the recipient forest. You configure the SMTP domain as an internal relay domain to make sure that email that's addressed to that domain is accepted by the Exchange organization. The connector configuration of your organization

determines how messages are routed.

To learn more, see Configure an accepted domain for a business unit with mailboxes outside your Exchange organization.

External relay domain

When you configure an external relay domain, messages are relayed to an email server that's outside your Exchange organization and outside the organization's network perimeter.

For more information, see Configure an accepted domain for an independent business unit.

Accepted domains and email address policies

You need to configure an accepted domain before that SMTP address space can be used in an email address policy. When you create an accepted domain, you can use a wildcard character (*) in the address space to indicate that all subdomains of the SMTP address space are also accepted by the Exchange organization. For example, to configure contoso.com and all its subdomains as accepted domains, enter *.contoso.com as the SMTP address space. The accepted domain entries are automatically available for use in an email address policy.

If you delete an accepted domain that's used in an email address policy, the policy is no longer valid, and recipients with email addresses in that SMTP domain will be unable to send or receive email.

Configure an accepted domain within your Exchange organization as authoritative

Mail flow > Domains > Accepted domains >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-17*

If a domain belonging to your organization hosts mailboxes for all the recipients within an SMTP namespace, that domain is considered to be authoritative. By default, one accepted domain is configured as authoritative for the Exchange organization. If your organization has more than one SMTP namespace, you can configure more than one accepted domain as authoritative.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Accepted domains" entry in the Mail flow permissions topic.
- If you have a subscribed Edge Transport server in your perimeter network, you configure accepted domains on a Mailbox server in your Exchange organization. The accepted domains configuration is replicated to the Edge Transport server during EdgeSync synchronization. For more information, see Edge Subscriptions.
- You can't create an accepted domain that has the same name as an already configured remote domain. For example, if you have fabrikam.com configured as a remote domain, you can't create an accepted domain for fabrikam.com.
- Before you configure an accepted domain, you must verify that a public Domain Name System (DNS) MX resource record for that SMTP namespace exists and that the MX resource record references a server name and an IP address associated with your Exchange organization.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to configure an accepted domain as authoritative

If an accepted domain for your Exchange organization hosts all the mailboxes for recipients within that domain's SMTP namespace, you may want to configure it as an authoritative domain.

1. In the EAC, navigate to **Mail flow** > **Accepted domains**, and click **Add +**.
2. In the **Name** field, enter the display name for the accepted domain. Each accepted domain for your organization must have a unique display name This may be different than the accepted domain. For example, the domain contoso.com could have a display name of Contoso Local Accepted Domain.
3. In the **Accepted domain** field, enter the accepted domain. Specify an SMTP namespace for which your organization accepts email messages. (for example, Contoso.com).
4. Select **Authoritative domain**. This option is for email relayed to servers within your Exchange organization for an accepted domain that hosts mailboxes for all the recipients within an SMTP namespace.
5. Click **Save**.

Tip:
To configure an accepted domain that has already been created, select the domain from the

accepted domains list and click **Edit** . You can configure more than one domain as authoritative.

How do you know this worked?

Your new accepted domain will appear in the accepted domains list in the EAC. To verify that you have successfully configured the accepted domain as authoritative, send mail to the domain and verify that it is received.

Configure an accepted domain for an independent business unit

Mail flow > Domains > Accepted domains >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-17*

In some situations you may want to configure an accepted domain for an independent business unit with email servers outside your Exchange organization. In such scenarios, you can configure the accepted domain as an external relay domain.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Accepted domains" entry in the Mail flow permissions topic.
- If you have a subscribed Edge Transport server in your perimeter network, you configure accepted domains on a Mailbox server in your Exchange organization. The accepted domains configuration is replicated to the Edge Transport server during EdgeSync synchronization. For more information, see Edge Subscriptions.
- You can't create an accepted domain that has the same name as an already configured remote domain. For example, if you have fabrikam.com configured as a remote domain, you can't create an accepted domain for fabrikam.com.
- Before you configure an accepted domain, you must verify that a public Domain Name System (DNS) MX resource record for that SMTP namespace exists and that the MX resource record references a server name and an IP address associated with your Exchange organization.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to configure the an accepted domain as an external relay domain

You may want to configure an accepted domain for a business unit with email servers outside your Exchange organization.

1. In the EAC, navigate to **Mail flow** > **Accepted domains**, select the domain you wish to configure, and click **Edit** .
2. In the **Name** field, enter the display name for the accepted domain. Each accepted domain for your organization must have a unique display name. This may be different than the accepted domain. For example, the domain Contoso.com could have a display name of Contoso Local Accepted Domain.
3. Select **External Relay Domain**. This option is for email is relayed to a server outside your Exchange organization.
4. Click **Save**.

How do you know this worked?

To verify that you have successfully configured an accepted domain as an external relay domain, send a message from the accepted domain you've configured as an external relay domain, and verify that it is received.

Configure an accepted domain for a business unit with mailboxes outside your Exchange organization

Mail flow > Domains > Accepted domains >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-06*

In some instances, you may want to configure an accepted domain for a business unit in which some or all recipients in the domain don't have mailboxes in your Exchange organization. This can occur, for example, when an organization shares the same SMTP address space between two or

more different email systems. For such scenarios, you can configure an accepted domain as an internal relay domain.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Accepted domains" entry in the Mail flow permissions topic.
- If you have a subscribed Edge Transport server in your perimeter network, you configure accepted domains on a Mailbox server in your Exchange organization. The accepted domains configuration is replicated to the Edge Transport server during EdgeSync synchronization. For more information, see Edge Subscriptions.
- Before you configure an accepted domain, you must verify that a public Domain Name System (DNS) MX resource record for that SMTP namespace exists and that the MX resource record references a server name and an IP address associated with your Exchange organization.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to configure an accepted domain as an internal relay domain

1. In the EAC, navigate to **Mail flow** > **Accepted domains**, select the domain you wish to configure, and then click **Edit** .
2. In the **Name** field, enter the display name for the accepted domain. Each accepted domain for your organization must have a unique display name. This may be different than the accepted domain. For example, the domain Contoso.com could have a display name of Contoso Local Accepted Domain.
3. Select **Internal Relay Domain**.
4. Click **Save**.

How do you know this worked?

To verify that you have successfully configured an accepted domain as an internal relay domain, send a message from the internal relay domain to a mailbox within your Exchange organization and verify that it is received.

Configure Exchange to accept mail for multiple authoritative domains

Mail flow > Domains > Accepted domains >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-17*

In Microsoft Exchange Server 2013, it's easy to add multiple authoritative domains to your organization. However, after you add the authoritative domain, you need to decide how to use the authoritative domain in your organization. For example:

- If you want to add an email address in the new authoritative domain for recipients in your organization, do you want to replace the existing primary (reply to) address for the recipients, or add the new email address as a proxy (secondary) address?
- Do you want the email address in the new authoritative domain to apply to all recipients, and all recipient types? Or do you want the new email address to apply to specific types of recipients, or specific recipients based on their user properties, for example, only users in the Finance department?

The following examples are scenarios in which your Exchange organization may have to receive and process email for more than one authoritative SMTP domain:

- You are changing your SMTP domain name, but have to continue to accept email for the old domain name for a time, in case customers send email messages to the previous email addresses. You can set the new email address as the primary (reply to) address. This means that the new address will be the default email address displayed on all email messages sent by the recipient. You can set the old email address as a secondary address. This will enable the recipient to continue to receive email sent to the old email address.
- You want to provision different email addresses for business units within your organization. For example, if the contoso.com Active Directory forest contains subdomains for the subsidiaries Tailspin Toys and Fourth Coffee, you may want to assign the SMTP domain names contoso.com, tailspintoys.com, and fourthcoffee.com to the recipients in those respective business units.
- You provide email hosting services and have to accept email for more than one SMTP domain name.

What do you need to know before you begin?

- Estimated time to complete this task: 30 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Accepted domains" entry in the Mail flow permissions topic and the "Email address policies" entry in the Recipients Permissions topic.
- If you have a subscribed Edge Transport server in your perimeter network, you configure accepted

domains on a Mailbox server in your Exchange organization. The accepted domains configuration is replicated to the Edge Transport server during EdgeSync synchronization. For more information, see Edge Subscriptions.

- When you create an accepted domain, you can use a wildcard character (*) in the address space to indicate that all subdomains of the SMTP address space are also accepted by the Exchange organization. For example, to configure contoso.com and all its subdomains as accepted domains, enter ***.contoso.com** as the SMTP address space. However, if the subdomain names will be used in an email address policy, each subdomain must have an explicit accepted domain entry.
- An MX record in public DNS is required for each SMTP domain for which you accept email from the Internet. Each MX record should resolve to the Internet-facing server that receives email for your organization.
- You need to configure Send connectors and Receive connectors so your Exchange organization can send email to and receive email from the Internet. The configuration of the Internet Send connectors and Receive connectors is determined by your Exchange topology. For more information about configuring connectors, see Connectors.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Create an authoritative domain

Use the Exchange Administration Center to create an authoritative domain

1. In the EAC, navigate to **Mail flow** > **Accepted domains**, and click **Add +**.
2. In the **Name** field, enter the display name for the accepted domain. Each accepted domain for your organization must have a unique display name. This may be different than the accepted domain. For example, the domain contoso.com could have a display name of Contoso Local Accepted Domain.
3. In the **Accepted domain** field, specify an SMTP namespace for which your organization accepts email messages. For example, contoso.com.
4. Select **Authoritative domain**.
5. Click **Save**.

Use the Shell to create an authoritative domain

To create a new authoritative domain, use the following syntax.

```
New-AcceptedDomain -Name "<Unique Name>" -DomainName <SMTP
```

```
domain> -DomainType Authoritative
```

For example, to create a new authoritative domain named "Fourth Coffee subsidiary" for the domain fourthcoffee.com, run the following command:

```
New-AcceptedDomain -Name "Fourth Coffee subsidiary" -
DomainName fourthcoffee.com -DomainType Authoritative
```

How do you know this step worked?

To verify that you have successfully created an authoritative domain, do one of the following:

- In the EAC, navigate to **Mail flow** > **Accepted domains**. Verify the accepted domain you created is displayed, and the **Domain Type** value is **Authoritative**.
- In the Shell, run the command **Get-AcceptedDomain**. Verify the domain you created is displayed, and the **DomainType** value is `Authoritative`.

Step 2: Configure an email address policy for the authoritative domain

To use the authoritative accepted domain you created, you need to configure an email address policy for the authoritative domain that meets the objectives of your scenario. For example, you may need to create a new email address policy, or modify an existing policy. You may elect to replace the primary email address for some or all of your recipients, and you can elect to keep or remove the old primary email address. Two example scenarios are presented in this section.

Change the existing primary email address

To change the primary (reply to) email address assigned to recipients and keep the old primary email address as a proxy (secondary) email address, follow these steps:

Use the EAC to change the existing primary email address

1. In the EAC, navigate to **Mail flow** > **Email address policies**. Select the email address policy you want to modify, and click **Edit** .
2. On the **Email Address Policy** page, click the **Email address format** tab. In the **Email address format** section, click **Add+**.
3. On the **Email Address Format** page that appears, make the following selections:
   - **Select an accepted domain** Click the drop-down list, and select the new authoritative domain.
   - Select **Make this format the reply email address**.

When you are finished, click **Save**.

4. On the **Email Address Policy** page, click **Save** to save your changes to the policy.
5. You'll get a warning that the email address policy won't be applied until you update it. After it's created, select it, and then, in the details pane, click **Apply**.

Use the Shell to change the existing primary email address

In the Shell, you use two separate commands: one command to modify the existing email address policy, and another command to apply the updated email address policy to the recipients in your organization.

To change the existing primary email address, and keep the old primary email address as a proxy address, run the following command:

```
Set-EmailAddressPolicy <EmailAddressPolicyIdentity> -
EnabledEmailAddressTemplates
SMTP:<NewPrimaryEmailAddress>,smtp:<OldPrimaryEmailAddress>
```

For example, suppose the email address policy in your organization uses the email addresses format *useralias*`@contoso.com`. This example changes the domain of primary (reply to) address in the email address policy named "Default Policy" to `@fourthcoffee.com`, and keeps the old primary reply address in the `@contoso.com` domain as a proxy (secondary) address.

```
Set-EmailAddressPolicy "Default Policy" -
EnabledEmailAddressTemplates
SMTP:@fourthcoffee.com,smtp:@contoso.com
```

Note:

The `SMTP` qualifier in uppercase lettersspecifies the primary (reply to) address. The `smtp` qualifier in lowercase letters specifies a proxy (secondary) address.

To apply the updated email address policy to recipients, use the following syntax.

```
Update-EmailAddressPolicy <EamilAddressPolicyIdentity>
```

For example, to apply the updated email address policy named "Default Policy", run the following command:

```
Update-EmailAddressPolicy "Default Policy"
```

Replace the existing primary email address for a filtered set of recipients

You can't modify the default email address policy to apply to a filtered set of recipients. You need to create a new email address policy, or modify an existing custom email address policy. The examples in this section create a new email address policy. In these examples, the primary (reply to) address in the new accepted domain replaces the old primary address for the specified recipients without keeping the old primary address as a proxy (secondary) email address. Therefore, the affected recipients can no longer receive email at their old primary email address.

Also, email address policies that apply to specific users should have a higher priority (indicated by a lower integer value) than other email address policies, including the default policy, so the specific

policy is applied first. Because two policies can't have the same priority value, you may first need lower the priority of your organization's default email address policy.

Use the EAC to replace the existing primary email address for a filtered set of recipients

To create additional email addresses that will be used as the primary email address for a filtered set of recipients, follow these steps.

1. In the EAC, navigate to **Mail flow** > **Email address policies**, and then click **Add +**.
2. On the **Email Address Policy** page, complete the following fields:
   a. **Policy name** Enter a unique, descriptive name.
   b. **Email address format** Click **Add +**. On the **Email Address Format** page that appears, make the following selections:
      - **Select an accepted domain** Click the drop-down list, and select the new authoritative domain.
      - **Email address format** Select the appropriate email address format for your organization.
      - Select **Make this format the reply email address**.

   When you are finished, click **Save**.
3. **Run this policy in this sequence with other policies** Typically, policies that apply to specific users should have a higher priority (indicated by a lower integer value) than other email address policies, including the default policy.
4. **Specify the types of recipients this email address will apply to** Select the recipient types to which you want the email address policy applied.
5. **Create rules to further define the recipients that this email address policy applies to** Click **Add a rule** to restrict the recipients that this policy will apply to. This creates a Boolean **And** statement. Repeat this step as many times as necessary.

> **Caution:**
> If you apply too many rules, it's possible to restrict the email address policy to the point that it doesn't contain any users.

6. Click **Preview recipients the policy applies to** to view the recipients that policy will apply to.
7. Click **Save** to save your changes and create the policy.
8. You'll get a warning that the email address policy won't be applied until you update it. After it's created, select it, and then, in the details pane, click **Apply**.

Use the Shell to replace the existing primary email address for a filtered set of recipients

To replace the primary email address for a filtered set of recipients, use the following command:

```
New-EmailAddressPolicy -Name <Policy Name> -Priority
<Integer> -IncludedRecipients <RecipientTypes> <Conditional
Recipient Properties> -EnabledEmailAddressTemplates
SMTP:@<NewPrimaryEmailAddress>
```

This example creates an email address policy named "Fourth Coffee Recipients", assigns that policy to mailbox users in the Fourth Coffee department, and sets the highest priority for that email

address policy so the policy is applied first. Note that the old primary email address isn't preserved for these recipients, so they can't receive email at their old primary email address.

```
New-EmailAddressPolicy -Name "Fourth Coffee Recipients" -
Priority 1 -IncludedRecipients MailboxUsers -
ConditionalDepartment "Fourth Coffee" -
EnabledEmailAddressTemplates SMTP:@fourthcoffee.com
```

To apply the new email address policy to the affected recipients, run the following command:.

```
Update-EmailAddressPolicy "Fourth Coffee Recipients"
```

How do you know this step worked?

To verify that you have successfully configured an email address policy for the authoritative domain, do one of the following:

- In the EAC, navigate to **Mail flow** > **Email address policies**. Verify the policies are applied in the correct order. Also, select any new or updated policies, and in the details pane, verify the email address format, included recipients, and if the policy has been applied,
- In the Shell, run the commands **Get-EmailAddressPolicy** and `Get-EmailAddressPolicy "<Policy Name>"| Format-List` to verify the details of the policies.

How do you know this task worked?

To verify that you have configured Exchange to accept mail for multiple authoritative domains, do the following:

1. Send test messages to an affected recipient from a mailbox outside your Exchange organization. Verify the email addresses that successfully accept mail.
2. Send test messages from an affected recipient mailbox to an external recipient, and verify the From address of the message.

Remote domains

Exchange Server 2013 > Mail flow > Domains >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-11*

You can create remote domain entries to define the settings for message transfer between the Microsoft Exchange Server 2013 organization and domains outside your Exchange organization. When you create a remote domain entry, you control the types of messages that are sent to that domain. You can also apply message format policies and acceptable character sets for messages

that are sent from users in your organization to the remote domain. The settings for remote domains are global configuration settings for the Exchange organization.

The remote domain settings are applied to messages during categorization in the Transport service on Mailbox servers. When recipient resolution occurs, the recipient domain is matched against the configured remote domains. If a remote domain configuration blocks a specific message type from being sent to recipients in that domain, the message is deleted. If you specify a particular message format for the remote domain, the message headers and content are modified. The settings apply to all messages that are processed by the Exchange organization.

Note:

If you configure message settings per user, the per-user settings override the organizational configuration.

By default, there's a single remote domain entry. The domain address space is configured as an asterisk (*). This represents all remote domains. If you don't create additional remote domain entries, all messages that are sent to all recipients in all remote domains have the same settings applied to them.

When you configure remote domains, you can prevent certain types of messages from being sent to that domain. These message types include out-of-office messages, auto-reply messages, non-delivery reports (NDRs), and meeting forward notifications. If you have a multiple forest environment, you may want to allow the sending of those types of messages to those domains. However, if you have identified a domain from which spam originates, you may want to block sending of those types of messages to those remote domains.

Contents

Message format

You can specify the message format and the character set to use for email messages that are sent to remote domains. These settings can be useful to make sure that email sent by senders in your domain to the remote domain is compatible with the receiving email system. For example, if you know that the remote domain's messaging system is Exchange, you can specify to always use Exchange rich text format (RTF). For more information, see Content conversion.

Automatic replies settings

In Exchange 2013, users can specify different automatic replies for internal and external recipients. Furthermore, the types of automatic replies available in your organization also depend on the Microsoft Outlook version in use.

In Exchange 2013, there are two types of automatic replies:

- **External** Supported by Exchange 2013 and Exchange 2010. Can only be set by Outlook 2010 or Office Outlook 2007, or using Microsoft Office Outlook Web App.
- **Internal** Supported by Exchange 2013 and Exchange 2010. Can only be set by Outlook 2010 or Outlook 2007, or using Outlook Web App.

The following table describes various client and server combinations and the types of automatic replies that can be used in each scenario.

Client and server support for automatic replies

Client version	Exchange version	Automatic replies supported
Outlook 2010 or Outlook 2007	Exchange 2013 Exchange 2010 Exchange 2007	Internal, External
Outlook Web App	Exchange 2013 Exchange 2010 Exchange 2007	Internal, External

Controlling NDR information

As mentioned at the beginning of this topic, you can prevent NDRs from being sent to a remote domain. By blocking NDRs from being sent to a remote domain, you can prevent the information contained within the NDR message from leaving your organization, thereby limiting the knowledge that a malicious user can obtain about your organization. However, this also prevents legitimate senders from receiving NDRs, resulting in confusion and lost productivity.

Exchange 2013 provides you with granular control over the contents of an NDR destined for a remote domain. With Exchange 2013, you can allow NDRs to a remote domain, while stripping any diagnostic information. This way, you can still prevent information about your Exchange deployment from leaving your organization while at the same time providing NDR notifications to external senders.

This feature is controlled with the *NDRDiagnosticInfoEnabled* parameter on the **Set-RemoteDomain** cmdlet. Because this setting is configurable for each remote domain, you can have different settings based on your needs. For example, you can choose to remove the NDR diagnostic information for the default remote domain, but allow full NDR diagnostic information for the remote domains that represent your partners.

For more information about this new setting, see Set-RemoteDomain.

Manage remote domains

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-18*

Remote domains are SMTP domains that are external to your Microsoft Exchange organization. You can create remote domain entries to define the settings for message transferred between your Exchange organization and specific external domains. The settings in the remote domain entry for a specific external domain override the settings in the default remote domain that normally apply to all external recipients. The remote domain settings are global for the Exchange organization.

If you remove a remote domain entry, the settings for message transfer no longer apply to messages sent to the remote domain. Removing a remote domain entry doesn't disable mail flow to the remote domain. After a remote domain entry is removed, the configuration settings of the default remote domain apply to new messages sent to that domain. You can't remove the default remote domain.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Remote domains" entry in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.
- You can't create a remote domain for an address space that's configured as an accepted domain in your organization. For example, if your organization accepts mail for fabrikam.com, you can't create a remote domain for fabrikam.com.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What Do You Want to Do?

Use the Shell to create a remote domain

To create a new remote domain entry, use the following syntax.

```
New-RemoteDomain -Name <Descriptive Name> -DomainName <SMTP address space>
```

This example creates a remote domain entry for messages sent to the contoso.com domain.

```
New-RemoteDomain -Name Contoso -DomainName contoso.com
```

This example creates a remote domain entry for messages sent to the fabrikam.com domain and all subdomains.

```
New-RemoteDomain -Name Fabrikam -DomainName *.fabrikam.com
```

How do you know this worked?

To verify that you have successfully created a remote domain, do the following:

1. Run the command **Get-RemoteDomain** and verify that the remote domain is listed.
2. Run the command `Get-RemoteDomain <Remote Domain Name> | Format-List` to verify the settings on the new remote domain. Send a test message to a recipient in the address space that's specified in the new remote domain entry, and verify that the message settings match those specified by the new remote domain entry.

Use the Shell to configure a remote domain

You configure the settings in the remote domain entry using the **Set-RemoteDomain** cmdlet. There are many different settings that relate to automatic replies, message format and encoding, and other message settings. For more information, see Set-RemoteDomain.

To configure remote domains for specific scenarios, see the following topics:

- Configure remote domain out of office replies
- Configure remote domain automatic replies
- Configure remote domain message reporting

Use the Shell to remove a remote domain

To remove a remote domain entry, use the following syntax.

```
Remove-RemoteDomain <RemoteDomainName>
```

This example removes the remote domain entry named Contoso

```
Remove-RemoteDomain Contoso
```

How do you know this worked?

To verify that you have successfully removed the remote domain, do the following:

1. Run the command **Get-RemoteDomain** and verify that the remote domain is isn't listed.
2. Run the command `Get-RemoteDomain Default | Format-List` to verify the settings on the default remote domain. Send a test message to a recipient in the address space that was specified in the removed remote domain, and verify that the message settings match those specified by the default remote domain.

Configure remote domain out of office replies

Mail flow > Domains > Remote domains >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-11*

You can use the Exchange Management Shell to configure the way emails are sent and received through remote domains. The following demonstrates how to use the Exchange Management Shell to configure the way Exchange handles out of office replies.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes
- You can only use the Shell to perform this procedure.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Remote domains" entry in the Mail flow permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to configure out-of-office replies

You can use the **Set-RemoteDomain** cmdlet to configure the properties of a remote domain.

This example disables out-of-office messages for the remote domain named Contoso.

```
Set-RemoteDomain Contoso -AllowedOOFType None
```

This example allows only external out-of-office messages.

```
Set-RemoteDomain Contoso -AllowedOOFType External
```

Configure remote domain automatic

replies

Mail flow > Domains > Remote domains >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-26*

You can use the Exchange Management Shell to configure the way emails are sent and received through remote domains. The following demonstrates how to use the Exchange Management Shell to configure the way Exchange handles automatic replies.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes
- You can only use the Shell to perform this procedure.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Remote domain" entry in the Mail flow permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to configure automatic replies

You can use the **Set-RemoteDomain** cmdlet to configure the properties of a remote domain.

This example allows automatic replies to the remote domain named Contoso. This setting is disabled by default.

```
Set-RemoteDomain Contoso -AutoReplyEnabled $true
```

This example allows automatic forwards to the remote domain. This setting is disabled by default.

```
Set-RemoteDomain Contoso -AutoForwardEnabled $true
```

Configure remote domain message reporting

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-26*

You can use the Exchange Management Shell to configure the way emails are sent and received through remote domains. The following demonstrates how to use the Exchange Management Shell configure the way Exchange handles delivery and non-delivery reports.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes
- You can only use the Shell to perform this procedure.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Remote domains" entry in the Mail flow permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to configure message reporting

You can use the **Set-RemoteDomain** cmdlet to configure the properties of a remote domain.

This example disables delivery reports to the remote domain named Contoso. This setting is enabled by default.

```
Set-RemoteDomain Contoso -DeliveryReportEnabled $false
```

This example disables non-delivery reports to the remote domain. This setting is enabled by default.

```
Set-RemoteDomain Contoso -NDREnabled $false
```

Supported character sets for remote domains

Mail flow > Domains > Remote domains >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-21*

The following character sets can be specified for messages sent to remote domains.

- In the Exchange admin center (EAC), on the **Remote domain** settings page, select the name from the **MIME character set** and **Non-MIME character set** drop-down lists.
- In the Shell, use the value in the Name column in the following table for the *CharacterSet* parameter or *NonMimeCharacterSet* parameter in the Set-RemoteDomain cmdlet.

Supported character sets for remote domain configuration

Name	Description
big5	Chinese Traditional (Big5)
DIN_66003	German (IA5)
euc-jp	Japanese (EUC)
euc-kr	Korean (EUC)
GB18030	Chinese Simplified (GB18030)
gb2312	Chinese Simplified (GB2312)
hz-gb-2312	Chinese Simplified (HZ)
iso-2022-jp	Japanese (JIS)
iso-2022-kr	Korean (ISO)
iso-8859-1	Western European (ISO)
iso-8859-2	Central European (ISO)
iso-8859-3	Latin 3 (ISO)
iso-8859-4	Baltic (ISO)
iso-8859-5	Cyrillic (ISO)
iso-8859-6	Arabic (ISO)
iso-8859-7	Greek (ISO)

iso-8859-8	Hebrew (ISO)
iso-8859-9	Turkish (ISO)
iso-8859-13	Estonian (ISO)
iso-8859-15	Latin 9 (ISO)
koi8-r	Cyrillic (KOI8-R)
koi8-u	Cyrillic (KOI8-U)
ks_c_5601-1987	Korean (Windows)
NS_4551-1	Norwegian (IA5)
SEN_850200_B	Swedish (IA5)
shift_jis	Japanese (Shift-JIS)
utf-8	Unicode (UTF-8)
windows-1250	Central European (Windows)
windows-1251	Cyrillic (Windows)
windows-1252	Western European (Windows)
windows-1253	Greek (Windows)
windows-1254	Turkish (Windows)
windows-1255	Hebrew (Windows)
windows-1256	Arabic (Windows)
windows-1257	Baltic (Windows)
windows-1258	Vietnamese (Windows)
windows-874	Thai (Windows)

Transport agents

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-08*

Transport agents let you install custom software that is created by Microsoft, by third-party vendors, or by your organization, on an Exchange server. This software can then process email messages that pass through the transport pipeline. In Microsoft Exchange Server 2013, the transport pipeline is made of the following processes:

- The Front End Transport service on Client Access servers
- The Transport service on Mailbox servers
- The Mailbox Transport service on Mailbox servers
- The Transport service on Edge Transport servers

For more information about the transport pipeline, see Mail flow

Like the previous version of Exchange, Exchange 2013 transport provides extensibility through the Microsoft Exchange Server 2013 Transport Agents SDK. The Exchange 2013 version of the SDK is based on the Microsoft .NET Framework version 4.0 and allows third parties to implement the following predefined classes:

- **SmtpReceiveAgent**
- **RoutingAgent**
- **DeliveryAgent**

When complied against libraries in the SDK, the resulting assemblies are registered with Exchange 2013, which loads the agents and invokes their event handlers during specific stages of the SMTP sessions or message processing. These stages, or events, are part of the agent definitions. The agent registration information is stored in an XML configuration file.

The following list explains the requirements for using transport agents in Exchange 2013.

- The Transport service on Mailbox servers and Edge Transport servers fully supports all the predefined classes in the SDK, and therefore any third party transport agents written for the Hub Transport or Edge Transport server roles in Microsoft Exchange Server 2010 should work in the Transport service in Exchange 2013.
- The Front End Transport service only supports the **SmtpReceiveAgent** class in the SDK, and third party agents can't operate on the **OnEndOfData** SMTP event.
- The Mailbox Transport service doesn't support the SDK at all, so you can't use any third party agents in the Mailbox Transport service.

Support for legacy transport agents based on versions of the .NET Framework prior to version 4.0 isn't enabled by default, but you can enable it. For instructions, see Enable support for legacy transport agents.

Contents

Updates to transport agent management

Due to the updates to the Exchange 2013 transport pipeline, the transport agent cmdlets need to distinguish between the Transport service and the Front End Transport service, especially if the Client Access server and the Mailbox server are installed on the same computer. For more information, see Manage transport agents.

Transport Agent management cmdlets manipulate a configuration file located at `%ExchangeInstallPath%TransportRoles\Shared`. For the Transport service on Mailbox servers and Edge Transport servers, the file is `agents.config`. For the Front End Transport service on Client Access servers, the file is `fetagents.config`. Both files use the same format as in Exchange 2010. For more information about managing transport agents, see Manage transport agents.

Return to top

Transport agents and SMTP events

Transport agents use SMTP events. These events are triggered as messages move through the transport pipeline. SMTP events give transport agents access to messages at specific points during the SMTP conversation and during routing of messages through the organization.

Note that there are new SMTP Receive events in Exchange 2013. SMTP Receive exists in the Front End Transport service on Client Access servers, the Transport service on Mailbox servers and Edge Transport servers and the Mailbox Transport Delivery service on Mailbox servers. The categorizer exists only in the Transport service on Mailbox servers and Edge Transport servers. For more information about transport services and the categorizer, see Mail routing.

The following tables list the SMTP events that provide access to messages in the transport pipeline.

SMTP Receive events

Sequence	SMTP event	Description
1	**OnConnectEvent**	This event is triggered by the initial connection from a remote SMTP host.

2	**OnHeloCommand**	This event is triggered when the HELO command is issued by the remote SMTP host.
3	**OnEhloCommand**	This event is triggered when the EHLO command is issued by the remote SMTP host.
4	**OnStartTlsCommand**	This event is triggered when the STARTTLS command is issued by the remote SMTP host.
5	**OnAuthCommand**	This event is triggered when the AUTH command is issued by the remote SMTP host.
6	**OnProcessAuthentication**	This event is triggered when authentication with the remote SMTP host is being processed.
7	**OnEndOfAuthentication**	This event is triggered when the remote SMTP host has completed authentication.
8	**OnXSessionParamsCommand**	This event is triggered when the XSESSIONPARAMS command is issued by the remote SMTP host.
9	**OnMailCommand**	This event is triggered when the MAIL FROM command is issued by the remote SMTP host.
10	**OnRcptToCommand**	This event is triggered when the RCPT TO command is

		issued by the remote SMTP host.
11	**OnDataCommand**	This event is triggered when the DATA (text) or BDAT (binary data) command is issued by the remote SMTP host.
12	**OnEndOfHeaders**	This event is triggered when the remote SMTP host has completed submitting the email message headers. This is indicated by a blank line (<CRLF>) that separates the message headers and the message body.
13	**OnProxyInboundMessage**	This event is triggered when an inbound SMTP session is relayed or *proxied* by the Front End Transport service on a Client Access server to the Transport service on a Mailbox server.
14	**OnEndOfData**	This event is triggered when the remote SMTP host issues an end of data command. For text sessions started by the DATA command, the end of data indicator is <CRLF>.<CRLF>. For binary sessions started by the BDAT command, the end of data indicator is BDAT LAST.

**	**OnHelpCommand**	This event is triggered if the HELP command is issued by the remote SMTP host.
**	**OnNoopCommand**	This event is triggered if the NOOP command is issued by the remote SMTP host.
**	**OnReject**	This event is triggered if the receiving SMTP host issues a temporary or permanent delivery status notification (DSN) code to the sending SMTP host.
**	**OnRsetCommand**	This event is triggered if the RSET command is issued by the sending SMTP host.
15	**OnDisconnectEvent**	This event is triggered by the disconnection of the SMTP conversation by either the receiving or sending SMTP host. Typically, this happens when the QUIT command is issued by the remote SMTP host.

** These events can occur at any time after **OnConnectEvent** but before **OnDisconnectEvent**.

Categorizer events

Sequence	Categorizer event	Description
1	**OnSubmittedMessage**	This event is triggered when a message arrives in the Submission queue in the Transport service on the

		receiving Mailbox server or Edge Transport server.
2	**OnResolvedMessage**	This event is triggered after all the recipients have been resolved, but before the next hop has been determined for each recipient. The **OnResolvedMessage** routing event enables subsequent events to override the default routing behavior by using the per-recipient **SetRoutingOverride** method.
3	**OnRoutedMessage**	This event is triggered after messages have been categorized, distribution lists have been expanded, and recipients have been resolved.
4	**OnCategorizedMessage**	This event is triggered when the categorizer completes processing the message.

Return to top

Priority of transport agents

There are two factors that determine the order that transport agents act on messages in the transport pipeline:

1. The SMTP event where the transport agent is registered, and when that SMTP event encounters messages.
2. The priority value that's assigned to the transport agent if there are multiple agents registered to the same SMTP event. The highest priority is 1. A higher integer value indicates a lower agent priority.

For example, suppose you configured the following transport agents:

- Transport Agent A with a priority of 1 and Transport Agent C with a priority of 2 are registered to the **OnEndOfHeaders** SMTP event.
- Transport Agent B with a priority of 4 is registered to the **OnMailCommand** SMTP event.

Transport Agent B is applied to messages first because the **OnMailCommand** event encounters messages before the **OnEndOfHeaders** event. When messages reach the **OnEndOfHeaders** event, Transport Agent A is applied before Transport Agent C because Transport Agent A has a higher priority (lower integer value) than Transport Agent C.

Built-in transport agents

Exchange 2013 includes many built-in transport agents that provide features such as anti-spam, transport rules and journaling. Most of the built-in transport agents on Exchange 2013 Mailbox servers and Client Access servers are invisible and unmanageable by the transport agent management cmdlets. Virtually all of the built-in transport agents that are visible and manageable are in the Transport service on Mailbox servers and on Edge Transport servers.

The more interesting built-in transport agents on Mailbox servers are described in the following table. Note that this table doesn't include many of the invisible and unmanageable transport agents.

Interesting built-in transport agents on Mailbox servers

Agent name	Manageable?	Priority	SMTP or categorizer events
Transport Rule Agent	Yes	1	**OnResolvedMessage**
Malware Agent	Yes	2	**OnSubmittedMessage**
Text Messaging Routing Agent	Yes	3	**OnSubmittedMessage**
Text Messaging Delivery Agent	Yes	4	n/a
Journal Agent	No	Not configurable	**OnRoutedMessage**
Journal Report Decryption Agent	No	Not configurable	**OnCategorizedMessage**
RMS Decryption Agent	No	Not configurable	**OnSubmittedMessage**

RMS Encryption Agent	No	Not configurable	**OnSubmittedMessage**, **OnRoutedMessage**
RMS Protocol Decryption Agent	No	Not configurable	**OnEndOfData**

On Edge Transport servers, most of the built-in transport agents are visible and manageable by the transport agent management cmdlets or by other feature-specific cmdlets.

The more interesting built-in transport agents on Edge Transport servers are described in the following table. Note that this table doesn't include invisible or unmanageable transport agents.

Interesting built-in transport agents on Edge Transport servers

Agent name	Manageable?	Priority	SMTP or categorizer events
Connection Filtering Agent	Yes	1	**OnConnectEvent**, **OnMailCommand**, **OnRcptComand**, **OnEndOfHeaders**
Address Rewriting Inbound Agent	Yes	2	**OnRcptCommand**, **OnEndOfHeaders**
Edge Rule Agent	Yes	3	**OnEndOfData**
Content Filter Agent*	Yes	4	**OnEndOfData**
Sender ID Agent*	Yes	5	**OnEndOfHeaders**
Sender Filter Agent*	Yes	6	**OnMailCommand**, **OnEndOfHeaders**
Recipient Filter Agent	Yes	7	**OnRcptCommand**
Protocol Analysis Agent*	Yes	8	**OnConnectEvent**, **OnEndOfHeaders**, **OnEndOfData**, **OnReject**, **OnRsetCommand**, **OnDisconnectEvent**

Attachment Filtering Agent	Yes	9	**OnEndOfData**
Address Rewriting Outbound Agent	Yes	10	**OnSubmittedMessage**, **OnRoutedMessage**

* You can also install and configure these anti-spam agents on Mailbox servers. For more information, see Enable anti-spam functionality on Mailbox servers.

Return to top

Troubleshoot transport agents

To help you troubleshoot issues with transport agents, you can use the following features:

- **Get-TransportPipeline** This cmdlet shows the SMTP events and the corresponding transport agents that encounter messages on the Exchange server. For more information, see View transport agents in the transport pipeline.
- **Pipeline Tracing** Pipeline tracing creates an exact snapshot of a message before and after it encounters each transport agent. This allows you to find a transport agent that's causing unexpected results. For more information, see Pipeline tracing.

Return to top

Enable support for legacy transport agents

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-22*

In Microsoft Exchange Server 2013, transport agents that were created using the Microsoft .NET Framework version 4.0 are supported by default. Exchange 2013 supports transport agents that were created using previous versions of the .NET Framework, but support for these legacy transport agents isn't enabled by default. To enable support for legacy transport agents, you need to modify the appropriate XML application configuration file. The files you need to modify depend on where the transport agent is installed:

Server	Application configuration files	Microsoft Windows service
Client Access server	%ExchangeInstallPath%Bin	Microsoft Exchange Front End

	\MSExchangeFrontendTransport.exe.config	Transport (MSExchangeFrontendTransport)
Mailbox server	• %ExchangeInstallPath%Bin\EdgeTransport.exe.config • %ExchangeInstallPath%Bin\MSExchangeTransport.exe.config	Microsoft Exchange Transport (MSExchangeTransport)

Support for legacy transport agents is controlled by keys in the application configuration files. By default, none of the required keys are present in the application configuration files. You must add the keys manually. The following table explains each key in more detail.

Key	Description
useLegacyV2RuntimeActivationPolicy	This key enables or disables support for legacy transport agents. Valid values for this key are `true` or `false`. If this key isn't specified, the default value is `false`.
supportedRuntime version	This key specifies the version of the Microsoft .NET Framework that's required by the agent. Valid values for this key are: • `v4.0` or `v4.0.30319` • `v3.5` or `v3.5.21022` • `v3.0` or `v3.0.4506` • `v2.0` or `v2.0.50727` You specify multiple values using multiple separate instances of the *supportedRuntime version* key. When you enable legacy transport agent support using the *useLegacyV2RuntimeActivationPolicy* key, you should always specify the value `v4.0` in addition to the values required by the legacy transport agent.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- Exchange permissions don't apply to the procedures in this topic. These procedures are performed in the operating system of the Exchange Server.
- Changes you save to an application configuration file are applied after you restart the corresponding service.
- When you restart any of the services that are associated with the application configuration files, mail flow on the server is temporarily interrupted.
- Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Command Prompt to configure support for legacy transport agents

Use the following procedure to enable support for legacy transport agents:

1. In a Command prompt window, on the Exchange 2013 server where you want to configure the legacy transport agent support, open the appropriate application configuration file in Notepad by running the following command:

```
Notepad %ExchangeInstallPath%Bin\<AppConfigFile>
```

For example, to open the EdgeTransport.exe.config file on a Mailbox server, run the following command:

```
Notepad %ExchangeInstallPath%Bin\EdgeTransport.exe.config
```

2. Locate the *</configuration>* key at the end of the file, and paste the following keys before the *</configuration>* key:

```
<startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0" />
    <supportedRuntime version="v3.5" />
    <supportedRuntime version="v3.0" />
    <supportedRuntime version="v2.0" />
</startup>
```

3. When you are finished, save and close the application configuration file.
4. Repeat Steps 1 through 3 to modify the other application configuration files.
5. Restart the associated Windows service by running the following command:

```
net stop <service> && net start <service>
```

For example, if you modified the EdgeTransport.exe.config file, you need to restart the Microsoft Exchange Transport service by running the following command:

```
net stop MSExchangeTransport && net start
MSExchangeTransport
```

6. Repeat Step 5 to restart services associated with the other modified application configuration files.

How do you know this worked?

You'll know this procedure works if the legacy transport agent installs successfully. If you try to install a legacy transport agent without performing the procedures in this topic, you'll receive an error that's similar to the following:

```
Mixed mode assembly is built against version '<version>' of
the runtime and cannot be loaded in the 4.0 runtime without
additional configuration information.
```

Manage transport agents

Exchange Server 2013 > Mail flow > Transport agents >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Transport agents use SMTP events to operate on messages as the messages move through the transport pipeline. Most of the built-in transport agents that are included with Microsoft Exchange Server 2013 are invisible and unmanageable. However, you can install and configure third-party transport agents on Exchange servers in your organization. For more information about transport agents, see Transport agents.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport agents" entry in the Mail flow permissions

topic.

- You can only use the Shell to perform this procedure.
- Support for legacy transport agents isn't enabled by default, but you can enable it. For instructions, see Enable support for legacy transport agents.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

About transport agent procedures in the Front End Transport service on Client Access servers

You can't use the Exchange Management Shell to manage transport agent in the Front End Transport service on a Client Access server. Instead, you need to open Windows PowerShell on the Client Access server, and then import the Exchange cmdlets into the Windows PowerShell session.

Caution:

Running Exchange cmdlets in Windows PowerShell for tasks other than managing transport agents in the Front End Transport service is not supported. There are serious consequences that can result if you bypass the Exchange Management Shell and role-based access control (RBAC) by running Exchange cmdlets in Windows PowerShell. You should always run Exchange cmdlets in the Exchange Management Shell. For more information, see Release notes for Exchange 2013.

To perform any of the Transport Agent procedures described in this topic in the Front End Transport service, you need to perform the following additional steps:

1. On the Client Access server, open Windows PowerShell and run the following command:

```
Add-PSSnapin
Microsoft.Exchange.Management.PowerShell.SnapIn
```

2. Run the command as described, but add the following value to the command: `-TransportService FrontEnd`.

For example, to view the transport agents in the Front End Transport service on a Client Access server, run the following command:

```
Get-TransportAgent -TransportService FrontEnd
```

Use the Shell to install a transport agent

When you install a transport agent, Exchange only registers the DLLs associated with the transport agent. You need to make sure all files, registry keys, and other objects that the transport agent depends on are installed correctly and configured. After Exchange loads the DLLs, it continues to reference the DLLs after the command has completed.

Transport agents have full access to all e-mail messages that they encounter. Exchange puts no restrictions on a transport agent's behavior. Transport agents that are unstable or contain security flaws may affect the stability and security of Exchange. Therefore, you should only install transport agents that you fully trust and that have been fully tested in a test environment.

Transport agents are installed in a disabled state to make sure mail flow isn't affected by transport agents that haven't been configured. Therefore, after a transport agent has been configured correctly, you need to enable the transport agent.

Use the following syntax to install a transport agent.

```
Install-TransportAgent -Name <TransportAgentIdentity> -
TransportAgentFactory <"TransportAgentFactory"> -
AssemblyPath <"FilePath">
```

This example installs a fictitious transport agent named Contoso Transport Agent in the Transport service on a Mailbox server.

```
Install-TransportAgent -Name "Contoso Transport Agent" -
TransportAgentFactory
"vendor.exchange.ContosoTransportAgentfactory" -
AssemblyPath "C:\Program Files\Vendor\TransportAgent
\ContosoTransportAgentFactory.dll"
```

How do you know this worked?

To verify that you have successfully installed the transport agent, run the command `Get-TransportAgent` and verify the transport agent is listed.

Use the Shell to enable a transport agent

Use the following syntax to enable a transport agent.

```
Enable-TransportAgent <TransportAgentIdentity>
```

This example enables the transport agent named Contoso Transport Agent in the Transport service on a Mailbox server.

```
Enable-TransportAgent "Contoso Transport Agent"
```

How do you know this worked?

To verify that you have successfully enabled a transport agent, run the command `Get-TransportAgent | Format-List Name,Enabled` and verify the transport agent is enabled.

Use the Shell to disable a transport agent

Use the following syntax to disable a transport agent:

```
Disable-TransportAgent <TransportAgentIdentity>
```

This example disables the transport agent named Fabirkam Transport Agent in the Transport service on a Mailbox server.

```
Disable-TransportAgent "Fabrikam Transport Agent"
```

How do you know this worked?

To verify that you have successfully disabled a transport agent, run the command `Get-TransportAgent | Format-List Name,Enabled` and verify the transport agent is disabled.

Use the Shell to view transport agents

To view a summary list of transport agents, run the following command:

```
Get-TransportAgent
```

To view the detailed configuration of a specific transport agent, run the following command:

```
Get-TransportAgent <TransportAgentIdentity> | Format-List
```

This example provides detailed configuration of the transport agent named Transport Rule Agent.

```
Get-TransportAgent "Transport Rule Agent" | Format-List
```

Use the Shell to configure the priority of a transport agent

Transport agents with a priority closest to 0 process email messages first. However, the SMTP event in the transport pipeline where the transport agent is registered may cause a lower priority agent to act on the message before a higher priority agent.

To modify the priority of an existing transport agent, run the following command:

```
Set-TransportAgent <TransportAgentIdentity> -Priority
<Integer>
```

This example sets the priority agent value of 3 for the existing transport agent named Contoso Transport Agent in the Transport service on a Mailbox server.

```
Set-TransportAgent "Contoso Transport Agent" -Priority 3
```

How do you know this worked?

To verify that you have successfully configured the priority of a transport agent, run the command `Get-TransportAgent | Format-List Name,Priority` and verify the priority value of the transport agent.

Use the Shell to uninstall a transport agent

When the transport agent is uninstalled, Exchange unregisters the DLL files used with the agent. Exchange doesn't remove any files, registry keys, or other objects added by the installation of the transport agent.

To uninstall a transport agent, run the following command:

```
Uninstall-TransportAgent <TransportAgentIdentity>
```

This example uninstalls the transport agent named Fabrikam Transport Agent from the Transport service on a Mailbox server.

```
Uninstall-TransportAgent "Fabrikam Transport Agent"
```

How do you know this worked?

To verify that you have successfully uninstalled the transport agent, run the command `Get-TransportAgent` and verify the transport agent isn't listed.

View transport agents in the transport pipeline

Exchange Server 2013 > Mail flow > Transport agents >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-21*

You can use the Exchange Management Shell to view a list of transport agents in the transport pipeline on Microsoft Exchange Server 2013 Mailbox servers and Client Access servers. Specifically, the **Get-TransportPipeline** cmdlet shows information about the following types of transport

agents in the transport pipeline:

- Agents based on the **SmtpReceiveAgent**, **RoutingAgent**, **DeliveryAgent**, and **StorageAgent** classes in the Transport service on Mailbox servers.
- Agents based on the **SmtpReceiveAgentClass** in the Mailbox Transport Delivery service on Mailbox servers.
- Agents based on the **SmtpReceiveAgentClass** in the Front End Transport service on Client Access servers.

You can view a list of all the enabled transport agents that have encountered messages in the transport pipeline and the SMTP events they are registered on. For more information about transport agents, see Transport agents.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport agents" entry in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to view a list of transport agents in the transport pipeline

To use the Shell to view a list of transport agents in the transport pipeline on an Exchange server, run the following command:

```
Get-TransportPipeline | Format-List
```

To export the results to a text file named C:\My Documents\Transport Agents.txt, run the following command:

```
Get-TransportPipeline | Format-List > "C:\My Documents
\Transport Agents.txt"
```

How do you know this worked?

Only transport agents that have encountered messages in the transport pipeline between the time when the transport service was started and the time when the **Get-TransportPipeline** cmdlet was run are displayed by the cmdlet. A transport agent that hasn't encountered a message in the transport pipeline won't appear in the results displayed by the **Get-TransportPipeline** cmdlet, even if that transport agent is enabled.

Transport high availability

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-15*

In Microsoft Exchange Server 2013, transport high availability is responsible for keeping redundant copies of messages before and after the messages are successfully delivered. Exchange 2013 improves upon the transport high availability features introduced in Exchange Server 2010, for example, shadow redundancy and the transport dumpster, to help ensure messages aren't lost in transit.

Here's a summary of the major transport high availability improvements in Exchange 2013:

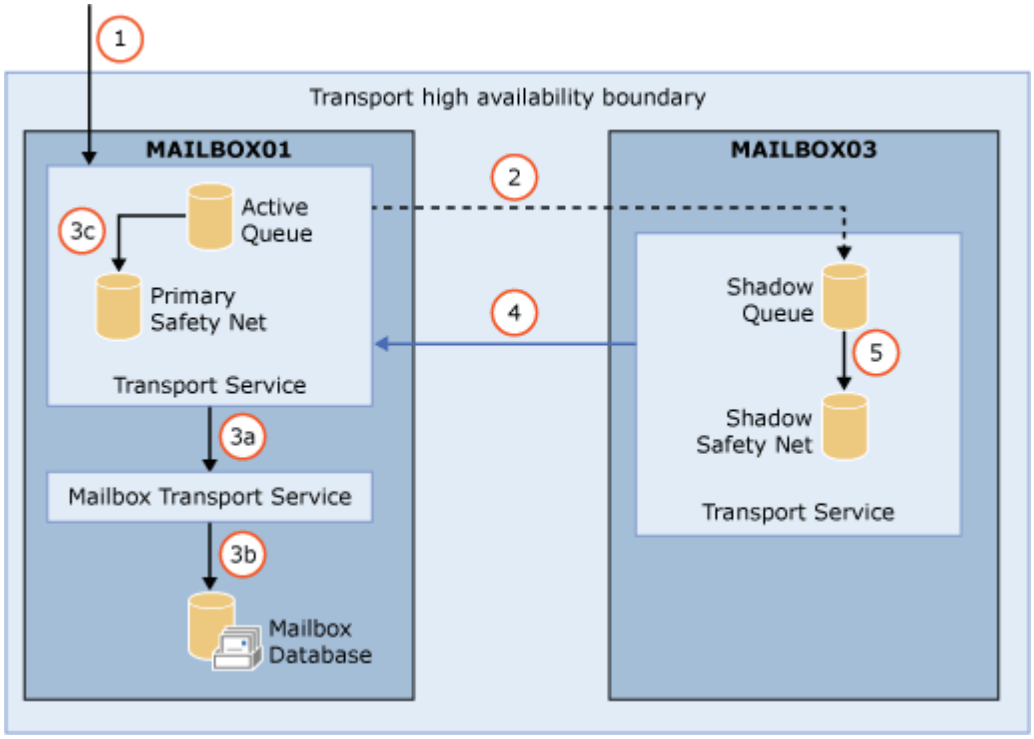
- Shadow redundancy creates a redundant copy of the message on another server before the message is accepted or acknowledged. The sending server's support or lack of support for shadow redundancy is irrelevant.
- Shadow redundancy recognizes both database availability groups (DAGs) and Active Directory sites as transport high availability boundaries. This reduces the number of servers that can hold redundant copies of messages, and eliminates unnecessary redundant message maintenance traffic across DAGs or Active Directory sites.

For more information, see Shadow redundancy.

- The transport dumpster has been improved and is now named *Safety Net*. Safety Net stores messages that were successfully processed by the Transport service on Mailbox servers. Safety Net works best for Mailbox servers in a DAG, but Safety Net also works for multiple Mailbox servers in the same Active Directory site that don't belong to a DAG.
- Safety Net itself is now made redundant on another server. This is important to avoid a single point of failure in Exchange 2013, because the Transport service and the mailbox databases are both located on the Mailbox server.

For more information, see Safety Net.

The following diagram provides a high-level overview of how transport high availability works in Exchange 2013.



1. An Exchange 2013 Mailbox server named Mailbox01 receives a message from an SMTP server that's outside the transport high availability boundary. The *transport high availability boundary* is a DAG or an Active Directory site in non-DAG environments. The message could come from a third-party SMTP server, from an Internet SMTP server proxied through a Client Access server, or from another Exchange 2013 server.
2. Before acknowledging receipt of the message, Mailbox01 initiates a new SMTP session to another Exchange 2013 Mailbox server named Mailbox03 that's within the Transport high availability boundary, and Mailbox03 makes a shadow copy of the message. In DAG environments, a shadow server in a remote Active Directory site is preferred. Mailbox01 is the primary server holding the primary message, and Mailbox03 is the shadow server holding the shadow message.
3. The Transport service on Mailbox01 processes the primary message.
   a. In this example, the recipient's mailbox is located on Mailbox01, so the Transport service transmits the message to the local Mailbox Transport service.
   b. The Mailbox Transport service delivers the message to the local mailbox database.
   c. Mailbox01 queues a discard status for Mailbox03 that indicates the primary message was successfully processed, and Mailbox01 moves a copy of the primary message into the local Primary Safety Net. Note that the message moves between queues within the same queue database.
4. Mailbox03 periodically polls Mailbox01 for the discard status of the primary message.
5. When Mailbox03 determines Mailbox01 successfully processed the primary message, Mailbox03 moves the shadow message into the local Shadow Safety Net. Note that the message moves between queues within the same queue database.

The message is retained in Primary Safety Net and Shadow Safety Net until the message expires based on a configurable timeout value. If a mailbox database failover occurs before the message expires, the Primary Safety Net on Mailbox01 resubmits the message. If the Mailbox01 isn't available, the Shadow Safety Net on Mailbox03 takes over and resubmits the message.

Message redundancy in the Front End Transport service on Client Access servers

A Client Access server has no message queues. It's a stateless proxy server that uses the Front End Transport service to accept incoming SMTP connections and proxy them to the Transport service on a Mailbox server. The Front End Transport service keeps the SMTP session with the sending server open while the primary message is transmitted to the Transport service on a Mailbox server, and a shadow copy of the message is made by the Transport service on a different Mailbox server within the transport high availability boundary. Only after both the primary message and shadow message are successfully created, the end of data SMTP command is sent back to the sending SMTP server through the Client Access server.

Shadow redundancy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-18*

Shadow redundancy was introduced in Microsoft Exchange Server 2010 to provide redundant copies of messages before they're delivered to mailboxes. In Exchange 2010, shadow redundancy delayed deleting a message from the transport database on a transport server until the server verified the next hop in the message delivery path completed delivery. If the next hop failed before reporting successful delivery back to the transport server, the transport server resubmitted the message to that next hop. Exchange 2010 servers used the XSHADOW verb to advertise their shadow redundancy support. If an SMTP server didn't support shadow redundancy, Exchange 2010 used delayed acknowledgement based on a configured time interval on the Receive connector to make a redundant copy of the message.

The major improvement to shadow redundancy in Microsoft Exchange Server 2013 is that the transport server now makes a redundant copy of any messages it receives before it acknowledges successfully receiving the message back to the sending server. The sending server's support or lack of support for shadow redundancy doesn't matter. This helps to ensure that all messages in the Exchange 2013 transport pipeline are made redundant while they're in transit. If Exchange 2013 determines the original message was lost in transit, the redundant copy of the message is redelivered.

Contents

Shadow redundancy components

The following table describes the components of shadow redundancy. These terms are used throughout the topic.

Term	Description
Transport server	An Exchange server that has message queues and is responsible for routing messages. In Exchange 2013, a transport server is a Mailbox server (the Transport service on the Mailbox server).
Transport database	The message queue database on an Exchange 2013 transport server. Shadow queues and Safety Net are also stored in the transport database.
Transport high availability boundary	A database availability group (DAG) in DAG environments, or an Active Directory site in non-DAG environments. When a message arrives on a transport server in the transport high availability boundary, Exchange tries to maintain 2 redundant copies of the message on transport servers within the boundary. When a message leaves the transport high availability boundary, Exchange stops maintaining redundant copies of the message.
Primary message	The message submitted into the transport

	pipeline for delivery.
Shadow message	The redundant copy of the message that the shadow server retains until it confirms the primary message was successfully processed by the primary server.
Primary server	The transport server that's currently processing the primary message.
Shadow server	The transport server that holds the shadow message for the primary server. A transport server may be the primary server for some messages and the shadow server for other messages simultaneously.
Shadow queue	The delivery queue where the shadow server stores shadow messages. For messages with multiple recipients, each next hop for the primary message requires separate shadow queues.
Discard status	The information a transport server maintains for shadow messages that indicate the primary message has been successfully processed.
Discard notification	The response a shadow server receives from a primary server indicating a shadow message is ready to be discarded.
Safety Net	The Exchange 2013 improved version of the transport dumpster. Messages that are successfully processed or delivered to a mailbox recipient by the Transport service on a Mailbox server are moved into Safety Net. For more information, see Safety Net.

Shadow Redundancy Manager	The transport component that manages shadow redundancy.
Heartbeat	The process that allows primary servers and shadow servers to verify the availability of each other.

Return to top

Requirements for shadow redundancy

Although it may seem obvious, shadow redundancy requires multiple Exchange 2013 Mailbox servers. The Mailbox server can be standalone servers, or Mailbox servers and Client Access servers installed on the same computer.

- If the Mailbox server isn't a member of a DAG, the other Mailbox servers must be in the local Active Directory site.
- If the Mailbox server is a member of a DAG, the other Mailbox servers must belong to the same DAG. The other Mailbox servers that belong to the DAG can be in the local Active Directory site or in a remote Active Directory site. If the DAG spans multiple Active Directory sites, shadow redundancy prefers creating a redundant copy of the message in a remote Active Directory site for site resiliency.

These are the situations where shadow redundancy can't protect messages in transit:

- In single Exchange server environments.
- In under-provisioned DAGs.
- During the simultaneous failure of two or more transport servers involved in the shadow redundancy of a message.

Return to top

Shadow redundancy is enabled by default

By default, shadow redundancy is enabled globally in the Transport service on all Mailbox servers by using the *ShadowRedundancyEnabled* parameter on the **Set-TransportConfig** cmdlet. By default, if the Transport service on a Mailbox server can't create a redundant copy of a message, the message is not rejected. However, you can configure Exchange 2013 to reject a message if a redundant copy of the message isn't created by using the *RejectMessageOnShadowFailure* parameter on the **Set-TransportConfig** cmdlet. The message is rejected with a transient failure, but the sending server can transmit the message again. The SMTP response code is `451 4.4.0 Message failed to be made redundant.` You should configure Exchange 2013 to reject messages that can't be made redundant only when your organization has multiple Exchange 2013 Mailbox servers available.

The following table describes the parameters that enable shadow redundancy.

Parameters that enable shadow redundancy

Parameter	Default value	Description
ShadowRedundancyEnabled on **Set-TransportConfig**	`$true`	• `$true` enables shadow redundancy on all transport servers in the organization. • `$false` disables shadow redundancy on all transport servers in the organization.
RejectMessageOnShadowFailure on **Set-TransportConfig**	`$false`	• `$false` When a shadow copy of the message can't be created, the primary message is accepted anyway by transport servers in the organization. Those messages aren't redundantly persisted while they're in transit. • `$true` No message is accepted or acknowledged by any transport server in the organization until a shadow copy of the message is successfully created. If a shadow copy of the message can't be created, the primary message is rejected with a transient error. All messages in the organization are redundantly persisted while they're in transit. You should set this value to `$true` only if you have multiple Exchange 2013 Mailbox servers in a DAG or Active Directory site where a shadow copy of the message can be created. This parameter is only meaningful when *ShadowRedundancyEnabled* is `$true`.

Return to top

How shadow messages are created

The main goal of shadow redundancy is to always have two copies of a message within a transport high availability boundary while the message is in transit. Where and when the redundant copy of the message is created depends on where the message came from and where the message is going. There are three major determining factors:

- Messages received from outside a transport high availability boundary.
- Messages sent outside a transport high availability boundary.
- Messages received from the Mailbox Transport Submission service from a Mailbox server within the transport high availability boundary.

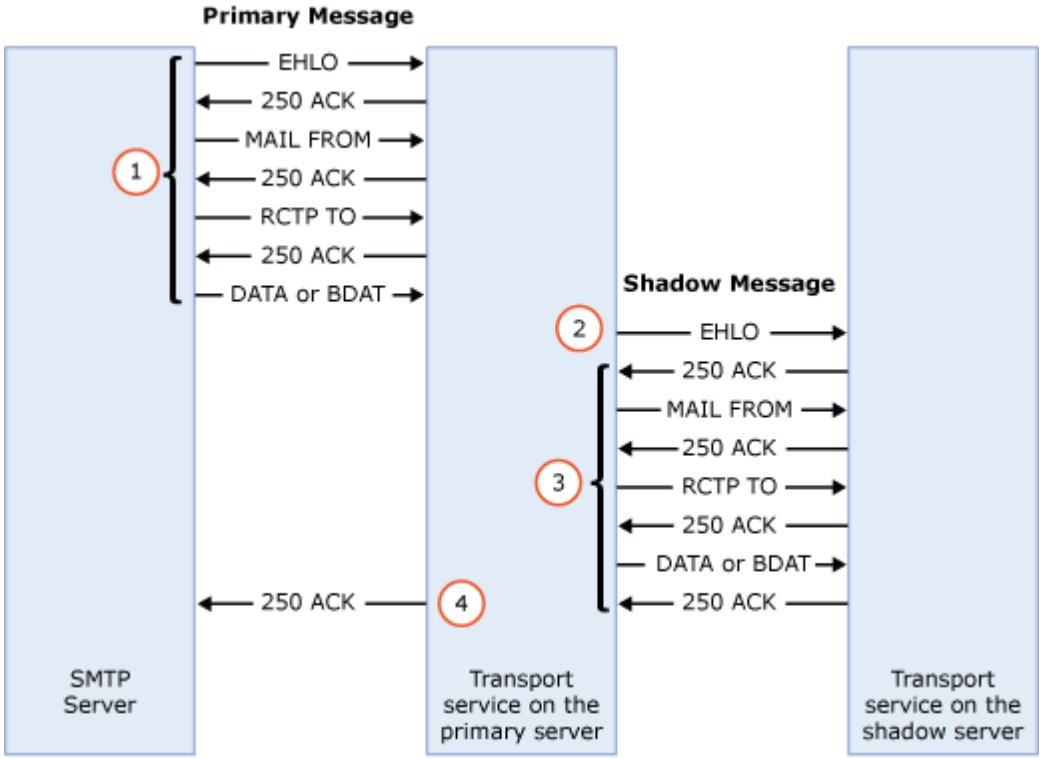
A *transport high availability boundary* is one of the following:

- A DAG, for Mailbox servers that are members of a DAG. This includes a DAG that spans multiple Active Directory sites.
- An Active Directory site, for Mailbox servers that don't belong to a DAG.

Shadow redundancy never tracks shadow messages across a transport high availability boundary. When a message crosses the transport high availability boundary, shadow redundancy begins or restarts. This reduces shadow message maintenance traffic and prevents shadow message resubmissions from occurring across the transport high availability boundary. Exchange 2010 Hub Transport servers are a special case, and are discussed later in this topic.

Messages received from outside a transport high availability boundary

When the Transport service on an Exchange 2013 Mailbox server receives a message from outside the transport high availability boundary, the Mailbox server isn't concerned about the support or lack of support for shadow redundancy by the sending server. As long as shadow redundancy is enabled, the Mailbox server that receives the message makes a redundant copy of the message on another Mailbox server within the transport high availability boundary before acknowledging receipt of the message back to the sending server. Here's an example of how the process works:



1. An SMTP server transmits a message to the Transport service on a Mailbox server. The Mailbox server is the primary server, and the message is the primary message.
2. While the original SMTP session with the SMTP server is still active, the Transport service on primary server opens a new, simultaneous SMTP session with the Transport service on a different Mailbox server in the organization to create a redundant copy of the message.
   - If the primary server is a member of a DAG, the primary server connects to a different Mailbox server in the same DAG. If the DAG spans multiple Active Directory sites, a Mailbox server in a different Active Directory site is preferred by default. This setting is controlled by the *ShadowMessagePreference* parameter on the **Set-TransportService** cmdlet. The default value is `PreferRemote`, but you can change it to `RemoteOnly` or `LocalOnly`.
   - If the primary server isn't a member of a DAG, the primary server connects to a different Mailbox server in the same Active Directory Site, regardless of the value of the *ShadowMessagePreference* parameter.
3. The primary server transmits a copy of the message to the Transport service on other Mailbox server, and Transport service on the other Mailbox server acknowledges that the copy of the message was created successfully. The copy of the message is the shadow message, and the Mailbox server that holds it is the shadow server for the primary server. The message exists in a shadow queue on the shadow server.
4. After the primary server receives acknowledgement from the shadow server, the primary server acknowledges the receipt of the primary message to the original SMTP server in the original SMTP session, and the SMTP session is closed.

Messages sent outside a transport high availability boundary

When an Exchange 2013 transport server transmits a message outside the transport high availability boundary, and the SMTP server on the other side acknowledges successful receipt of the message, the transport server moves the message into Safety Net. No resubmission of the message from Safety Net can occur after the primary message has been successfully transmitted across the transport high availability boundary. For more information about Safety Net, see Safety Net.

Messages transmitted within a transport high availability boundary

Message routing is optimized in Exchange 2013 so that when the ultimate destination is in a DAG or Active Directory site, multiple hops between the Transport service on Mailbox servers in that DAG or Active Directory site aren't typically required. Once the message is accepted by the Transport service on a Mailbox server in the DAG or Active Directory site that holds the ultimate destination for the message, the next hop for the message is typically the ultimate destination itself. Shadow redundancy's goal of keeping two copies of a message in transit is fulfilled when one shadow copy of the message exists anywhere within the DAG or Active Directory site. Typically, only failover scenarios in a DAG that require the **Redirect-Message** cmdlet to drain the active queues on a Mailbox server would require multiple hops within the same transport high availability boundary.

Shadow redundancy with Exchange 2010 Hub Transport servers in the same Active Directory site

When an Exchange 2010 Hub Transport server transmits a message to an Exchange 2013 Mailbox server in the same Active Directory site, the Exchange 2010 Hub Transport server advertises support for shadow redundancy using the XSHADOW command, but the Mailbox server doesn't advertise support for shadow redundancy. This prevents the Exchange 2010 Hub Transport server from creating a shadow copy of the message on an Exchange 2013 Mailbox server.

When the Transport service on an Exchange 2013 Mailbox server transmits a message to an Exchange 2010 Hub Transport in the same Active Directory site, the Exchange 2013 Mailbox server shadows the message for the Exchange 2010 Hub Transport server. After the Exchange 2013 Mailbox server receives acknowledgement from the Exchange 2010 Hub Transport server that the message was successfully received, the Exchange 2013 Mailbox server moves the successfully processed message into Safety Net. However, the successfully processed messages stored in Safety Net by Exchange 2013 Mailbox are never resubmitted to the Exchange 2010 Hub Transport servers.

Return to top

SMTP timeouts

During the attempt to make a redundant copy of the message, the SMTP connection between the sending SMTP server and the primary server, or the SMTP session between the primary server and the shadow server could timeout. Receive connectors and Send connectors both have a *ConnectionInactivityTimeOut* parameter for when data is actually being transmitted on the connector. Receive connectors also have an absolute *ConnectionTimeOut* parameter.

If any of the SMTP sessions time out before the shadow copy of the message is successfully created and acknowledged, the result is controlled by the *RejectMessageOnShadowFailure* parameter on the **Set-TransportConfig** cmdlet. By default, the value of this parameter is `$false`, which means the primary message is accepted without a shadow copy being created. If the value of this parameter is `$true` the primary message is rejected with the transient error `451 4.4.0`.

If the shadow copy of a message is successfully created, but the SMTP session between the sending SMTP server and the primary server times out, the primary server accepts and processes the primary message. The sending SMTP server will re-deliver the unacknowledged message, but duplicate message detection will prevent Exchange mailbox users from seeing the duplicate messages. When the sending SMTP server resubmits the message, the primary server will create another shadow copy of the message. There's no relationship between the shadow messages created during message resubmissions by the sending SMTP server.

The following table describes the parameters that control the creation of shadow messages

Shadow message creation parameters

Source	Default value	Description
ShadowMessagePreferenceSetting on **Set-TransportConfig**	`PreferRemote`	• `PreferRemote` Try to make a shadow copy of the message on a Mailbox server in a different Active Directory site. If the operation fails, try make a shadow copy of the message on a server in the local Active Directory site. • `LocalOnly` A shadow copy of the message should only be made on a transport server in the local Active Directory site. • `RemoteOnly`: A shadow copy of the message should only be made on a transport server in a different Active Directory site. This parameter is only meaningful when the primary server that's trying to make a shadow copy of the message is a Mailbox server that's a

		member of a DAG that spans multiple Active Directory sites.
MaxRetriesForRemoteSiteShadow on **Set-TransportConfig**	4	This parameter is used when the Mailbox server is a member of a DAG that spans multiple Active Directory sites. • If *ShadowMessagePreferenceSetting* is set to `PreferRemote`, first the Mailbox server tries to create a shadow copy of the message on another Mailbox server in a remote Active directory site up to the number of times specified by *MaxRetriesForRemoteSiteShadow*. If this fails, the Mailbox server tries to create a shadow copy of the message on a different Mailbox server in the local Active Directory site up to the number of times specified by *MaxRetriesForLocalSiteShadow*. • If *ShadowMessagePreferenceSetting* is set to `RemoteOnly`, the Mailbox server only tries to create a shadow copy of the message on a Mailbox server in a remote Active Directory site up to the number of times specified by *MaxRetriesForRemoteSiteShadow*. • The

		When a shadow copy of the message can't be successfully created: • If *RejectMessageOnShadowFailure* is `$true`, the primary message is rejected with a transient error. • If *RejectMessageOnShadowFailure* is `$false`, the primary message is accepted anyway, but isn't redundantly persisted.
MaxRetriesForLocalSiteShadow on **Set-TransportConfig**	2	This parameter is used in the following circumstances: • If the Mailbox server is a member of a DAG that spans multiple Active Directory sites. 1. If *ShadowMessagePreferenceSetting* is set to `PreferRemote`, first the Mailbox server tries to create a shadow copy of the message on another Mailbox server in a remote Active directory site up to the number of times specified by *MaxRetriesForRemoteSiteShadow*. If this fails, the Mailbox server tries to create a shadow copy of the message on a different Mailbox server in the local Active Directory site up to the number of times specified by *MaxRetriesForLocalSiteShadow*. 2. If *ShadowMessagePreferenceSetting* is set to `LocalOnly`, the

		Mailbox server only tries to create a shadow copy of the message on a different Mailbox server in the local Active Directory site up to the number of times specified by the *MaxRetriesForLocalSiteShadow*. • If the Mailbox server isn't a member of a DAG, or if the Mailbox server is a member of a DAG that's in one Active Directory site, the Mailbox server only tries to create a shadow copy of the message on a different Mailbox server in the local Active Directory site up to the number of times specified by *MaxRetriesForLocalSiteShadow*. When a shadow copy of the message can't be successfully created: • If *RejectMessageOnShadowFailure* is `$true`, the primary message is rejected with a transient error. • If *RejectMessageOnShadowFailure* is `$false`, the primary message is accepted anyway, but isn't redundantly persisted.
ConnectionInactivityTimeout on **Set-ReceiveConnector**	5 minutes in the Transport service on Mailbox servers 5 minutes in the Front End Transport service on Client Access servers.	This parameter specifies the maximum time that an open SMTP connection with a source messaging server can remain idle before the connection is closed. The value of this parameter must be smaller than

	1 minute on Edge Transport servers.	the value specified by the *ConnectionTimeout* parameter.
ConnectionTimeout on **Set-ReceiveConnector**	10 minutes in the Transport service on Mailbox servers 10 minutes in the Front End Transport service on Client Access servers. 5 minutes on Edge Transport servers.	This parameter specifies the maximum time that an SMTP connection with a source messaging server can remain open, even if the source messaging server is transmitting data. The value of this parameter must be larger than the value specified by the *ConnectionInactivityTimeout* parameter.
ConnectionInactivityTimeOut on **Set-SendConnector**	10 minutes	This parameter specifies the maximum time that an open SMTP connection with a destination messaging server can remain idle before the connection is closed.

Return to top

How shadow messages are maintained

After a shadow message is successfully created, the work of shadow redundancy has only just begun. The primary server and the shadow server need to stay in contact with each other to track the progress of the message.

When the primary server successfully transmits the message to the next hop, and the next hop acknowledges receipt of the message, the primary server updates the *discard status* of the message as delivery complete. The discard status is basically a message that contains of list of messages that are being monitored. A successfully delivered message doesn't need to be kept in a shadow queue, so once the shadow server knows the primary server has successfully transmitted the message to the next hop, the shadow server moves the shadow message from the shadow queue into Safety Net.

The shadow server determines the discard status of the shadow messages in its shadow queues by querying the primary server. If the shadow server opens an SMTP session with the primary server for any reason, including the transmission of other unrelated messages, the shadow server issues the **XQDISCARD** command to determine the discard status of the primary messages. If the shadow

server hasn't opened an SMTP session with the primary server after a preconfigured time interval, the shadow server will open an SMTP session with the primary server and issue the **XQDISCARD** command. The time interval is controlled by the *ShadowHeartbeatFrequentcy* parameter on the **Set-TransportConfig** cmdlet. The default value is 2 minutes. After the shadow server opens an SMTP session with the primary server, the primary server responds with the *discard notifications* for messages that apply to the querying shadow server. In Exchange 2013, discard notifications are stored on disk, not in memory. Therefore, if the Microsoft Exchange Transport service restarts, the discard notifications are persisted. After the service starts, the primary server still knows about the messages it successfully processed, and that information is available to the shadow server.

The SMTP communication between the shadow server and the primary server is used as the *heartbeat* that determines the availability of the servers. If the shadow server can't open an SMTP session with the primary server after a preconfigured time interval, or if the transport database of the primary server has a different database ID, the shadow server promotes itself as the primary server, promotes the shadow messages as primary messages, and transmits the messages to the next hop. The time interval is controlled by the *ShadowResubmitTimeSpan* parameter on the **Set-TransportConfig** cmdlet. The default value is 3 hours.

Shadow Redundancy Manager is the core component of an Exchange 2013 transport server that's responsible for managing shadow redundancy. Shadow Redundancy Manager is responsible for maintaining the following information for all the primary messages that a server is currently processing:

- The shadow server for each primary message being processed.
- The discard status to be sent to shadow servers.

Shadow Redundancy Manager is responsible for the following for all the shadow messages that a shadow server has in its shadow queues:

- Maintaining the list of primary servers for each shadow message.
- Comparing the original database ID and the current database ID of the queue database where the primary copy of the message is stored.
- Checking the availability of each primary server for which a shadow message is queued.
- Processing discard notifications from primary servers.
- Removing the shadow messages from the shadow queues after all expected discard notifications are received.
- Deciding when the shadow server should take ownership of shadow messages, becoming a primary server.
- Tracking message bifurcations and other side-effect messages like delivery status notifications (DSNs) and journal reports to verify the redundant copy of the message isn't released until all forks of the message are fully processed.

The following table describes the parameters that control how shadow messages are maintained.

Parameter	Default value	Description
ShadowHeartbeatFrequency	2 minutes	The maximum amount of time

on **Set-TransportConfig**		a shadow server waits before opening an SMTP connection to the primary server to check the discard status of messages.
ShadowResubmitTimeSpan on **Set-TransportConfig**	3 hours	How long a server waits before deciding that a primary server has failed and assumes ownership of shadow messages in the shadow queue for the primary server that's unreachable.
ShadowMessageAutoDiscardInterval on **Set-TransportConfig**	2 days	How long a server retains discard events for successfully delivered messages. A primary server queues discard events until queried by the shadow server. However, if the shadow server doesn't query the primary server for the duration specified in this parameter, the primary server deletes the queued discard events.
SafetyNetHoldTime on **Set-TransportConfig**	2 days	How long successfully processed messages are retained in Safety Net. Unacknowledged shadow messages eventually expire from Safety Net after the sum of *SafetyNetHoldTime* and *MessageExpirationTimeout* on **Set-TransportService**.

MessageExpirationTimeout on **Set-TransportService**	2 days	How long a message can remain in a queue before it expires.

Return to top

Message processing after an outage

Shadow redundancy minimizes message loss due to server outages. When a transport server comes back online after an outage, there are two scenarios:

- **The server comes back online with a new transport database** In this scenario, the transport database is unrecoverable due to data corruption or hardware failure. In this case, because the transport server will have a new database ID, it will be recognized as a new route by the other transport servers in the organization. This also applies to the situation where a server couldn't be recovered, and a new server was provisioned as a replacement.
- **The server comes back online with the same transport database** In this scenario, the particular transport server didn't fail, but was offline long enough for the shadow server to assume ownership of the messages and resubmit them. For example, a network card failure, or a long maintenance on the server would cause this scenario.

The following table summarizes how shadow redundancy reacts to these two scenarios. For clarity, assume that the server that had an outage is named Mailbox01.

Message processing in recovery scenarios

Recovery scenario	Actions taken
Mailbox01 comes back online with a new database.	When Mailbox01 becomes unavailable, each server that has shadow messages queued for Mailbox01 will assume ownership of those messages and resubmit them. The messages then get delivered to their destinations. The maximum delay for messages is the value of the *ShadowHeartbeatFrequency* parameter on the **Set-TransportConfig** cmdlet. The default value is 2 minutes.
Mailbox01 comes back online with the same database.	After Mailbox01 comes back online, it will deliver the messages in its queues, which have already been delivered by the servers that hold

	shadow copies of messages for Mailbox01. This will result in duplicate delivery of these messages. Exchange mailbox users won't see duplicate messages due to duplicate message detection. However, recipients on non-Exchange messaging systems may receive duplicate copies of messages. The maximum delay for messages is the value of the *ShadowResubmitTimeSpan* parameter on the **Set-TransportConfig** cmdlet. The default value is 3 hours.

Return to top

Safety Net

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-26*

In Microsoft Exchange Server 2013, the primary mechanism of mailbox high availability is the database availability group (DAG). For more information about DAGs, see Managing database availability groups. The *transport dumpster* was first introduced in Exchange 2007, and was further improved in Exchange 2010 to provide redundant copies of messages after they're successfully delivered to mailboxes in DAGs. In Exchange 2010, the transport dumpster helped protect against data loss by maintaining a queue of successfully delivered messages that hadn't replicated to the passive mailbox database copies in the DAG. When a mailbox database or server failure required the promotion of an out-of-date copy of the mailbox database, the messages in the transport dumpster were automatically resubmitted to the new active copy of the mailbox database.

The transport dumpster has been improved in Exchange 2013 and is now called *Safety Net*.

Here's how Safety Net is similar to the transport dumpster in Exchange 2010:

- Safety Net is a queue that's associated with the Transport service on a Mailbox server. This queue stores copies of messages that were successfully processed by the server.
- You can specify how long Safety Net stores copies of the successfully processed messages before they expire and are automatically deleted. The default is 2 days.

Here's how Safety Net is different in Exchange 2013:

- Safety Net doesn't require DAGs. For Mailbox servers that don't belong to a DAGs, Safety Net stores copies of the delivered messages on other Mailbox servers in the local Active Directory site.
- Safety Net itself is now redundant, and is no longer a single point of failure. This introduces the concept of the *Primary Safety Net* and the *Shadow Safety Net*. If the Primary Safety Net is unavailable for more than 12 hours, resubmit requests become shadow resubmit requests, and messages are re-delivered from the Shadow Safety Net.
- Safety Net takes over some responsibility from shadow redundancy in DAG environments. Shadow redundancy doesn't need to keep another copy of the delivered message in a shadow queue while it waits for the delivered message to replicate to the passive copies of mailbox database on the other Mailbox servers in the DAG. The copy of the delivered message is already stored in Safety Net, so the message can be resubmitted from Safety Net if necessary.
- In Exchange 2013, transport high availability is more than just a best effort for message redundancy. Exchange 2013 attempts to guarantee message redundancy. Because of this, you can't specify a maximum size limit for Safety Net. You can only specify how long Safety Net stores messages before they're automatically deleted.

Contents

How Safety Net works

Shadow redundancy keeps a redundant copy of the message while the message is in transit. Safety Net keeps a redundant copy of a message after the message is successfully processed. So, Safety Net begins where shadow redundancy ends. The same concepts about shadow redundancy, including the transport high availability boundary, primary messages, primary servers, shadow messages and shadow servers also apply to Safety Net. For more information, see Shadow redundancy.

The Primary Safety Net exists on the Mailbox server that held the primary message before the message was successfully processed by the Transport service. This could mean the message was delivered to the Mailbox Transport service on the destination Mailbox server. Or, the message could have been relayed through the Mailbox server in an Active Directory site that's designated as a hub site on the way to the destination DAG or Active Directory site. After the primary server processes the primary message, the message is moved from the active queue into the Primary Safety Net on the same server.

The Shadow Safety Net exists on the Mailbox server that held the shadow message. After the shadow server determines the primary server has successfully processed the primary message, the shadow server moves the shadow message from the shadow queue into the Shadow Safety Net on

the same server. Although it may seem obvious, the existence of the Shadow Safety Net requires shadow redundancy to be enabled, and shadow redundancy is enabled by default in Exchange 2013.

The parameters used by Safety Net are described in the following table.

Parameter	Default value	Description
SafetyNetHoldTime on **Set-TransportConfig**	2 days	The length of time successfully processed primary messages are stored in Primary Safety Net, and acknowledged shadow messages are stored in Shadow Safety Net. You can also specify this value in the Exchange Administration Center (EAC) at **Mail flow** > **Receive connectors** > **More options ···** > **Organization transport settings** > **Safety Net** > **Safety Net hold time**. Unacknowledged shadow messages eventually expire from Shadow Safety Net after the sum of *SafetyNetHoldTime* and *MessageExpirationTimeout* on **Set-TransportService**. To avoid data loss during Safety Net resubmits, the value of *SafetyNetHoldTime* must be greater than or equal to the value of *ReplayLagTime* on **Set-MailboxDatabaseCopy** for the lagged copy of the mailbox database.
ReplayLagTime on **Set-**	Not configured	The amount of time that the

MailboxDatabaseCopy		Microsoft Exchange Replication service should wait before replaying log files that have been copied to the passive database copy. Setting this parameter to a value greater than 0 creates a lagged copy of the mailbox database. The maximum value is 14 days. To avoid data loss during Safety Net resubmits, the value of *ReplayLagTime* must be less than or equal to the value of *SafetyNetHoldTime* on **Set-TransportConfig** for the lagged copy of the mailbox database.
MessageExpirationTimeout on **Set-TransportService**	2 days	How long a message can remain in a queue before it expires.
ShadowRedundancyEnabled on **Set-TransportConfig**	`$true`	• `$true` enables shadow redundancy on all transport servers in the organization. • `$false` disables shadow redundancy on all transport servers in the organization. A redundant Safety Net requires shadow redundancy to be enabled.

Return to top

Message resubmission from Safety Net

Message resubmissions from Safety Net are initiated by the Active Manager component of the Microsoft Exchange Replication service that manages DAGs and mailbox database copies. No manual actions are required to resubmit messages from Safety Net. For more information about

Active Manager, see Active Manager.

There are two basic Safety Net message resubmission scenarios:

- After the automatic or manual failover of a mailbox database in a DAG.
- After you active a lagged copy of a mailbox database.

A *lagged mailbox database copy* or *lagged copy* is a passive copy of a mailbox database where updates to the database are intentionally delayed to protect against logical corruption of the mailbox database. For more information, see Managing mailbox database copies.

The only significant difference between the two scenarios is how far back in time to go to resubmit messages from Safety Net. Typically, for failover in a DAG, the new active copy of the mailbox database is typically several minutes to several hours behind the old active copy. A lagged copy of a mailbox database is typically several days behind the old active copy.

The main requirement for successful resubmission from Safety Net for a lagged copy is the amount of time messages are stored in Safety Net must be greater than or equal to the lag time of lagged copy of the mailbox database. In other words, the value of *SafetyNetHoldTime* on **Set-TransportConfig** must be greater than or equal to the value of the *ReplayLagTime* on **Set-MailboxDatabaseCopy** for the lagged copy.

Return to top

Message resubmission from Shadow Safety Net

Like message resubmission from Primary Safety Net, message resubmissions from Shadow Safety Net are fully automated, and require no manual intervention.

When the Active Manager requests message resubmission from Safety Net over a specific time period, the request goes to the Transport service on the Mailbox servers where Primary Safety Net is holding the message copies for the required time period. In large Exchange organizations, it's likely that the required messages exist in Safety Net on multiple Mailbox servers, particularly if the required time period is large.

Without optimization, resubmitting messages from Safety Net would result in potentially large numbers of duplicate deliveries. Duplicate deliveries within the Exchange organization aren't a problem, because duplicate message detection prevents mailbox users from seeing duplicate copies of a message. But duplicate message delivery to recipients outside the Exchange organization will result in duplicate copies of messages. Fortunately, the resubmission of messages from Safety Net has been optimized in Exchange 2013 to reduce duplicate message delivery.

If the Primary Safety Net is initially unresponsive, or becomes unresponsive during message resubmission, Active Manager continues to attempt to contact it for 12 hours before giving up. After 12 hours, a broadcast is sent to the Transport service on all the Mailbox servers in the transport high availability bound requesting resubmission of message from Safety Net for the required time interval for the required mailbox database. When a Shadow Safety Net responds, it resubmits the messages for the required mailbox database during the required time interval only.

There are some important considerations for the shadow messages stored in Shadow Safety Net:

- Shadow Safety Net doesn't know where the primary server transmitted the primary message.
- The shadow messages in Shadow Safety Net only contain original message envelope recipients, not the actual recipients where the primary message was delivered. For example, the message envelope recipient may be a distribution group that requires expansion.
- The messages in Shadow Safety net don't have any of the message updates that occurred after the primary server processed the message. For example, message encoding or content conversion.

Shadow message resubmitted from Shadow Safety Net require full categorization and processing through the Transport service on the Mailbox server. Resubmission of large numbers of shadow messages from Shadow Safety Net can be expensive in terms of Mailbox server resources. Fortunately, resubmission of shadow messages from Shadow Safety Net is also optimized so only messages in the Shadow Safety Net for the requested time interval and the requested mailbox database are resubmitted.

The interaction of Primary Safety Net and Shadow Safety Net during message resubmission is described in the following scenario.

1. Active Manager requests a resubmission of messages from Safety Net for a mailbox database for the time interval 5:00 to 9:00. However, the Mailbox server that holds the Primary Safety Net has crashed due to a hardware failure. Active Manager repeatedly tries to contact the Primary Safety Net for 12 hours.
2. After 12 hours, Active Manager sends a broadcast message to the Transport service on all Mailbox servers in the transport high availability boundary looking for other Safety Nets that contain messages for the target mailbox database for the time interval 5:00 to 9:00. The Shadow Safety Net responds are resubmits messages for the mailbox database for the time interval 5:00 to 9:00.

An interesting interaction occurs if the Primary Safety Net was offline during part of the requested resubmit interval as described in the following scenario.

1. The queue database on Mailbox server that holds the Primary Safety Net is corrupt, and a new queue database is created at 7:00. All of the primary messages stored in the Primary Safety Net from 1:00 to 7:00 are lost, but the server is able to store copies of successfully delivered messages in Safety Net starting at 7:00.
2. Active Manager requests a resubmission of messages from Safety Net for a mailbox database for the time interval 1:00 to 9:00.
3. The Primary Safety Net resubmits messages for the time interval 7:00 to 9:00.
4. The Primary Safety Net sends a broadcast message to the Transport service on all Mailbox servers in the transport high availability boundary looking for other Safety Nets that contain messages for the target mailbox database for the time interval 1:00 to 7:00 for which the Primary Safety Net has no message. The Shadow Safety Net generates a second resubmit request on behalf of the Primary Safety Net for resubmitting the shadow messages for the target mailbox database for the time interval 1:00 to 7:00.

There are some other issues to consider when messages are resubmitted from Safety Net.

1. All delivery status notifications (DSNs) and non-delivery reports (NDRs) are suppressed for Safety Net resubmits. For example, if the primary message resulted in an NDR, the NDR for the resubmitted message won't be delivered.
2. Users removed from a distribution group may not receive a resubmitted message when the Shadow Safety Net resubmits the message. For example, a message is sent to a group containing User A and User B, and both recipients receive the message. User B is subsequently removed from the group. Later, a resubmit request from Primary Safety Net is made for the mailbox database that holds User B's mailbox. However, the Primary Safety Net is unavailable for more than 12 hours, so the Shadow Safety Net server responds and resubmits the affected message. During resubmission when the distribution group is expanded, User B isn't a member of the group, and won't receive a copy of the resubmitted message.
3. New Users added to a distribution group may receive an old resubmitted message when the Shadow Safety Net resubmits the message. For example, a message is sent to a group containing User A and User B, and both recipients receive the message. User C is subsequently added to the group. Later, a resubmit request from Primary Safety Net is made for the mailbox database that holds User C's mailbox. However, the Primary Safety Net server is unavailable for more than 12 hours, so the Shadow Safety Net server responds and resubmits the affected messages. During resubmission when the distribution group is expanded, User C is a member of the group, and will receive a copy of the resubmitted message.

Return to top

Transport logs

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-28*

Transport logs provide information about what's happening in the transport pipeline. The following transport logs are available in Microsoft Exchange Server 2013:

Agent logs Agent logging records the actions performed on a message by specific anti-spam agents. Agent logging is available in the Transport service on a Mailbox server. For more information, see Anti-spam agent logging.

Connectivity logs Connectivity logging records the outbound connection activity transport services. Connectivity logging is available in the Front End Transport service on Client Access servers, the Transport service on Mailbox servers, and the Mailbox Transport service on Mailbox servers. For more information, see Connectivity logging.

Message tracking and delivery reports Message tracking logs the details of all message activity as messages are transferred to and from an Exchange 2013 Mailbox server. Message tracking is

available in the Transport service on Mailbox servers, and in the Mailbox Transport service on Mailbox servers. For more information, see Message tracking.

Delivery reports uses the information stored in the message tracking log to search for information about messages sent to or sent from a specific mailbox. For more information, see Delivery reports for administrators.

Pipeline tracing Pipeline tracing records snapshots of messages before and after the message is affected by transport agents in the Transport service on Mailbox servers, and in the Mailbox Transport Delivery service on Mailbox servers. For more information, see Pipeline tracing.

Protocol logs Protocol logging records the SMTP conversations that occur on Send connectors and Receive connectors as part of message delivery. Protocol logging is available in the Front End Transport service on Client Access servers, the Transport service on Mailbox servers, and the Mailbox Transport service on Mailbox servers. For more information, see Protocol logging.

Routing table logs Routing table logging periodically records a snapshot of the routing table that's used by Exchange 2013 to route messages to their destinations. Routing table logging is available in the Transport service on Mailbox servers.

Anti-spam agent logging

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-14*

Agent logs record the actions performed on a message by specific anti-spam agents in Microsoft Exchange Server 2013. Only the following agents can write information to the agent log:

- Connection Filtering agent
- Content Filter agent
- Edge Rules agent
- Recipient Filter agent
- Sender Filter agent
- Sender ID agent

Note:
The Connection Filtering agent and the Edge Rules agent aren't available on Mailbox servers.

The information written to the agent log depends on the agent, the SMTP event, and the action performed on the message.

You use the **Set-TransportService** cmdlet in the Exchange Management Shell to perform all agent log configuration tasks. The following options are available for the agent logs:

- Enable or disable agent logging. The default is enabled.

- Specify the location of the agent log files. The default value is %ExchangeInstallPath% TransportRoles\Logs\Hub\AgentLog.
- Specify a maximum size for the individual agent log files. The default size is 10 megabytes (MB).
- Specify a maximum size for the directory that contains agent log files. The default size is 250 MB.
- Specify a maximum age for the agent log files. The default age is 7 days.

Exchange uses circular logging to limit the agent logs based on file size and file age to help control the hard disk space used by the log files.

Contents

Overview of transport agents

Agents can only act upon messages at specific points in the SMTP command sequence used to transport the messages through the Transport service on a Mailbox server or an Edge Transport server. These access points in the SMTP command sequence are called *SMTP events*. Each agent has a priority value that can be assigned. However, the SMTP events must always occur in a specific order. Therefore, the agent priority depends on the SMTP event. If two agents can act on a message during the same SMTP event, the agent that has the highest priority will act on the message first.

The following table lists the SMTP events in order of occurrence and the agents that write information to the agent log in order of priority from highest to lowest for each SMTP event.

SMTP events in order of occurrence and the agents that write information to the agent log in order of priority for each SMTP event

SMTP event	Agent
OnConnect	Connection Filtering agent
OnMailCommand	Connection Filtering agent Sender Filter agent
OnRcptCommand	Connection Filtering agent Recipient Filter agent
OnEndOfHeaders	Connection Filtering agent Sender ID agent

	Sender Filter agent
OnEndOfData	Edge Rules agent Content Filtering agent

Note:
The Connection Filtering agent and the Edge Rules agent aren't available on Mailbox servers.

For more information about agents, SMTP events, and agent priority, see Transport agents.

Return to top

Structure of the agent log files

The agent logs exist in %ExchangeInstallPath%TransportRoles\Logs\Hub\AgentLog.

The naming convention for the agent log files is AGENTLOG*yyyymmdd-nnnn*.log. The placeholders represent the following information:

- The placeholder *yyyymmdd* is the Coordinated Universal Time (UTC) date that the log file was created. The placeholder *yyyy* = year, *mm* = month, and *dd* = day.
- The placeholder *nnnn* is an instance number that starts at the value of 1 for each day.

Information is written to the log file until the file size reaches its maximum specified value, and a new log file that has an incremented instance number is opened. This process is repeated throughout the day. Circular logging deletes the oldest log files when the agent log directory reaches its maximum specified size, or when a log file reaches its maximum specified age.

The agent log files are text files that contain data in the comma-separated value file (CSV) format. Each agent log file has a header that contains the following information:

- **#Software** Name of the software that created the agent log file. Typically, the value is Microsoft Exchange Server.
- **#Version** Version number of the software that created the agent log file. Currently, the value is 15.0.0.0.
- **#Log-Type** Log type value, which is Agent Log.
- **#Date** UTC date-time when the log file was created. The UTC date-time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where *yyyy* = year, *mm* = month, *dd* = day, T indicates the beginning of the time component, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and Z signifies Zulu, which is another way to denote UTC.
- **#Fields** Comma delimited field names used in the agent log files.

Return to top

Information written to the agent log

The agent log stores each agent transaction on a single line in the log. The information stored on each line is organized by fields. These fields are separated by commas. The field name is generally descriptive enough to determine the type of information it contains. However, some of the fields may be blank. Or the type of information stored in the field may change based on the agent or the action performed on the message by the agent. The following table describes the fields used to classify each agent transaction.

Fields used to classify each agent transaction

Field name	Description
Timestamp	UTC date-time of the agent event. The UTC date-time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where *yyyy* = year, *mm* = month, *dd* = day, T indicates the beginning of the time component, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and Z signifies Zulu, which is another way to denote UTC.
SessionId	Unique SMTP session identifier. This identifier is represented as a 16-digit hexadecimal number.
LocalEndpoint	Local IP address and port number that accepted the message. SMTP sessions typically use port 25.
RemoteEndpoint	IP address and port number of the previous SMTP server that connected to this server to deliver the message. When Internet mail flows through an Edge Transport server in the perimeter network, the value of **RemoteEndpoint** in the agent log on the Mailbox server will be the IP address of the Edge Transport server. Even though the message is transmitted by SMTP, the port number used by the sending server will be a random number larger than 1,024.

EnteredOrgFromIP	IP address of the remote SMTP server that first connected to the Exchange organization to deliver the message. On an Edge Transport server, the value of **RemoteEndpoint** and **EnteredOrgFromIP** are the same. Anti-spam agents use the IP address in **EnteredOrgFromIP** to examine a message.
MessageId	Value of the `MessageID` header field. If this value is blank, the Exchange transport server assigns an arbitrary value, but only if the message is accepted. After a value is assigned, the value of `MessageID` is constant for the lifetime of the message.
P1FromAddress	Sender email address specified in `MAIL FROM` in the message envelope. This value is used to transport the message between SMTP messaging servers. This value serves as a comparison to the value of **P2FromAddresses** to determine whether the sender address in the message header is forged.
P2FromAddresses	Sender email address specified in the `From` header field or in the `Sender` header field in the message header.
Recipient	Email address of the recipients. Although the original message may contain multiple recipients, only one recipient is displayed per line in the agent log.
NumRecipients	Total number of recipients in the original message.
Agent	Name of the agent that took the action. The

	possible values are as follows: • Content Filter agent • Recipient Filter agent • Sender Filter agent • Sender ID agent
Event	SMTP event where the action was taken by the agent. The value of **Event** depends on the agent. The SMTP events available to each agent are described in the first table earlier in this topic. The possible values for **Event** are as follows: • OnConnect • OnEndOfHeaders • OnEndOfData • OnMailCommand • OnRcptCommand
Action	Action performed on the message by the agent. The possible values for **Action** are as follows: • AcceptMessage • DeleteMessage • DeleteRecipients • Disconnect • QuarantineMessage • QuarantineRecipients • RejectAuthentication • RejectCommand • RejectConnection • RejectMessage • RejectRecipients
SmtpResponse	Enhanced SMTP response as defined in RFC 2034.
Reason	Reason for the action supplied by the agent.

ReasonData	Descriptive details for the action supplied by the agent.

Return to top

Search the agent logs

You can use the **Get-AgentLog** cmdlet and the **Get-AntiSpamFilteringReport.ps1** script to search the agent logs.

The **Get-AntiSpamFilteringReport.ps1** script is located in `%ExchangeInstallPath%Scripts`. You need to run the script in the Shell from the Scripts folder. To change your location in the Shell to the Scripts folder, run the following command:

```
Cd $env:ExchangeInstallPath\Scripts
```

To run the script in the Scripts folder, use the following syntax:

```
.\Get-AntiSpamFilteringReport.ps1 -report <ReportValue>
[<OptionalParameters>]
```

For details about using the script, run the following command:

```
Get-Help -Detailed .\Get-AntiSpamFilteringReport.ps1
```

Return to top

Configure anti-spam agent logging

Mail flow > Transport logs > Anti-spam agent logging >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

Agent logging records the actions performed by specific Exchange anti-spam agents. The information written to the agent log depends on the agent, the SMTP event, and the action performed on the message.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "Transport Service" and "Edge Transport server" entries in the Mail flow permissions topic.

- By default, anti-spam features aren't enabled in the Transport service on a Mailbox server. Typically, you only enable the anti-spam features on a Mailbox server if your Exchange organization doesn't do any prior anti-spam filtering before accepting incoming messages. For more information, see Enable anti-spam functionality on Mailbox servers.
- You can only use the Shell to perform this procedure.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to configure anti-spam agent logging

Run the following command:

```
Set-TransportService <ServerIdentity> -AgentLogEnabled
<$true | $false> -AgentLogMaxAge <dd.hh:mm:ss> -
AgentLogMaxDirectorySize <Size> -AgentLogMaxFileSize <Size>
-AgentLogPath <LocalFilePath>
```

This example sets the following agent log settings on the Mailbox server named Mailbox01:

Sets the location of the agent log files to D:\Anti-Spam Agent Log. Note that if the folder doesn't exist, it will be created for you.

Sets the maximum size of an agent log file to 20 MB.

Sets the maximum size of the agent log directory to 400 MB.

Sets the maximum age of an agent log file to 14 days.

```
Set-TransportService Mailbox01 -AgentLogPath "D:\Anti-Spam
Agent Log" -AgentLogMaxFileSize 20MB -
AgentLogMaxDirectorySize 400MB -AgentLogMaxAge 14.00:00:00
```

Note:

- If you set the *AgentLogPath* parameter to the value `$null`, you effectively disable agent logging. However, if you set *AgentLogPath* to `$null` when the value of the *AgentLogEnabled* parameter is `$true`, event log errors are generated. The preferred method to disable agent

logging is to set *AgentLogEnabled* to `$false`.

- Setting the *AgentLogMaxAge* parameter to the value `00:00:00` prevents the automatic removal of agent log files because of their age.

For detailed syntax and parameter information, see the *AgentLog* parameters in Set-TransportService.

How do you know this worked?

To verify that you have successfully configured anti-spam agent logging, do the following:
1. In the Shell, run the following command:

```
Get-TransportService <ServerIdentity> | Format-List AgentLog*
```

2. Verify the values displayed are the values you configured.

Connectivity logging

Exchange Server 2013 > Mail flow > Transport logs >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-15*

Connectivity logging records the outbound connection activity that's used to transmit messages from a transport service on the Exchange server. The purpose of the connectivity log isn't to track the transmission of individual email messages. Rather, the connectivity log tracks the connection activity from source to the destination, regardless of how many messages are transmitted. Connectivity logging is available in the Front End Transport service on Client Access servers, the Transport service on Mailbox servers, and the Mailbox Transport service on Mailbox servers. The following list describes the type of information recorded in the connectivity log:

- Source
- Destination
- DNS resolution information
- Detailed information about connection failures
- Number of messages and bytes transmitted

You use the **Set-TransportService**, **Set-FrontEndTransportService** and **Set-MailboxTransportService** cmdlets in the Exchange Management Shell to perform all connectivity log configuration tasks. The following options are available for the connectivity logs:

- Enable or disable connectivity logging. The default is enabled.

- Specify the location of the connectivity log files.
- Specify a maximum size for the individual connectivity log files. The default size is 10 megabytes (MB).
- Specify a maximum size for the directory that contains connectivity log files. The default size is 1000 MB.
- Specify a maximum age for the connectivity log files. The default age is 30 days.

By default, Exchange uses circular logging to limit the connectivity logs based on file size and file age to help control the hard disk space used by the connectivity log files.

Contents

Structure of the connectivity log files

By default, the connectivity log files exist in the following locations:

- **Transport service** %ExchangeInstallPath%TransportRoles\Logs\Hub\Connectivity
- **Front End Transport service** %ExchangeInstallPath%TransportRoles\Logs\FrontEnd\Connectivity
- **Mailbox Transport service** %ExchangeInstallPath%TransportRoles\Logs\Mailbox\Connectivity

The naming convention for the connectivity log files is CONNECTLOG*yyyymmdd-nnnn*.log. The placeholders represent the following information:

- The placeholder *yyyymmdd* is the Coordinated Universal Time (UTC) date that the log file was created. The placeholder *yyyy* = year, *mm* = month, and *dd* = day.
- The placeholder *nnnn* is an instance number that starts at the value of 1 for each day.

Information is written to the log file until the file size reaches its maximum specified value, and a new log file that has an incremented instance number is opened. This process is repeated throughout the day. Circular logging deletes the oldest log files when the connectivity log directory reaches its maximum specified size, or when a log file reaches its maximum specified age.

The connectivity log files are text files that contain data in the comma-separated value file (CSV) format. Each connectivity log file has a header that contains the following information:

- **#Software** Name of the software that created the connectivity log file. Typically, the value is Microsoft Exchange Server.
- **#Version** Version number of the software that created the connectivity log file. Currently, the value is 15.0.0.0.
- **#Log-Type** Log type value, which is Transport Connectivity Log.
- **#Date** UTC date-time when the log file was created. The UTC date-time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where *yyyy* = year, *mm* = month, *dd* = day, T indicates the beginning of the time component, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and Z signifies Zulu, which is another way to denote UTC.
- **#Fields** Comma delimited field names used in the connectivity log files.

Return to top

Information written to the connectivity log

The connectivity log stores each outbound transport service connection event on a single line in the connectivity log. The information stored on each line is organized by fields. These fields are separated by commas. The following table describes the fields used to classify each outgoing connection event.

Fields used to classify each connection event

Field name	Description
date-time	UTC date-time of the connection event. The UTC date-time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where *yyyy* = year, *mm* = month, *dd* = day, T indicates the beginning of the time component, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and Z signifies Zulu, which is another way to denote UTC.
session	GUID that's unique for each SMTP session but is the same for each event associated with that SMTP session. For MAPI sessions in the Mailbox Transport service, the session field is blank.
source	**SMTP** for SMTP connections, **MAPI** for Mailbox Transport service connections to the local mailbox database.
destination	Name of the destination.
direction	Single character that represents the start, middle, or end of the connection. The possible values for the direction field are as follows: • **+** Connect • **-** Disconnect • **>** Send

description	Text information associated with the connection event. The following values are examples of values for the description field: • Number and size of messages that were transmitted • DNS MX resource record resolution information for destination domains • DNS resolution information for destination Mailbox servers • Connection establishment messages • Connection failure messages

When transport service establishes a connection to a destination, the transport service may be prepared to send one message or several messages. The connection and message transmission processes generate multiple events written on multiple lines in the connectivity log. Simultaneous connections to different destinations create connectivity log entries related to different destinations that are interlaced. However, you can use the date-time, session, source, and direction fields to arrange the connectivity log entries for each separate connection from start to finish.

Return to top

Configure connectivity logging

Mail flow > Transport logs > Connectivity logging >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-18*

Connectivity logging records the outbound connection activity that's used to transmit messages from a transport service on an Exchange server. Connectivity logging records the connection source, destination, number of messages and bytes transmitted, and connection failure information.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport service", "Front End Transport service", and "Mailbox Transport service" entries in the Mail flow permissions topic.
- You can use the Exchange admin center (EAC) to enabled or disabled connectivity logging, or set

the connectivity log path for the Transport service only. For all other connectivity logging options in other transport services, you need to use the Shell.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to configure connectivity logging in the Transport service

1. In the EAC, navigate to **Servers** > **Servers**.
2. Select the Mailbox server you want to configure, and then click **Edit** .
3. On the server properties page, click **Transport Logs**.
4. In the **Connectivity log** section, change any of the following:
   - **Enable connectivity log** To disable connectivity logging on the server, clear the check box. To enable connectivity logging on the server, select the check box.
   - **Connectivity log path** The value you specify must be on the local Exchange server. If the folder doesn't exist, it will be created for you when you click **Save**.

When you are finished, click **Save**.

Use the Shell to configure connectivity logging

To configure connectivity logging, run the following command:

```
<Set-TransportService | Set-MailboxTransportService | Set-
FrontEndTransportService> <ServerIdentity> -
ConnectivityLogEnabled <$true | $false> -
ConnectivityLogMaxAge <dd.hh:mm:ss> -
ConnectivityLogMaxDirectorySize <Size> -
ConnectivityLogMaxFileSize <Size> -ConnectivityLogPath
<LocalFilePath>
```

This example sets the following connectivity log settings in the Transport service on the Mailbox server named Mailbox01:

Sets the location of the connectivity log files to D:\Hub Connectivity Log. Note that if the folder doesn't exist, it will be created for you.

Sets the maximum size of a connectivity log file to 20 MB.

Sets the maximum size of the connectivity log directory to 1.5 GB.

Sets the maximum age of a connectivity log file to 45 days.

```
Set-TransportService Mailbox01 -ConnectivityLogPath "D:\Hub
Connectivity Log" -ConnectivityLogMaxFileSize 20MB -
ConnectivityLogMaxDirectorySize 1.5GB -
ConnectivityLogMaxAge 45.00:00:00
```

Note:

- To configure the connectivity log settings in the Mailbox Transport service on a Mailbox server, use the **Set-MailboxTransportService** cmdlet. To configure the connectivity log settings in the Front End Transport service on a Client Access server, use the **Set-FrontEndTransportService** cmdlet.
- Setting the *ConnectivityLogPath* parameter to the value `$null`, effectively disables connectivity logging. However, if the value of the *ConnectivityLogEnabled* parameter is `$true`, event log errors are generated.
- Setting the *ConnectivityLogMaxAge* parameter to the value `00:00:00` prevents the automatic removal of connectivity log files because of their age.

How do you know this worked?

To verify that you have successfully configured connectivity logging, do the following:
1. In the Shell, run the following command:

```
<Get-TransportService | Get-FrontEndTransportService | Get-
MailboxTransportService> <ServerIdentity> | Format-List
ConnectivityLog*
```

2. Verify the values displayed are the values you configured.

Pipeline tracing

Exchange Server 2013 > Mail flow > Transport logs >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-28*

Pipeline tracing captures copies of email messages from a specific sender as they move through the Transport service on Mailbox servers, the Mailbox Transport Delivery service on Mailbox servers, and through Edge Transport servers.. Pipeline tracing captures verbose information about the changes that each transport agent applies to messages in the transport pipeline in message snapshot files. By examining the contents of the message snapshot files, you can determine whether the transport agents have applied the changes to the messages in the transport pipeline that you expected. If you are troubleshooting a problem, you should determine which transport agent is at fault. Then you can focus your troubleshooting efforts on that agent to resolve the problem. You can then view the message snapshot files again to verify that your solution is successful.

Caution:

- Pipeline tracing copies the complete contents of email messages that are sent from the sender's email address. To avoid unwanted exposure of confidential information, you need to set appropriate security permissions on the pipeline tracing folder.
- Don't enable pipeline tracing for long periods of time. Pipeline tracing creates files that can accumulate quickly. Always monitor available disk space when pipeline tracing is enabled.

Configure pipeline tracing

Before you enable pipeline tracing, you need to specify the sender's email address you want to monitor. Pipeline tracing is designed to log messages sent from a specific email address. The sender's email address can be internal or external to your Exchange organization. Alternatively, you can enable pipeline tracing for system messages generated by the transport service on the specified Mailbox or Edge Transport server, such as automatic replies, delivery status notification (DSN) messages, journal reports, and other system-generated messages You can also modify the location of the pipeline tracing folder.

The parameters that you use to configure pipeline tracing are summarized in the following table

Cmdlet	Parameter	Default value	Description
Set-TransportService **Set-MailboxTransportService**	*PipelineTracingSenderAddress*	Blank (`$null`)	Specify the email address of the sender you want to monitor. Specify the value **"<>"** to monitor system-generated messages sent by the specified transport service on the

			server.
Set-TransportService **Set-MailboxTransportService**	*PipelineTracingPath*	**Transport service** `%ExchangeInstallPath%TransportRoles\Logs\Hub\PipelineTracing` **Mailbox Transport service** `%ExchangeInstallPath%TransportRoles\Logs\Mailbox\PipelineTracing`	The path must be on the local server. UNC paths aren't supported. The specified path contains the `MessageSnapshots` folder where pipeline tracing files are stored.
Set-TransportService **Set-MailboxTransportService**	*PipelineTracingEnabled*	`$false`	You can only enable pipeline tracing for the specified transport service on the server after you configure the sender address you want to monitor.

For more information about how to enable pipeline tracing and configure the sender address for pipeline tracing, see Configure pipeline tracing.

Message snapshot files

Message snapshots are files that capture any changes made to a message by transport agents in the Transport service or the Mailbox Transport Delivery service. These files are stored in the `MessageSnapshots` folder in the corresponding pipeline tracing path for the transport service.

In the `MessageSnapshots` folder, Exchange creates one folder for each message sent by the monitored sender that flows through the specified transport service. Each folder is named after a GUID that's assigned to the message. If you enable pipeline tracing for the Transport service and the Mailbox Transport service on the same Mailbox server, a different GUID is assigned to the same message by each transport service, so the folder name for a message in the `MessageSnapshots` folder for the Transport service is different than the folder name for the same message in the `MessageSnapshots` folder for the Mailbox Transport service. If you enable pipeline tracing on more than one Exchange server, a different GUID is assigned to the same message as it travels through the specified transport service on each Exchange server.

In each message folder, Exchange creates several message snapshot files that have .eml file extensions. These message snapshot files contain the contents of the message as it encounters each

SMTP event and transport agent.

If a transport agent is registered on an SMTP event, Exchange creates a message snapshot of the message before the message encounters any transport agents. This gives you a copy of the message before the message encounters transport agents that are registered on that event. Then, a new message snapshot is created for each transport agent that the message encounters, regardless of whether a transport agent modifies the contents of the message. However, if no agents are registered on an event, Exchange doesn't create any messages snapshots for that event.

For example, if three agents are registered on the **OnEndofData** event but only two of the transport agents modify a message, four message snapshots are created. The first message snapshot captures the message as it encounters the **OnEndofData** event before any modifications that are made by the transport agents that registered on that event. Then, one message snapshot is created for each transport agent regardless of whether a transport agent modifies the message.

The message snapshot files that are created are described in the following list:

- **Original.eml** This file contains the original unmodified contents of the email message before it encounters any SMTP events or transport agents.
- **Routingnnnn.eml** These files contain the contents of the email message as it encounters transport the SMTP events and transport agents registered on those events in the categorization part of the Transport service. The placeholder *nnnn* represents an integer value that starts with `0001`. The value is incremented for every SMTP event and transport agent registered on those events in the order in which the events and agents act on the message. The Mailbox Transport Delivery service doesn't generate these **Routing** snapshot files.
- **SmtpReceivennnn.eml** These files contain the contents of the email message as it encounters the **OnEndofData** and **OnEndOfHeaders** SMTP events and transport agents registered on those events during the SMTP receive part of the Transport service or the Mailbox Transport Delivery service. The placeholder *nnnn* represents an integer value that starts with `0001`. The value is incremented for every SMTP event and transport agent registered on those events in the order in which the events and agents act on the message.

You can open the message snapshot files by using Notepad or any text editor.

Each message snapshot file starts with headers that are added to the message contents and list the SMTP event and transport agent that the message snapshot file relates to. These headers start with `X-CreatedBy: MessageSnapshot-Begin injected headers` and end with `X-EndOfInjectedXHeaders: MessageSnapshot-End injected headers`. These headers are replaced in each message snapshot file by each subsequent transport agent and SMTP event. The following is an example of the headers that are added to an email message file:

```
X-CreatedBy: MessageSnapshot-Begin injected headers
X-MessageSnapshot-UTC-Time: 2013-01-23T23:20:18.138Z
X-MessageSnapshot-Record-Id: 21474836486
X-MessageSnapshot-Source: OnSubmittedMessageX-Sender:
michelle@nwtraders.com
```

```
X-Receiver: chris@contoso.com
X-EndOfInjectedXHeaders: MessageSnapshot-End injected
headers
```

After the message snapshot headers, the file contains the contents of the message including all the original message headers. If a transport agent modifies the contents of the message, the changes appear integrated with the message. As the message is processed by each transport agent, the changes that are made by each agent are applied to the message contents. If a transport agent makes no changes to the message contents, the message snapshot that is created by that agent will be identical to the message snapshot created by the previous transport agent.

Configure pipeline tracing

Mail flow > Transport logs > Pipeline tracing >

Topic Last Modified: *2013-11-08*

Pipeline tracing captures copies of email messages as they move through the transport pipeline in the Transport service or the Mailbox Transport service on Mailbox server and on Edge Transport servers.

What do you need to know before you begin?

- Estimated time to complete this procedure: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the"Transport Service" and "Mailbox Transport Service" entries in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.
- Pipeline tracing copies the complete contents of email messages that are sent from the sender's email address. To avoid unwanted exposure of confidential information, you need to set appropriate security permissions on the location of the pipeline tracing folder.
- Don't enable pipeline tracing for long periods of time. Pipeline tracing creates multiple message snapshot files that accumulate quickly. Always monitor available disk space when pipeline tracing is enabled.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Enable and configure pipeline tracing

Step 1: Use the Shell to configure the pipeline tracing sender address

Use the following syntax to configure the pipeline tracing sender address.

```
<Set-TransportService | Set-MailboxTransportService>
<ServerIdentity> -PipelineTracingSenderAddress <SMTPAddress
| "<>">
```

This example configures pipeline tracing to capture snapshots of all messages sent by the sender chris@contoso.com in the Transport service on the Mailbox server named Mailbox01.

```
Set-TransportService Mailbox01 -
PipelineTracingSenderAddress chris@contoso.com
```

This example configures pipeline tracing to capture snapshots of all the system-generated messages received by the Transport service on the Mailbox server named Mailbox02.

```
Set-TransportService Mailbox02 -
PipelineTracingSenderAddress "<>"
```

Caution:

Configuring pipeline tracing to capture all server-generated messages in a transport service may place a significant load on the server and may quickly consume available disk space. Always monitor available disk space when pipeline tracing is enabled.

Step 2: (Optional) Use the Shell to specify a custom pipeline tracing folder

The default pipeline tracing folder doesn't exist until after you enable pipeline tracing, and messages that meet the criteria you specify using the *PipelineTracingSenderAddress* parameter flow through the transport service on the server. For the Transport service on a Mailbox server, the default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\PipelineTracing`. For the Mailbox Transport service on a Mailbox server, the default location is `%ExchangeInstallPath%TransportRoles\Logs\Mailbox\PipelineTracing`. If you specify a custom path, the path must be on the local Exchange server.

Use the following syntax to configure the pipeline tracing folder.

```
<Set-TransportService | Set-MailboxTransportService>
<ServerIdentity> -PipelineTracingPath <LocalFilePath>
```

This example sets the pipeline tracing folder for the Transport service on the Mailbox server named Mailbox01 to D:\Hub\Pipeline Tracing.

```
Set-TransportService Mailbox01 -PipelineTracingPath "D:\Hub
\Pipeline Tracing"
```

Step 3: Use the Shell to enable pipeline tracing

By default, pipeline tracing is disabled on all Exchange servers. When you enable pipeline tracing, you are enabling pipeline tracing in the specified transport service on the specified Exchange server only. Before you enable pipeline tracing, you need to specify the sender address as described in Step 1.

Use the following syntax to enable pipeline tracing.

```
<Set-TransportService | Set-MailboxTransportService>
<ServerIdentity> -PipelineTracingEnabled $true
```

This example enables pipeline tracing in the Transport service on the Mailbox server named Mailbox01.

```
Set-TransportService Mailbox01 -PipelineTracingEnabled
$true
```

How do you know this worked?

To verify that you have successfully configured pipeline tracing, do the following:
1. Run the following command:

```
<Get-TransportService | Get-MailboxTransportService>
<ServerIdentity> | Format-List PipelineTracing*
```

2. Verify the values displayed are the values you configured.
3. Check the pipeline tracing folder for the Transport service or the Mailbox Transport service, and verify message snapshot files are being created in the folder.

Disable pipeline tracing

Because of the disk space and security concerns associated with pipeline tracing, pipeline tracing is a temporary action for diagnostic or troubleshooting purposes. Whenever you enable pipeline tracing, always remember to disable it when you are finished.

Use the following syntax to disable pipeline tracing.

```
<Set-TransportService | Set-MailboxTransportService>
<ServerIdentity> -PipelineTracingEnabled $false
```

This example disables pipeline tracing in the Transport service on the Mailbox server named Mailbox01.

```
Set-TransportService Mailbox01 -PipelineTracingEnabled
$false
```

How do you know this worked?

To verify that you have successfully disabled pipeline tracing, do the following:
1. Run the following command:

```
<Get-TransportService | Get-MailboxTransportService>
<ServerIdentity> | Format-List PipelineTracingEnabled
```

2. Verify the value of the *PipelineTracingEnabled* parameter is $false.
3. Check the pipeline tracing folder, and verify message snapshot files are no longer being created in the folder.

Protocol logging

Exchange Server 2013 > Mail flow > Transport logs >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-30*

Protocol logging records the SMTP conversations that occur between messaging servers as part of message delivery. These SMTP conversations occur on Send connectors and Receive connectors that exist in the Front End Transport service on Client Access servers, the Transport service on Mailbox servers, and the Mailbox Transport service on Mailbox servers. You can use protocol logging to diagnose mail flow problems.

By default, protocol logging is disabled on all Send connectors and Receive connectors. Protocol logging is enabled or disabled on each individual connector. Other protocol logging options are set for all the Receive connectors or all the Send connectors that exist in each individual transport service on the server. All the Receive connectors in a transport service share the same protocol log files and protocol log options. These protocol log files and protocol log options are separate from the Send connector protocol log files and protocol log options in the transport service on the same server.

The following options are available for the protocol logs of all Send connectors or all Receive connectors in each transport service on the Exchange server:
- Specify the location of the Send connector or the Receive connector protocol log files.
- Specify a maximum size for the Send connector or the Receive connector protocol log files. The

default size is 10 megabytes (MB).
- Specify a maximum size for the directory that contains the Send connector or Receive connector protocol log files. The default size is 250 MB.
- Specify a maximum age for the Send connector or Receive connector protocol log files. The default age is 30 days.

By default, Exchange uses circular logging to limit the protocol logs based on file size and file age to help control the hard disk space used by the log files.

A special Send connector named the intra-organization Send connector exists in the Transport service on every Mailbox server, and in the Front End Transport service on every Client Access server. This connector is implicitly created, invisible, and requires no management. The intra-organization Send connector is used by the following transport services:

- **Transport service on Mailbox servers**
  - Relays messages to the Transport service and the Mailbox Transport service on other Exchange 2013 Mailbox servers in the organization.
  - Relays messages to other Exchange 2007 or Exchange 2010 Hub Transport servers in the organization.
  - Relays messages to Edge Transport servers in the perimeter network.
- **Front End Transport service on Client Access servers** Relays messages to the Transport service on Exchange 2013 Mailbox servers in the organization.

An equivalent Send connector named the mailbox delivery Send connector exists in the Mailbox Transport service on every Mailbox server. This connector is also implicitly created, invisible, and requires no management. The mailbox delivery Send connector is used to relay messages to the Transport service and the Mailbox Transport service on other Mailbox servers in the organization.

By default, protocol logging for the mailbox delivery Send connector is also disabled. You can enable or disable protocol logging for the mailbox delivery Send connector by using the *MailboxDeliveryConnectorProtocolLoggingLevel* parameter on the **Set-MailboxTransportService** cmdlet. If you enable protocol logging for the mailbox delivery Send connector, logging occurs in the Send connector protocol logs for the Mailbox Transport service on the Mailbox server.

Contents

Structure of the protocol log files

By default, the protocol log files exist in the following locations:

- **Receive connector protocol log files for the Transport service on Mailbox servers** %ExchangeInstallPath%TransportRoles\Logs\Hub\ProtocolLog\SmtpReceive
- **Receive connector protocol log files for the Mailbox Transport service on Mailbox servers** %ExchangeInstallPath%TransportRoles\Logs\Mailbox\ProtocolLog\SmtpReceive
- **Receive connector protocol log files for the Front End Transport service on Client Access**

servers %ExchangeInstallPath%TransportRoles\Logs\FrontEnd\ProtocolLog\SmtpReceive

- **Send connector protocol log files for the Transport service on Mailbox servers** % ExchangeInstallPath%TransportRoles\Logs\Hub\ProtocolLog\SmtpSend
- **Send connector protocol log files for the Mailbox Transport service on Mailbox servers** % ExchangeInstallPath%TransportRoles\Logs\Mailbox\ProtocolLog\SmtpSend
- **Send connector protocol log files for the Front End Transport service on Client Access servers** %ExchangeInstallPath%TransportRoles\Logs\FrontEnd\ProtocolLog\SmtpSend

The naming convention for log files in each protocol log directory is *prefixyyyymmdd-nnnn*.log. The placeholders represent the following information:

- The placeholder *prefix* is SEND for Send connectors or RECV for Receive connectors.
- The placeholder *yyyymmdd* is the Coordinated Universal Time (UTC) date on which the log file was created. The placeholder *yyyy* = year, *mm* = month, and *dd* = day.
- The placeholder *nnnn* is an instance number that starts at the value of 1 for each day.

Information is written to the log file until the file size reaches its maximum specified value, and a new log file that has an incremented instance number is opened. This process is repeated throughout the day. Circular logging deletes the oldest log files when the protocol log directory reaches its maximum specified size, or when a log file reaches its maximum specified age.

The protocol log files are text files that contain data in the comma-separated value file (CSV) format. Each protocol log file has a header that contains the following information:

- **#Software** Name of the software that created the protocol log file. Typically, the value is Microsoft Exchange Server.
- **#Version** Version number of the software that created the protocol log file. Currently, the value is 15.0.0.0.
- **#Log-Type** Log type value of this field, which is either SMTP Receive Protocol Log or SMTP Send Protocol Log.
- **#Date** UTC date-time when the log file was created. The UTC date-time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where *yyyy* = year, *mm* = month, *dd* = day, T indicates the beginning of the time component, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and Z signifies Zulu, which is another way to denote UTC.
- **#Fields** Comma-delimited field names used in the protocol log files.

Return to top

Information written to the protocol log

The protocol log stores each SMTP protocol event on a single line in the protocol log. The information stored on each line is organized by fields. These fields are separated by commas. The following table describes the fields used to classify each protocol.

Fields used to classify each protocol event

Field name	Description

date-time	UTC date-time of the protocol event. The UTC date-time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where *yyyy* = year, *mm* = month, *dd* = day, T indicates the beginning of the time component, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and Z signifies Zulu, which is another way to denote UTC.
connector-id	Distinguished name (DN) of the connector associated with the SMTP event.
session-id	GUID that's unique for each SMTP session but is the same for each event associated with that SMTP session.
sequence-number	Counter that starts at 0 and is incremented for each event in the same SMTP session.
local-endpoint	Local endpoint of an SMTP session. This consists of an IP address and TCP port number formatted as <*IP address*>:<*port*>.
remote-endpoint	Remote endpoint of an SMTP session. This consists of an IP address and TCP port number formatted as <*IP address*>:<*port*>.
event	Single character that represents the protocol event. The possible values for the event are as follows: • **+** Connect • **-** Disconnect • **>** Send • **<** Receive • ***** Information

data	Text information associated with the SMTP event.
context	Additional contextual information that may be associated with the SMTP event.

A single SMTP conversation that represents the sending or receiving of a single email message generates multiple SMTP events. These SMTP events cause multiple lines to be written to the protocol log. Multiple SMTP conversations that represent the sending or receiving of multiple email messages can occur at the same time. This creates protocol log entries from different SMTP conversations that are interspersed. You can use the session-id and sequence-number fields to sort the protocol log entries by SMTP conversation.

Return to top

Configure protocol logging

Mail flow > Transport logs > Protocol logging >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-15*

Protocol logging records the SMTP conversations that occur on Send Connectors and Receive connectors as part of message delivery.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport Service", "Front End Transport service", "Mailbox Transport service", "Receive connectors" and "Send connectors" entries in the Mail flow permissions topic.
- You can use the Exchange admin center (EAC) to enable or disable protocol logging for Send connectors and Receive connectors in the Transport service on Mailbox servers, and for Receive connectors in the Front End Transport service on Client Access servers. You can also use the EAC to configure the protocol log paths for the Transport service only. For all other protocol logging options, you need to use the Shell.
- Protocol logging is enabled or disabled on each individual connector. All the Receive connectors on the Exchange server share the same protocol log files and protocol log options. These protocol log settings are separate from the Send connector protocol log files and protocol log options that are on the same server.

Caution:

Don't perform this procedure on an Edge Transport server that has been subscribed to the Exchange organization by using EdgeSync. Instead, make the changes in the Transport service on the Mailbox server. The changes are then replicated to the Edge Transport server next time EdgeSync synchronization occurs.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to configure protocol logging

To use the EAC to enable or disable protocol logging on a Send connector or a Receive connector in the Transport service on a Mailbox server, or on a Receive connector in the Front End Transport service on a Client Access server, do the following:

1. In the EAC, navigate to **Mail flow** > **Send connectors** or **Mail flow** > **Receive connectors**.
2. Select the connector you want to configure, and then click **Edit** .
3. On the **General** tab in the **Protocol logging level** section, select one of the following options:
   - **None** Protocol logging disabled on the connector.
   - **Verbose** Protocol logging is enabled on the connector.

When you are finished, click **Save**.

To use the EAC to configure the protocol log paths for the Send connectors and Receive connectors in the Transport service on a Mailbox server, do the following:

1. In the EAC, navigate to **Servers** > **Servers**.
2. Select the Mailbox server you want to configure, and then click **Edit** .
3. On the server properties page, click **Transport logs**.
4. In the **Protocol log** section, change any of the following settings:
   - **Send protocol log path** The value you specify must be on the local Exchange server. If the folder doesn't exist, it will be created for you when you click **Save**.
   - **Receive protocol log path** The value you specify must be on the local Exchange server. If the folder doesn't exist, it will be created for you when you click **Save**.

When you are finished, click **Save**.

How do you know this worked?

To verify that you have successfully used the EAC to configure the protocol log settings, do the following:

1. Browse to the location you specified for the Send connector or the Receive connector protocol

logs.
2. If you enabled protocol logging, verify a log file is created. If you disabled protocol logging, verify the latest log file is no longer being updated.

Use the Shell to enable or disable protocol logging on a Send connector or a Receive connector

To enable or disable protocol logging on a Send connector or a Receive connector, run the following command:

```
<Set-SendConnector |Set-ReceiveConnector>
<ConnectorIdentity> -ProtocolLoggingLevel <Verbose | None>
```

This example enables protocol logging for the Receive connector named Connection from Contoso.com.

```
Set-ReceiveConnector "Connection from Contoso.com" -
ProtocolLoggingLevel Verbose
```

How do you know this worked?

To verify that you have successfully enabled or disabled protocol logging, do the following:
1. In the Shell, run the following command:

```
<Get-SendConnector |Get-ReceiveConnector> | Format-List
Name,ProtocolLoggingLevel
```

2. Verify the values displayed are the values you configured.

Use the Shell to enable or disable protocol logging on the intra-organization Send connector

To enable or disable protocol logging on the implicit and invisible intra-organization Send connector that exists in the Transport service on a Mailbox server and in the Front End Transport service on a Client Access server, run the following command:

```
<Set-TransportService | Set-FrontEndTransportService> -
IntraOrgConnectorProtocolLoggingLevel <Verbose | None>
```

This example enables protocol logging on the intra-organization Send connector in the Transport service on a Mailbox server named Mailbox01.

```
Set-TransportService Mailbox01 -
```

```
IntraOrgConnectorProtocolLoggingLevel Verbose
```

How do you know this worked?

To verify that you have successfully enabled or disabled protocol logging on the intra-org Send connector, do the following:
1. In the Shell, run the following command:

```
<Get-TransportService | Get-FrontEndTransportService>
<ServerIdentity> | Format-List
IntraOrgConnectorProtocolLoggingLevel
```

2. Verify the value displayed is the value you configured.

Use the Shell to enable or disable protocol logging on the mailbox delivery Send connector

To enable or disable protocol logging on the implicit and invisible mailbox delivery Send connector that exists in the Mailbox Transport service on a Mailbox server, run the following command:

```
Set-MailboxTransportService -
MailboxDeliveryConnectorProtocolLoggingLevel <Verbose |
None>
```

This example enables protocol logging on the mailbox delivery Receive connector in the Mailbox Transport service on a Mailbox server named Mailbox01.

```
Set-MailboxTransportService Mailbox01 -
MailboxDeliveryConnectorProtocolLoggingLevel Verbose
```

How do you know this worked?

To verify that you have successfully enabled or disabled protocol logging on the mailbox delivery connector, do the following:
1. In the Shell, run the following command:

```
Get-MailboxTransportService <ServerIdentity> | Format-List
MailboxDeliveryConnectorProtocolLoggingLevel
```

2. Verify the value displayed is the value you configured.

Use the Shell to configure protocol logging settings

To configure the protocol log settings, run the following command:

```
<Set-TransportService | Set-MailboxTransportService | Set-
FrontEndTransportService> <ServerIdentity> -
ReceiveProtocolLogPath <LocalFilePath> -SendProtocolLogPath
<LocalFilePath> -ReceiveProtocolLogMaxFileSize <Size> -
SendProtocolLogMaxFileSize <Size> -
ReceiveProtocolLogMaxDirectorySize <Size> -
SendProtocolLogMaxDirectorySize <Size> -
ReceiveProtocolLogMaxAge <dd.hh:mm:ss> -
SendProtocolLogMaxAge <dd.hh:mm:ss>
```

This example sets the following protocol log settings in the Transport service on the Mailbox server named Mailbox01:

Sets the location of all Receive connector protocol logs to D:\Hub Receive SMTP Log and all Send connector protocol logs to D:\Hub Send SMTP Log. Note that if the folder doesn't exist, it will be created for you.

Sets the maximum size of a Receive connector protocol log file and a Send connector protocol log file to 20 MB.

Sets the maximum size of the Receive connector protocol log folder and the Send connector protocol log folder to 400 MB.

Sets the maximum age of a Receive connector protocol log file and a Send Connector protocol log file to 45 days.

```
Set-TransportService Mailbox01 -ReceiveProtocolLogPath "D:
\Hub Receive SMTP Log" -SendProtocolLogPath "D:\Hub Send
SMTP Log" -ReceiveProtocolLogMaxFileSize 20MB -
SendProtocolLogMaxFileSize 20MB -
ReceiveProtocolLogMaxDirectorySize 400MB -
SendProtocolLogMaxDirectorySize 400MB -
ReceiveProtocolLogMaxAge 45.00:00:00 -SendProtocolLogMaxAge
45.00:00:00
```

Note:

- To configure the protocol log settings in the Mailbox Transport service on a Mailbox server, use the **Set-MailboxTransportService** cmdlet. To configure the protocol log settings in the Front End Transport service on a Client Access server, use the **Set-**

FrontEndTransportService cmdlet.

- Setting the *SendProtocolLogPath* or *ReceiveProtocolLogPath* parameters to the value `$null` effectively disables protocol logging for all Send connectors or all Receive connectors on the server. However, setting either of these parameters to `$null` when protocol logging is enabled for any other connectors on the server, including the intra-organization Send connector or the mailbox delivery Send connector, event log errors are generated.
- Setting the *ReceiveProtocolLogMaxAge* or *SendProtocolLogMaxAge* parameters to the value `00:00:00` prevents the automatic removal of protocol log files because of their age.

How do you know this worked?

To verify that you have successfully configured the protocol log settings, do the following:
1. In the Shell, run the following command:

```
<Get-TransportService | Get-MailboxTransportService | Get-FrontEndTransportService> <ServerIdentity> | Format-List SendConnectorProtocolLog*,ReceiveConnectorProtocolLog*
```

2. Verify the values displayed are the values you configured.

Configure routing table logging

Exchange Server 2013 > Mail flow > Transport logs >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-22*

Routing table logging periodically records a snapshot of the routing table used by Microsoft Exchange Server 2013 to route messages to their destinations.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport service" and "Edge Transport server" entries in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.
- To configure the time interval for automatic recalculation of the routing table, you modify the EdgeTransport.exe.config XML application configuration file. Changes you save to this file are applied after you restart the Microsoft Exchange Transport service. When you restart this service,

mail flow on the server is temporarily interrupted.

- Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to configure routing table logging

Run the following command:

```
Set-TransportService <ServerIdentity> -
RoutingTableLogMaxAge <dd.hh:mm:ss> -
RoutingTableLogMaxDirectorySize <Size>  -
RoutingTableLogPath <LocalFilePath>
```

This example sets the following routing table log settings on the Mailbox server named Mailbox01:

- Sets the location of the routing table log files to D:\Routing Table Log. Note that if the folder doesn't exist, it will be created for you.
- Sets the maximum size of the routing table log folder to 70 MB.
- Sets the maximum age of a routing table log file to 45 days.

```
Set-TransportService Mailbox01 -RoutingTableLogPath "D:
\Routing Table Log" -RoutingTableLogMaxDirectorySize 70MB -
RoutingTableLogMaxAge 45.00:00:00
```

Note:

Setting the *RoutingTableLogMaxAge* parameter to the value `00:00:00` prevents the automatic removal of routing table log files because of their age.

How do you know this worked?

To verify that you have successfully configured routing table logging, do the following:

1. In the Shell, run the following command:

```
Get-TransportService <ServerIdentity> | Format-List
```

```
RoutingTableLog*
```

2. Verify the values displayed are the values you configured.

Use the Command Prompt to configure the interval for automatic recalculation of the routing table in the EdgeTransport.exe.config file

1. In a Command prompt window, open the EdgeTransport.exe.config application configuration file in Notepad by running the following command:

```
Notepad %ExchangeInstallPath%Bin\EdgeTransport.exe.config
```

2. Modify the following key in the `<appSettings>` section.

```
<add key="RoutingConfigReloadInterval" value="<hh:mm:ss>" /
>
```

For example, to change the interval for automatic recalculation of the routing table to 10 hours, use the following value:

```
<add key="RoutingConfigReloadInterval" value="10:00:00" />
```

3. When you are finished, save and close the EdgeTransport.exe.config file.
4. Restart the Microsoft Exchange Transport service by running the following command:

```
net stop MSExchangeTransport && net start
MSExchangeTransport
```

How do you know this worked?

To verify that you have successfully configured the interval for the automatic recalculation of the routing table, verify the routing table log is updated during the time interval you specified.

Note that the routing table will be recalculated and logged earlier than the value specified by the *RoutingConfigReloadInterval* key if any of the following conditions occur:

- A routing configuration change is detected. For example, a Send connector or a Receive connector is added, removed, or modified, or the 6 hour Kerberos token renewal occurs.
- The Microsoft Exchange Transport service is started.

Message tracking

Exchange Server 2013 > Mail flow >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-04*

In Microsoft Exchange Server 2013, the message tracking log is a detailed record of all message activity as messages are transferred to and from the Transport service on Mailbox servers, mailboxes on Mailbox servers, and Edge Transport servers. You can use message tracking logs for message forensics, mail flow analysis, reporting, and troubleshooting.

In Exchange 2013, you can use the **Set-TransportService** cmdlet or the **Set-MailboxServer** cmdlet for all message tracking configuration tasks, because the Exchange 2013 Mailbox server holds the Transport service and the mailboxes. You can use either of these cmdlets to make the following message tracking configuration changes:

- Enable or disable message tracking. The default is enabled.
- Specify the location of the message tracking log files.
- Specify a maximum size for the individual message tracking log files. The default is 10 MB.
- Specify a maximum size for the directory that contains the message tracking log files: The default is 1000 MB.
- Specify maximum age for the message tracking log files: The default is 30 days.
- Enable or disable message subject logging in the message tracking logs. The default is enabled.

Note:

You can also use the Exchange admin center (EAC) to enable or disable message tracking, and to specify the location of the message tracking log files.

By default, Exchange uses circular logging to limit the message tracking logs based on file size and file age to help control the hard disk space used by the message tracking log files.

Contents

Search the message tracking log

Message tracking logs contain vast amounts of data as messages move through an Exchange 2013 Mailbox server. When it comes to searching the message tracking logs, you have different options.

- **Get-MessageTrackingLog** Administrators can use this cmdlet to search the message tracking log for information about messages using a wide range of filter criteria. For more information,

see Search message tracking logs.

- **Delivery reports for administrators** Administrators can use the **Delivery reports** tab in the Exchange admin center (EAC) or the underlying **Search-MessageTrackingReport** and **Get-MesageTrackingReport** cmdlets to search the message tracking logs for information about messages sent by or received by a specific mailbox in the organization. For more information see Delivery reports for administrators.
- **Delivery reports for users** Users can use the **Delivery reports** tab in Outlook Web App to search the message tracking logs for information about messages sent to or sent by their own mailbox. For more information, see Delivery Reports for Users.

Return to top

Structure of the message tracking log files

By default, the message tracking log files exist in %ExchangeInstallPath%TransportRoles\Logs\MessageTracking.

The naming convention for log files in the message tracking log directory is `MSGTRK`*`yyyymmdd-nnnn`*`.log`, `MSGTRKMA`*`yyyymmdd-nnnn`*`.log`, `MSGTRKMD`*`yyyymmdd-nnnn`*`.log`, and `MSGTRKMS`*`yyyymmdd-nnnn`*`.log` . The different logs are used by the following services:

- **MSGTRK** These logs are associated with the Transport service.
- **MSGTRKMA** These logs are associated with the approvals and rejections used by moderated transport. For more information, see Managing message approval.
- **MSGTRKMD** These logs are associated with messages delivered to mailboxes by the Mailbox Transport Delivery service.
- **MSGTRKMS** These logs are associated with messages sent from mailboxes by the Mailbox Transport Submission service.

The placeholders in the log file names represent the following information:

- The placeholder *yyyymmdd* is the coordinated universal time (UTC) date on which the log file was created. *yyyy* = year, *mm* = month, and *dd* = day.
- The placeholder *nnnn* is an instance number that starts at the value of 1 daily for each message tracking log file name prefix.

Information is written to each log file until the file size reaches its maximum specified value for each log file. Then, a new log file that has an incremented instance number is opened. This process is repeated throughout the day. The log file rotation functionality deletes the oldest log files when either of the following conditions is true:

- A log file reaches its maximum specified age.
- The message tracking log directory reaches its maximum specified size.

Important:
The maximum size of the message tracking log directory is calculated as the total size of all log files that have the same name prefix. Other files that do not follow the name prefix convention are not counted in the total directory size calculation. Renaming old log files or copying other

files into the message tracking log directory could cause the directory to exceed its specified maximum size.
On Exchange 2013 Mailbox servers, the maximum size of the message tracking log directory is three times the specified value. Although the message tracking log files that are generated by the four different services have four different name prefixes, the amount and frequency of data written to the **MSGTRKMA** log files is negligible compared to the three other log file prefixes.

The message tracking log files are text files that contain data in the comma-separated value (CSV) format. Each message tracking log file has a header that contains the following information:

- **#Software:** Name of the software that created the message tracking log file. Typically, the value is Microsoft Exchange Server.
- **#Version:** Version number of the software that created the message tracking log file. Currently, the value is 15.0.0.0.
- **#Log-Type:** Log type value, which is Message Tracking Log.
- **#Date:** The UTC date-time when the log file was created. The UTC date-time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where *yyyy* = year, *mm* = month, *dd* = day, T indicates the beginning of the time component, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and Z signifies Zulu, which is another way to denote UTC.
- **#Fields:** Comma-delimited field names used in the message tracking log files.

Return to top

Fields in the message tracking log files

The message tracking log stores each message event on a single line in the log. The message event information is organized by fields, and these fields are separated by commas. The field name is generally descriptive enough to determine the type of information that it contains. However, some fields may be blank, or the type of information that is stored in the field may change based on the message event type and the type of message tracking log file where the event was recorded. General descriptions of the fields that are used to classify each message tracking event are explained in the following table.

Field name	Description
date-time	The UTC date-time of the message tracking event. The UTC date-time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where *yyyy* = year, *mm* = month, *dd* = day, T indicates the beginning of the time component, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and Z signifies Zulu, which is another way to denote UTC.

client-ip	The IPv4 or IPv6 address of the messaging server or messaging client that submitted the message.
client-hostname	The host name or FQDN of the messaging server or messaging client that submitted the message.
server-ip	The IPv4 or IPv6 address of the source or destination Exchange server.
server-hostname	The host name or FQDN of the destination server.
source-context	Extra information associated with the **source** field. For example, transport agent information.
connector-id	The name of the source or destination Send connector or Receive connector. For example, *ServerName**ConnectorName* or *ConnectorName*.
source	The Exchange transport component responsible for the message tracking event. The values found in this field are described in the Source values in the message tracking log section later in this topic.
event-id	The message event type. The event types are described in the Event types in the message tracking log section later in this topic.
internal-message-id	A message identifier assigned by the Exchange server currently processing the message. A specific message's value of **internal-message-id** is different in the message tracking log of every Exchange server that's involved in the transmission of the message. An example

	value is `73014444033`.
message-id	The value of the **Message-Id:** header field found in the message header. If the **Message-Id:** header field does not exist or is blank, an arbitrary value is assigned. This value is constant for the lifetime of the message. For messages created in Exchange, the value is in the format `<GUID@ServerFQDN>`, including the angle brackets (`< >`). For example, `<4867a3d78a50438bad95c0f6d072fca5@mailbox01.contoso.com>`. Other messaging systems may use different syntax or values.
network-message-id	A unique message ID value that persists across copies of the message that may be created due to bifurcation or distribution group expansion. An example value is `1341ac7b13fb42ab4d4408cf7f55890f`.
recipient-address	The email addresses of the message's recipients. Multiple email addresses are separated by the semicolon character (;).
recipient-status	This field contains the recipient status for each recipient separated by the semicolon character (;). The status values are presented for the recipients in the same order as the values in the **recipient-address** field. Example status values include `250 2.1.5 Recipient OK` or `550 4.4.7 QUEUE.Expired;<ErrorText>`.
total-bytes	The size of the message that includes attachments, in bytes.
recipient-count	The number of recipients in the message.

related-recipient-address	This field is used with **EXPAND**, **REDIRECT**, and **RESOLVE** events to display other recipient email addresses associated with the message.
reference	This field contains additional information for specific types of events. For example: **DSN** Contains the report link, which is the **Message-Id** value of the associated delivery status notification (DSN) if a DSN is generated subsequent to this event. If this is a DSN message, the **Reference** field contains the **Message-Id** value of the original message for which this DNS was generated. **EXPAND** The Reference field contains the **related-recipient-address** value of the related messages. **RECEIVE** The Reference field may contain the **Message-Id** value of the related message if the message was generated by other processes, for example, journaling or Inbox rules. **SEND** The Reference field contains the **Internal-Message-Id** value of any DSN messages. **THROTTLE** The Reference field contains the reason why the message was throttled. **TRANSFER** The Reference field contains the Internal-Message-Id of the message that is being forked. For messages generated by inbox rules, the **Reference** field contains the **Internal-Message-Id** value of the inbound message that caused the inbox rule to generate the outbound message.

	For other types of events, the **Reference** field may contain the **Internal-Message-Id** value for forked messages. For other types of events, the **Reference** field is usually blank.
message-subject	The message's subject found in the `Subject:` header field. The tracking of message subjects is controlled by the *MessageTrackingLogSubjectLoggingEnabled* parameter in the **Set-TransportService** or **Set-MailboxServer** cmdlets. By default, message subject tracking is enabled.
sender-address	The email address specified in the `Sender:` header field, or the `From:` header field if `Sender:` is not present.
return-path	The return email address specified by `MAIL FROM:` in the message envelope. Although this field is never empty, it can have the null sender address value represented as `<>`.
message-info	Additional information about the message. For example: • The message origination UTC date-time for **DELIVER** and **SEND** events. The origination date-time is the time when the message first entered the Exchange organization. The UTC date-time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where *yyyy* = year, *mm* = month, *dd* = day, T indicates the beginning of the time component, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and Z

	signifies Zulu, which is another way to denote UTC. • Authentication errors. For example you may see the value `11a` and the type of authentication used when authentication errors occur.
directionality	The direction of the message. Example values include `Incoming`, `Undefined`, and `Originating`.
tenant-id	This field isn't used in on-premises Exchange 2013 organizations.
original-client-ip	The IPv4 or IPv6 address of the original client.
original-server-ip	The IPv4 or IPv6 address of the original server.
custom-data	This field contains data related to a specific event types. For example, the Transport Rule agent uses this field to record the GUID of the transport rule or DLP policy that acted on the message. For more information about these Transport Rule agent values, see the "Data logging" section in the DLP policy detection reports topic,

Return to top

Event types in the message tracking log

Various event types in the **event-id** field are used to classify the message events in the message tracking log. Some message events appear in only one type of message tracking log file, and some message events appear in all types of message tracking log files. The events types that are used to classify each message event are explained in the following table.

Event name	Description
AGENTINFO	This event is used by transport agents to log custom data.
BADMAIL	A message submitted by the Pickup directory or

	the Replay directory that can't be delivered or returned.
DEFER	Message delivery was delayed.
DELIVER	A message was delivered to a local mailbox.
DSN	A delivery status notification (DSN) was generated.
DUPLICATEDELIVER	A duplicate message was delivered to the recipient. Duplication may occur if a recipient is a member of multiple nested distribution groups. Duplicate messages are detected and removed by the information store.
DUPLICATEEXPAND	During the expansion of the distribution group, a duplicate recipient was detected.
DUPLICATEREDIRECT	An alternate recipient for the message was already a recipient.
EXPAND	A distribution group was expanded.
FAIL	Message delivery failed. Sources include **SMTP**, **DNS**, **QUEUE**, and **ROUTING**.
HADISCARD	A shadow message was discarded after the primary copy was delivered to the next hop. For more information, see Shadow redundancy.
HARECEIVE	A shadow message was received by the server in the local database availability group (DAG) or Active Directory site.
HAREDIRECT	A shadow message was created.
HAREDIRECTFAIL	A shadow message failed to be created. The details are stored in the **source-context** field.

INITMESSAGECREATED	A message was sent to a moderated recipient, so the message was sent to the arbitration mailbox for approval. For more information, see Managing message approval.
LOAD	A message was successfully loaded at boot.
MODERATOREXPIRE	A moderator for a moderated recipient never approved or rejected the message, so the message expired. For more information about moderated recipients, see Managing message approval.
MODERATORAPPROVE	A moderator for a moderated recipient approved the message, so the message was delivered to the moderated recipient.
MODERATORREJECT	A moderator for a moderated recipient rejected the message, so the message wasn't delivered to the moderated recipient.
MODERATORSALLNDR	All approval requests sent to all moderators of a moderated recipient were undeliverable, and resulted in non-delivery reports (NDRs).
NOTIFYMAPI	A message was detected in the Outbox of a mailbox on the local server.
NOTIFYSHADOW	A message was detected in the Outbox of a mailbox on the local server, and a shadow copy of the message needs to be created.
POISONMESSAGE	A message was put in the poison message queue or removed from the poison message queue.
PROCESS	The message was successfully processed.
RECEIVE	A message was received by the SMTP receive

	component of the transport service or from the Pickup or Replay directories (source: `SMTP`), or a message was submitted from a mailbox to the Mailbox Transport Submission service (source: `STOREDRIVER`).
REDIRECT	A message was redirected to an alternative recipient after an Active Directory lookup.
RESOLVE	A message's recipients were resolved to a different email address after an Active Directory lookup.
RESUBMIT	A message was automatically resubmitted from Safety Net. For more information, see Safety Net.
RESUBMITDEFER	A message resubmitted from Safety Net was deferred.
RESUBMITFAIL	A message resubmitted from Safety Net failed.
SEND	A message was sent by SMTP between transport services.
SUBMIT	The Mailbox Transport Submission service successfully transmitted the message to the Transport service. For **SUBMIT** events, the **source-context** property contains the following details: • **MDB** The mailbox database GUID. • **Mailbox** The mailbox GUID. • **Event** The event sequence number. • **MessageClass** The type of message. For example, `IPM.Note`. • **CreationTime** Date-time of the message submission. • **ClientType** For example, `User`, `OWA` ,or

	`ActiveSync`.
SUBMITDEFER	The message transmission from the Mailbox Transport Submission service to the Transport service was deferred.
SUBMITFAIL	The message transmission from the Mailbox Transport Submission service to the Transport service failed.
SUPPRESSED	The message transmission was suppressed.
THROTTLE	The message was throttled.
TRANSFER	Recipients were moved to a forked message because of content conversion, message recipient limits, or agents. Sources include **ROUTING** or **QUEUE**.

Return to top

Source values in the message tracking log

The values in the **source** field in the message tracking log indicate the transport component that's responsible for the message tracking event. The following table describes the values of the **source** field.

Source value	Description
ADMIN	The event source was human intervention. For example, an administrator used Queue Viewer to delete a message, or submitted message files using the Replay directory.
AGENT	The event source was a transport agent.
APPROVAL	The event source was the approval framework that's used with moderated recipients. For more information, see Managing message approval.

DNS	The event source was DNS.
DSN	The event source was a delivery status notification (DSN). For example, a non-delivery report (NDR).
GATEWAY	The event source was a Foreign connector. For more information, see Foreign connectors.
MAILBOXRULE	The event source was an Inbox rule. For more information, see Inbox rules.
ORAR	The event source was an Originator Requested Alternate Recipient (ORAR). You can enable or disable support for ORAR on Receive connectors using the *OrarEnabled* parameter on the **New-ReceiveConnector** or **Set-ReceiveConnector** cmdlets.
PICKUP	The event source was the Pickup directory. For more information, see Pickup directory and Replay directory.
POISONMESSAGE	The event source was the poison message identifier. For more information about poison messages and the poison message queue, see Queues
PUBLICFOLDER	The event source was a mail-enabled public folder.
QUEUE	The event source was a queue.
REDUNDANCY	The event source was Shadow Redundancy. For more information, see Shadow redundancy.
ROUTING	The event source was the routing resolution component of the categorizer in the Transport

	service.
SAFETYNET	The event source was Safety Net. For more information, see Safety Net.
SMTP	The message was submitted by the SMTP send or SMTP receive component of the transport service.
STOREDRIVER	The event source was a MAPI submission from a mailbox on the local server.

Return to top

Example entries in the message tracking log

An uneventful message sent between two users generates several entries in the message tracking log. You can see the results using the **Get-MessageTrackingLog** cmdlet. For more information, see Search message tracking logs.

This is a condensed example of the message tracking log entries created when the user chris@contoso.com successfully sends a test message to the user michelle@contoso.com. Both users have mailboxes on the same server.

```
EventId    Source      Sender            Recipients
     MessageSubject
-------    ------      ------            ----------
     --------------
NOTIFY     STOREDRIVER                   {}
RECEIVE    STOREDRIVER chris@contoso.com
{michelle@contoso.com} test
SUBMIT     STOREDRIVER chris@contoso.com
{michelle@contoso.com} test
HAREDIRECT SMTP        chris@contoso.com
{michelle@contoso.com} test
RECEIVE    SMTP        chris@contoso.com
{michelle@contoso.com} test
AGENT      AGENT       chris@contoso.com
{michelle@contoso.com} test
SEND       SMTP        chris@contoso.com
```

```
{michelle@contoso.com} test
DELIVER    STOREDRIVER chris@contoso.com
{michelle@contoso.com} test
```

Return to top

Security concerns for the message tracking log

No message content is stored in the message tracking log. By default, the subject line of an email message is stored in the message tracking log. You may want to disable message subject logging to comply with increased security or privacy requirements. Before you enable or disable message subject logging, make sure that you verify your organization's policy about revealing subject line information. For more information, see Configure message tracking.

Return to top

Configure message tracking

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-18*

Message tracking records the SMTP transport activity of all messages transferred to and from the Transport service or mailboxes on a Microsoft Exchange Server 2013 Mailbox server. You can use message tracking logs for message forensics, mail flow analysis, reporting, and troubleshooting.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport service" entries in the Mail flow permissions topic or the "Mailbox server configuration" entry in the Recipients Permissions topic.
- You can use the Exchange admin center (EAC) to enable or disable message tracking, or set the message tracking log path. For all other message tracking options, you need to use the Exchange Management Shell.
- On an Exchange 2013 Mailbox server, you can use either the **Set-TransportService** or the **Set-MailboxServer** cmdlet to configure the message tracking options. The procedures in this topic use the **Set-TransportService** cmdlet.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the EAC to configure message tracking on Mailbox servers

1. In the EAC, navigate to **Servers** > **Servers**.
2. Select the Mailbox server you want to configure, and then click **Edit** .
3. On the server properties page, click **Transport Logs**.
4. In the **Message tracking log** section, change any of the following:
   - **Enable message tracking log** To disable message tracking on the server, clear the check box. To enable message tracking on the server, select the check box.
   - **Message tracking log path** The value you specify must be on the local Exchange server. If the folder doesn't exist, it will be created for you when you click **Save**.
5. Click **Save**.

Use the Shell to configure message tracking

To configure message tracking, run the following command:

```
Set-TransportService <ServerIdentity> -
MessageTrackingLogEnabled <$true | $false> -
MessageTrackingLogMaxAge <dd.hh:mm:ss> -
MessageTrackingLogMaxDirectorySize <Size> -
MessageTrackingLogMaxFileSize <Size> -
MessageTrackingLogPath <LocalFilePath> -
MessageTrackingLogSubjectLoggingEnabled <$true|$false>
```

This example sets the following message tracking log settings on the Mailbox server named Mailbox01:

Sets the location of the message tracking log files to D:\Message Tracking Log. Note that if the folder doesn't exist, it will be created for you.

Sets the maximum size of a message tracking log file to 20 MB.

Sets the maximum size of the message tracking log directory to 1.5 GB.

Sets the maximum age of a message tracking log file to 45 days.

```
Set-TransportService Mailbox01 -MessageTrackingLogPath "D:
\Hub Message Tracking Log" -MessageTrackingLogMaxFileSize
20MB -MessageTrackingLogMaxDirectorySize 1.5GB -
MessageTrackingLogMaxAge 45.00:00:00
```

Note:

- Setting the *MessageTrackingLogPath* parameter to the value `$null`, effectively disables message tracking. However, if the value of the *MessageTrackingLogEnabled* parameter is `$true`, event log errors are generated.
- Setting the *MessageTrackingLogMaxAge* parameter to the value `00:00:00` prevents the automatic removal of message tracking log files because of their age.
- On Exchange 2013 Mailbox servers, the maximum size of the message tracking log directory is three times the value of the *MessageTrackingLogMaxDirectorySize* parameter. Although the message tracking log files that are generated by the four different services have four different name prefixes, the amount and frequency of data written to the **MSGTRKMA** log files is negligible compared to the three other log file prefixes. For more information, see the "Structure of the message tracking log files" section in the Message tracking topic.

This example disables message subject logging in the message tracking log on the Mailbox server named Mailbox01:

```
Set-TransportService Mailbox01 -
MessageTrackingLogSubjectLoggingEnabled $false
```

This example disables message tracking on the Mailbox server named Mailbox01:

```
Set-TransportService Mailbox01 -MessageTrackingLogEnabled
$false
```

How do you know this worked?

To verify that you have successfully configured message tracking, do the following:
1. In the Shell, run the following command:

```
Get-TransportService <ServerIdentity> | Format-List
MessageTrackingLog*
```

2. Verify that the values displayed are the values you configured.

Search message tracking logs

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-25*

In Microsoft Exchange Server 2013, the message tracking log is a detailed record of all message activity as messages are transferred to and from the Transport service on Mailbox servers, mailboxes on Mailbox servers, and Edge Transport servers.

You can use the **Get-MessageTrackingLog** cmdlet in the Exchange Management Shell to search for entries in the message tracking log by using specific search criteria.

What do you need to know before you begin?

- Estimated time to complete: 30 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Message tracking" entry in the Mail flow permissions topic.
- Searching the message tracking logs requires the Microsoft Exchange Transport Log Search service to be running. If you disable or stop this service, you can't search the message tracking logs or run delivery reports. However, stopping this service does not affect other features in Exchange.
- The field names displayed by the results from the **Get-MessageTrackingLog** cmdlet are similar to the actual field names used in the message tracking logs. The biggest differences are:
  - The dashes are removed from the field names. For example **internal-message-id** is displayed as `InternalMessageId`.
  - The **date-time** field is displayed as `Timestamp`.
  - The **recipient-address** field is displayed as `Recipients`.
  - The **sender-address** field is displayed as `Sender`.
- The **date-time** field in the message tracking log stores information in Coordinated Universal Time (UTC). However, you should enter your date-time search criteria for the *Start* or *End* parameters in the regional date-time format of the computer that you are using to perform the search.
- You can't copy the message tracking log files from another Exchange server and then search them by using the **Get-MessageTrackingLog** cmdlet. Also, if you manually save an existing message tracking log file, the change in the file's date-time stamp breaks the query logic that Exchange uses to search the message tracking logs.
- The Exchange 2013 **Get-MessageTrackingLog** cmdlet is able to search the message tracking logs on Exchange 2007 and Exchange 2010 servers in the same Active Directory site.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see

Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to search the message tracking logs

To search the message tracking log entries for specific events, use the following syntax.

```
Get-MessageTrackingLog [-Server <ServerIdentity.] [-
ResultSize <Integer> | Unlimited] [-Start <DateTime>] [-End
<DateTime>] [-EventId <EventId>] [-InternalMessageId
<InternalMessageId>] [-MessageId <MessageId>] [-
MessageSubject <Subject>] [-Recipients
<RecipientAddress1,RecipientAddress2...>] [-Reference
<Reference>] [-Sender <SenderAddress>]
```

To view the 1000 most recent message tracking log entries on the server, run the following command:

```
Get-MessageTrackingLog
```

This example searches the message tracking logs on the local server for all entries from 3/28/2013 8:00 AM to 3/28/2013 5:00 PM for all **FAIL** events where the message sender was pat@contoso.com.

```
Get-MessageTrackingLog -ResultSize Unlimited -Start
"3/28/2013 8:00AM" -End "3/28/2013 5:00PM" -EventId "Fail"
-Sender "pat@contoso.com"
```

Use the Shell to control the output of a message tracking log search

Use the following syntax.

```
Get-MessageTrackingLog <SearchFilters> | <Format-Table |
Format-List> [<FieldNames>] [<OutputFileOptions>]
```

This example searches the message tracking logs using the following search criteria:

- Return results for the first 1,000 **Send** events.
- Display the results in the list format.
- Display only those field names that begin with `Send` or `Recipient`.
- Write the output to a new file named `D:\Send Search.txt`

```
Get-MessageTrackingLog -EventId Send | Format-List
Send*,Recipient* > "D:\Send Search.txt"
```

Use the Shell to search the message tracking logs for message entries on multiple servers

Typically, the value in the **MessageID:** header field remains constant as the message travels throughout the Exchange organization. This property is named **InternetMessageId** in queue viewing utilities, and **MessageId** in the message tracking log viewing utilities. After you have determined the `MessageID:` value of a specific message, you can search for information about that message in the message tracking logs on every Mailbox server in your Exchange organization.

To search all message tracking log entries for a specific message across all Mailbox servers, use the following syntax.

```
Get-ExchangeServer | where {$_.isHubTransportServer -eq
$true -or $_.isMailboxServer -eq $true} | Get-
MessageTrackingLog -MessageId <MessageID> | Select-Object
<CommaSeparatedFieldNames> | Sort-Object -Property
<FieldName>
```

This example searches the message tracking logs on all Exchange 2013 Mailbox servers using the following search criteria:

- Find any entries related to a message that has a **MessageID:** value of `<ba18339e-8151-4ff3-aeea-87ccf5fc9796@mailbox01.contoso.com>`. Note that you can omit the angle bracket characters (< >). If you don't, you need to enclose the entire **MessageID:** value in quotation marks.
- For each entry, display the fields **date-time**, **server-hostname**, **client-hostname**, **source**, **event-id**, and **recipient-address**.
- Sort the results by the **date-time** field.

```
Get-ExchangeServer | where {$_.isHubTransportServer -eq
$true -or $_.isMailboxServer -eq $true} | Get-
MessageTrackingLog -MessageId ba18339e-8151-4ff3-aeea-
87ccf5fc9796@mailbox01.contoso.com | Select-Object
Timestamp,ServerHostname,ClientHostname,Source,EventId,Reci
pients | Sort-Object -Property Timestamp
```

Use the EAC to search the message tracking logs

You can use the Delivery Reports for administrators feature in the Exchange admin center (EAC) to search the message tracking logs for information about messages sent by or received by a specific mailbox in your organization. For more information see Track messages with delivery reports.

Delivery reports for administrators

Exchange Server 2013 > Mail flow >

Topic Last Modified: *2013-02-18*

With delivery reports for administrators, you can track delivery information about messages sent by or received from any specific mailbox in your organization. Specifically, delivery reports for administrators uses the Exchange admin center (EAC) to perform a targeted search of the message tracking logs. The search is always scoped to a specific mailbox. You can search for messages sent by the mailbox, or sent to the mailbox, and you can filter the search results by the message subject.

The content of the message body isn't returned in a delivery report, but the subject line is displayed in the results. If you want to search the mailboxes in your organization for specific email messages based on message content, see In-Place eDiscovery.

You may find delivery report searches useful in the following situations:

- A manager gives a poor review for a trainee because the trainee didn't turn in an assignment on time. The trainee insists he sent a message with the assignment attached. The manager asks you to verify the status of the message.
- A security bulletin has been sent to users asking that they reply immediately, but no one has replied. Are they ignoring the message or did they just not receive it?
- Users complain that no one is receiving their messages. They check delivery status for their mail but can't figure out what is going on. This may be because a rule is being applied to messages at the organization level.

After you create a delivery report search, the resulting delivery report will show the following information: Who the message was sent from and to, the subject line, and when the message was sent. The delivery report also shows message delivery status and reasons why delivery may be delayed or failed.

More about delivery reports

- Here's how to create a new delivery report: Track messages with delivery reports.
- On-premises Exchange organizations can use the Exchange Management Shell to query the message tracking logs directly. For more information, see Search message tracking logs.

- Users can track their own messages. For more information, see Delivery Reports for Users.
- If your organization contains a previous version of Exchange, you need to consider how the delivery reports feature in Exchange 2013 works across Exchange versions.
  - Exchange 2013 delivery reports can track messages across Exchange 2010 servers in the same Active Directory site.
  - Exchange 2013 delivery reports can't track messages across Exchange 2007 servers in the same Active Directory site. The delivery reports feature uses a remote procedure call and a web service interface that doesn't exist in Exchange 2007.

Track messages with delivery reports

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-13*

Delivery Reports is a message tracking tool in the Exchange Administration Center (EAC) that you can use to search for delivery status on email messages sent to or from users in your organization's address book, with a certain subject. You can track delivery information about messages sent by or received from any specific mailbox in your organization. The content of the message body isn't returned in a delivery report, but the subject line is displayed in the results. You can track messages for up to 14 days after they were sent or received.

Note:

Delivery Reports tracks messages sent by people using the Microsoft Outlook or Outlook Web App email clients. It doesn't track messages sent from POP or IMAP email clients, such as Windows Mail, Outlook Express, or Mozilla Thunderbird.

What do you need to know before you begin?

- Estimated time to complete each procedure: Time to complete will vary based on the scope of your search.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Message tracking" entry in the Mail flow permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.
- Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to track messages

1. In the EAC, navigate to **Mail Flow** > **Delivery Reports**.
2. Enter the following information:
   - *** Mailbox to search:** Click **Browse** to select the mailbox from the address book and then click **OK**. Selecting the mailbox to search is required.
   - Select one of the following:
     - **Search for messages sent to** Use this option to search for messages sent to specific users. Click **Select users** and then pick users from the address book by selecting a user from the list and clicking **Add**. You can select more than one user here. When you're finished selecting users, click **OK** to return to the **Delivery Reports** page. If you select this option, you can also leave the field blank to find messages sent to anyone.
     - **Search for messages received from** Use this option to search for messages received from a specific user. Again, just select the user from the address book and click **OK** to return to the **Delivery Reports** page. If you select this option, you have to specify a sender.
   - **Search for these words in the subject line** Enter subject line information here, or leave it blank to expand your search.
3. When you are finished, click **Search**. If you want to start over, click **Clear**.

Use the EAC to review a delivery report

To view delivery information, select a message in the **Search results** pane and click **Details**.

The delivery report shows delivery status and detailed delivery information for the message you have selected from the **Search results** pane. At the top of the report, you'll see the following fields:

- **Subject** The subject line of the message appears as the heading of the report.
- **From** Alias, display name, or email address of the person who sent the message.
- **To** Alias, display name, or email address for each recipient of the message.
- **Sent** Date and time the message was sent.

Summary to date section

This section appears in the delivery report if a message was sent to more than one person or recipient. The top of this section tells you the total number of recipients that the message was sent to and gives brief delivery information for each recipient.

- **Summary to date** Displays total number of recipients, and if there are messages Pending, Delivered, or Unsuccessful. Click the hyperlinks to sort by status.
- **Search box** The search box is useful if you sent the message to a group of more than 30 recipients. In the search box, type an email address that you want to get delivery information about and click the magnifying glass.
- **To** Shows the email address of the recipient.

- **Status** This column displays the status of the message for each recipient.

Detailed report information

This section contains detailed delivery information for a message sent to the recipient you select in the Summary to date section.

- **Delivery Report for** The email address of the selected recipient is shown here.
- **Submitted** Date and time that the message was submitted for delivery by the system.

Depending on the delivery status of the message, you may see a variety of status states, including:

- **Delivered** Indicates successful delivery.
- **Deferred** Indicates that a message is delayed.
- **Pending** If message delivery is pending because a message meets the criteria for an organization-wide rule or policy or because it's subject to message approval, the status message explains what action a rule is performing or that the message must be approved by a moderator before delivery.
- **Moderator** The status indicates whether the message was approved or rejected by the moderator.
- **Groups Expanded** If a message was sent to a group, the individual users are shown in the **Summary to date** section so you can see the delivery status for each recipient. If you need to remove or add a user to a group during a delivery report investigation, you can modify a group by clicking **Edit Groups**.
- **Failed** Shows the date, time, and reason for a message delivery failure. For example, an organization-wide rule may be blocking message delivery or the message couldn't be delivered.

When you're done reviewing the report, click **Close**. Delivery reports aren't saved, but you can re-run a report at any time. Remember there is a two-week search window.

How do you know this worked?

If your search was successful, messages that fit the search criteria are listed in the **Search results** pane. To view the delivery information for a specific message, select it and then click **Details**. If no messages are displayed in the **Search results** pane, change the search criteria and then re-run the search.

Content conversion

Exchange Server 2013 > Mail flow >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

Content conversion is the process of correctly formatting a message for each recipient. The decision to perform content conversion on a message depends on the destination and format of the message being processed. In Microsoft Exchange Server 2013, there are two different kinds of content conversion:

- **Message conversion for external recipients** This type of content conversion includes the Transport Neutral Encapsulation Format (TNEF) conversion options and message encoding options for external recipients. Messages sent to recipients inside the Exchange organization don't require this type of content conversion. This type of content conversion is handled by the categorizer in the Transport service on Mailbox server. Categorization on each message happens after a newly arrived message is put in the Submission queue. In addition to recipient resolution and routing resolution, content conversion is performed on the message before the message is put in a delivery queue. If a single message contains multiple recipients, the categorizer determines the appropriate encoding for each message recipient. Content conversion tracing doesn't capture any content conversion failures that the categorizer encounters as it converts messages sent to external recipients.
- **MAPI conversion for internal recipients** This type of content conversion is handled by the Mailbox Transport service. The Mailbox Transport service exists on Mailbox servers to transmit messages between mailbox databases on the local server, and the Transport service on Mailbox servers. Specifically, the Mailbox Transport Submission service transmits messages from the sender's Outbox to the Transport service on a Mailbox server. The Mailbox Transport Delivery service transmits messages from the Transport service on a Mailbox server to the recipient's Inbox. The Mailbox Transport Submission service converts all outgoing messages from MAPI and the Mailbox Transport Delivery service converts all incoming messages to MAPI. Content conversion tracing captures these MAPI conversion failures. For more information, see Content conversion tracing.

This topic explains the message conversion options for external recipients.

Contents

Exchange and Outlook message formats

The following list describes the basic message formats available in Exchange and Microsoft Outlook:

- **Plain text** A plain text message uses only US-ASCII text as described in RFC 2822. The message can't contain different fonts or other text formatting. The following two formats can be used for a plain text message:
  - The message headers and the message body are composed of US-ASCII text. Attachments must be encoded by using *Uuencode*. Uuencode represents Unix-to-Unix encoding and defines an

encoding algorithm to store binary attachments in the body of an email message by using US-ASCII text characters.
    - The message is MIME-encoded with a Content-Type value of text/plain, and a Content-Transfer-Encoding value of 7bit for the text parts of a multipart message. Any message attachments are encoded by using Quoted-printable or Base64 encoding. By default, when you compose and send a plain text message in Outlook, the message is MIME-encoded with a Content-Type value of text/plain.
- **HTML** An HTML message supports text formatting, background images, tables, bullet points, and other graphical elements. By definition, an HTML-formatted message must be MIME-encoded to preserve these formatting elements.
- **Rich text format (RTF)** RTF supports text formatting and other graphical elements. RTF is synonymous with TNEF. TNEF and RTF can be used interchangeably. The rich text message format is completely different from the rich text document format available in Microsoft Word.

Only Outlook and a few other MAPI email clients understand RTF messages.

- **TNEF** The Transport Neutral Encapsulation Format is a Microsoft-specific format for encapsulating MAPI message properties. A TNEF message contains a plain text version of the message and an attachment that packages the original formatted version of the message. Typically, this attachment is named Winmail.dat. The Winmail.dat attachment includes the following information:
    - Original formatted version of the message, including, for example, fonts, text sizes, and text colors
    - OLE objects, including, for example, embedded pictures or embedded Microsoft Office documents
    - Special Outlook features, including, for example, custom forms, voting buttons, or meeting requests
    - Regular message attachments that were in the original message

The resulting plain text message can be represented in the following formats:

- RFC 2822-compliant message composed of only US-ASCII text with a Winmail.dat attachment encoded in Uuencode
- Multipart MIME-encoded message that has a Winmail.dat attachment

A MAPI-compliant email client that fully understands TNEF, such as Outlook, processes the Winmail.dat attachment and displays the original message content without ever displaying the Winmail.dat attachment. An email client that doesn't understand TNEF may present a TNEF message in any of the following ways:

- The plain text version of the message is displayed, and the message contains an attachment named Winmail.dat, Win.dat, or some other generic name such as Att*nnnnn*.dat or Att*nnnnn*.eml where the *nnnnn* placeholder represents a random number.
- The plain text version of the message is displayed. The TNEF attachment is ignored or removed. The result is a plain text message.
- Messaging servers that understand TNEF can be configured to remove TNEF attachments from incoming messages. The result is a plain text message. Moreover, some email clients such as Microsoft Outlook Express may not understand TNEF, but recognize and ignore TNEF

attachments. The result is a plain text message.

There are third-party utilities that can help convert Winmail.dat attachments.

TNEF is understood by all versions of Exchange since Exchange Server version 5.5.

- **Summary Transport Neutral Encapsulation Format (STNEF)** STNEF is equivalent to TNEF. However, STNEF messages are encoded differently than TNEF messages. Specifically, STNEF messages are always MIME-encoded and always have a Content-Transfer-Encoding value of Binary. Therefore, there's no plain text representation of the message, and there's no distinct Winmail.dat attachment contained in the body of the message. The whole message is represented by using only binary data. Messages that have a Content-Transfer-Encoding value of Binary can only be transferred between SMTP messaging servers that support and advertise the BINARYMIME and CHUNKING SMTP extensions as defined in RFC 3030. The messages are always transferred between SMTP messaging by using the BDAT command, instead of the standard DATA command.

STNEF is understood by all versions of Exchange since Exchange 2000. STNEF is automatically used for all messages transferred between Exchange servers in the organization since native mode Exchange Server 2003.

Exchange never sends STNEF messages to external recipients. Only TNEF messages can be sent to recipients outside the Exchange organization.

Return to top

Content conversion options for external recipients

The content conversion options that you can set in an Exchange organization for external recipients can be described in the following categories:

- **TNEF conversion options** These conversion options specify whether TNEF should be preserved or removed from messages that leave the Exchange organization.
- **Message encoding options** These options specify message encoding options, such as MIME and non-MIME character sets, message encoding, and attachment formats.

These conversion and encoding options are independent of one another. For example, whether TNEF messages can leave the Exchange organization isn't related to the MIME encoding settings or plain text encoding settings of those messages.

You can specify the content conversion at various levels of the Exchange organization as described in the following list:

- **Remote domain settings** Remote domains define the settings for outgoing message transfers between the Exchange organization and external domains.. Even if you don't create remote domain entries for specific domains, there's a predefined remote domain named Default that applies to all remote address spaces (*).
- **Mail user and mail contact settings** Mail users and mail contacts are similar because both have external email addresses and contain information about people outside the Exchange organization. The main difference is mail users have accounts that can be used to log on to the

Active Directory domain and access resources in the organization.

- **Outlook settings** In Outlook, you can set the message formatting and encoding options described in the following list:
  - **Message format** You can set the default message format for all messages. You can override the default message format as you compose a specific message.
  - **Internet message format** You can control whether TNEF messages are sent to remote recipients or whether they are first converted to a more compatible format. You can also specify various message encoding options for messages sent to remote recipients. These settings don't apply to messages sent to recipients in the Exchange organization.
  - **Internet recipient message format** You can control whether TNEF messages are sent to specific recipients or whether they are first converted to a more compatible format. You can set the conversion options for specific contacts in your Contacts folder, and you can override the conversion options for a specific recipient in the To, Cc, or Bcc fields as you compose a message. These conversion options aren't available for recipients in the Exchange organization.
  - **Internet recipient message encoding options** You can control the MIME or plain text encoding options for specific contacts in your Contacts folder, and you can override the conversion options for a specific recipient in the To, Cc, or Bcc fields as you compose a message. These conversion options aren't available for recipients in the Exchange organization.
  - **International options** You can control the character sets used in messages.

TNEF conversion options

You can specify the TNEF conversion options at the following levels:

- Remote domain settings
- Mail user and mail contact settings
- Outlook settings, including:
  - Message format
  - Internet message format
  - Internet recipient message format

Message encoding options

You can specify the message encoding options at the following levels:

- Remote domain settings
- Mail user and mail contact settings
- Outlook settings, including:
  - Message format
  - Internet message
  - Internet recipient message format
  - Message character set encoding options

For detailed information, see Message encoding options.

Return to top

Understanding the structure of email messages

To better understand the content conversion options for external recipients, you need to understand the structure of email messages. An SMTP message is based on plain 7-bit US-ASCII text to compose and send email messages. A standard SMTP message consists of the following elements:

- **Message envelope** The message envelope is defined in RFC 2821. The message envelope contains information required to transmit and deliver the message. Recipients never see the message envelope, because it's generated by the message transmission process and isn't actually part of the message contents.
- **Message contents** The message contents are defined in RFC 2822. The message contents consist of the following elements:
  - **Message header** The message header is a collection of header fields. Header fields consist of a field name, followed by a colon (:) character, followed by a field body, and ended by a carriage return/line feed (CR/LF) character combination.

A field name must be composed of printable US-ASCII text characters except the colon (:) character. Specifically, ASCII characters that have values from 33 through 57 and 59 through 126 are permitted.

A field body may be composed of any US-ASCII characters, except for the carriage return (CR) character and the line feed (LF) character. However, a field body may contain the CR/LF character combination when used in *header folding*. Header folding is the separation of a single header field body into multiple lines as described in section 2.2.3 of RFC 2822. Other field body syntax requirements are described in sections 3 and 4 of RFC 2822.

  - **Message body** The message body is a collection of lines of US-ASCII text characters that appears after the message header. The message header and the message body are separated by a blank line that ends with the CR/LF character combination. The message body is optional. Any line of text in the message body must be less than 998 characters. The CR and LF characters can only appear together to indicate the end of a line.

When SMTP messages contain elements that aren't plain US-ASCII text, the message must be encoded to preserve those elements. The MIME standard defines a method of encoding content in messages that isn't text. MIME allows for text in other character sets, attachments without text, multipart message bodies, and header fields in other character sets. MIME is defined in RFC 2045, RFC 2046, RFC 2047, RFC 2048, and RFC 2077. MIME defines a collection of header fields that specifies additional message attributes. The following table describes some important MIME header fields.

Important MIME header fields

Header field name	Default value	Description
MIME-Version	1.0	This header field is the first MIME header field that appears in a

		MIME-formatted message. This header field appears after the other standard RFC 2822 header fields, but before any other MIME header fields. MIME-aware email clients use this header field to identify a MIME-encoded message. When this header field is absent, MIME-aware email clients identify the message as plain text.
Content-Type	text/plain	This header field identifies the media type of the message content as described in RFC 2046. A media type consists of a type, a subtype, and one or more optional parameters, such as a *charset*= parameter that defines the MIME character encoding. Types that begin with "x-" aren't standard. Subtypes that begin with "vnd." are vendor-specific. The Internet Assigned Numbers Authority (IANA) maintains a list of registered media types. For more information, see MIME Media Types. The *multipart* media type allows for multiple message parts in the same message by using sections defined by different media types. Some Content-Type field values include text/plain, text/html,

		multipart/mixed, and multipart/alternative.
Content-Transfer-Encoding	7bit	This header field can describe the following information about a message: • The encoding algorithm used to transform any non-US-ASCII text or binary data that exists in the message body. • An indicator that describes the current condition of the message body. There can be multiple values of the Content-Transfer-Encoding header field in a MIME message. When the Content-Transfer-Encoding header field appears in the message header, it applies to the whole body of the message. When the Content-Transfer-Encoding header field appears in one of the parts of a multipart message, it applies only to that part of the message. When an encoding algorithm is applied to the message body data, the message body data is transformed into plain US-ASCII text. This transformation allows the message to travel through older SMTP messaging servers that only support messages in US-ASCII text. The values of the Content-

		Transfer-Encoding header field that indicate an encoding algorithm was used on the message body are as follows: • **Quoted-printable** This encoding algorithm uses printable US-ASCII characters to encode the message body data. If the original message text is mostly US-ASCII text, Quoted-printable encoding gives somewhat readable and compact results. All printable US-ASCII text characters except the equal sign (=) character can be represented without encoding. • **Base64** This encoding algorithm is based primarily on the privacy-enhanced mail (PEM) standard defined in RFC 1421. Base64 encoding uses the 64-character alphabet encoding algorithm and output padding characters defined by PEM to encode the message body data. A Base64 encoded message is typically 33 percent larger than the original message. Base64 encoding creates a predictable increase in message size and is optimal for binary data and non-US-ASCII text. Typically, you won't see multiple

		encoding algorithms used in the same message. When no encoding algorithm has been used on the message body, the Content-Transfer-Encoding header field merely identifies the current condition of the message body data. The following values of the Content-Transfer-Encoding header field indicate that no encoding algorithms were used on the message body: • **7bit** This value indicates that the message body data is already in the RFC 2822 format. Specifically, this means that the following conditions must be true: ○ All lines of text must be less than 998 characters long. ○ All characters must be US-ASCII text that have character values from 1 through 127. ○ The CR and LF characters can only be used together to indicate the end of a line of text. The whole message body may be 7bit, or part of the message body in a multipart message may be 7bit. If the multipart message contains other parts that have any binary data or non-US-ASCII text, that part of the message must be encoded using the Quoted-

		printable or Base64 encoding algorithms. Messages that have 7bit bodies can travel between SMTP messaging servers by using the standard DATA command. • **8bit** This value indicates that the message body data contains non-US-ASCII characters. Specifically, this means that the following conditions must be true: ○ All lines of text must be less than 998 characters long. ○ One or more characters in the message body have values larger than 127. ○ The CR and LF characters can only be used together to indicate the end of a line of text. The whole message body may be 8bit, or part of the message body in a multipart message may be 8bit. If the multipart message contains other parts that have binary data, that part of the message must be encoded using the Quoted-printable or Base64 encoding algorithms. Messages that have 8bit bodies can only travel between SMTP messaging servers that support the 8BITMIME SMTP extension as defined in RFC 1652, such as servers running Exchange 2000 Server or newer versions.

		Specifically, this means that the following conditions must be true: ○ The 8BITMIME keyword must be advertised in the server's EHLO response. ○ Messages are still transferred by using the SMTP standard DATA command. However, the BODY=8BITMIME parameter must be added to the end of the MAIL FROM command. • **Binary** This value indicates that the message body contains non-US-ASCII text or binary data. Specifically, this means that the following conditions are true: ○ Any sequence of characters is allowed. ○ There is no line length limitation. ○ Binary message elements don't require encoding. Messages that have Binary bodies can only travel between SMTP messaging servers that support the BINARYMIME SMTP extension as defined in RFC 3030, such as servers running Exchange 2000 Server or newer versions. Specifically, this means that the following conditions must be true: ○ The BINARYMIME keyword must be advertised in the server's EHLO response. ○ The BINARYMIME SMTP extension can only be used with the CHUNKING SMTP extension. *Chunking* enables

		large message bodies to be sent in multiple, smaller chunks. Chunking is also defined in RFC 3030. The CHUNKING keyword must also be advertised in the server's EHLO response. ○ Messages are transferred using the BDAT command instead of the standard DATA command. ○ The *BODY=BINARYMIME* parameter must be added to the end of the MAIL FROM command when the message has a message body. The values 7bit, 8bit, and Binary never exist together in the same multipart message. The values are mutually exclusive. The Quoted-printable or Base64 values may appear in a 7bit or 8bit multipart message body, but never in a Binary message body. If a multipart message body contains different parts composed of 7bit and 8bit content, the whole message is classified as 8bit. If a multipart message body contains different parts composed of 7bit, 8bit, and Binary content, the whole message is classified as Binary.
Content-Disposition	Attachment	This header field instructs a MIME-enabled email client on how it should display an attached file, and

		is described in RFC 2183. The values of this field may be Inline or Attachment. When the value of this field is Inline, the attachment is displayed in the message body. When the value of this field is Attachment, the attached file appears as a regular attachment separate from the message body. Other parameters are available when the value is Attachment, such as Filename, Creation-date, and Size.

Return to top

Configure content transfer encoding

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-03*

Content transfer encoding defines encoding methods for transforming binary email message data into the US-ASCII plain text format. This transformation allows the message to travel through older SMTP messaging servers that only support messages in US-ASCII text. Content transfer encoding is defined in RFC 2045. The transfer encoding method is stored in the **Content-Transfer-Encoding** header field in the message. In Microsoft Exchange Server 2013, the following content transfer encoding methods are available:

- **7-bit** This value indicates that the message body data is already in the US ASCII plain text format, and no message encoding has been done to the message.
- **Quoted-printable (QP)** This encoding method uses printable US-ASCII characters to encode the message body data. If the original message text is mostly US-ASCII text, QP encoding gives somewhat readable and compact results. By default, Exchange 2013 uses QP for encoding binary message data.
- **Base64** This encoding method is based primarily on the privacy-enhanced mail (PEM) standard defined in RFC 1421. Base64 encoding uses the 64-character alphabet encoding method and

output padding characters defined by PEM to encode the message body data. Base64 encoding creates a predictable increase in message size and is optimal for binary data and non-US-ASCII text.

You configure the transfer encoding method using the *ByteEncoderTypeFor7BitCharsets* parameter on the **Set-OrganizationConfig** and **Set-RemoteDomain** cmdlets. The content transfer encoding settings you configure with **Set-OrganizationConfig** apply to all messages in the Exchange organization. The content transfer encoding settings you configure with **Set-RemoteDomain** apply only to message sent to external recipients in the remote domain.

The following table lists the values that you can use to set the transfer encoding method.

Parameter in Set-OrganizationConfig	Parameter in Set-RemoteDomain	Description
0	`Use7Bit`	Always use 7-bit encoding for HTML and for plain text. This is the default value.
1	`UseQP`	Always use QP encoding for HTML and for plain text.
2	`UseBase64`	Always use Base64 encoding for HTML and for plain text.
5	`UseQPHtmlDetectTextPlain`	Use QP encoding for HTML and for plain text unless line wrapping is enabled in plain text. If line wrapping is enabled, use 7-bit encoding for plain text.
6	`UseBase64HtmlDetectTextPlain`	Use Base64 encoding for HTML and for plain text, unless line wrapping is enabled in plain text. If line wrapping is enabled in plain text, use Base64 encoding for HTML, and use 7-bit encoding for plain text.
13	`UseQPHtml7BitTextPlain`	Always use QP encoding for

		HTML. Always use 7-bit encoding for plain text.
14	`UseBase64Html7BitTextPlain`	Always use Base64 encoding for HTML. Always use 7-bit encoding for plain text.

For more details about **Content-Transfer-Encoding** header field, see the "Understanding the structure of email messages" section in Content conversion.

For more information about remote domains, see Remote domains.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport service" entry in the Mail flow permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to configure the content transfer encoding method for the organization

To configure the content transfer encoding method for the organization, run the following command:

```
Set-OrganizationConfig -ByteEncoderTypeFor7BitCharsets <Integer>
```

For example, to set the content transfer encoding method to Base64, run the following command:

```
Set-OrganizationConfig -ByteEncoderTypeFor7BitCharsets 2
```

Use the Shell to configure the content transfer encoding

method for a remote domain

To configure the content transfer encoding method for all the recipients in a remote domain, run the following command:

```
Set-RemoteDomain -ByteEncoderTypeFor7BitCharsets <Value>
```

For example, to set the content transfer encoding method to Base64, run the following command:

```
Set- RemoteDomain -ByteEncoderTypeFor7BitCharsets UseBase64
```

How do you know this worked?

To verify that you have successfully configured the method for content transfer encoding, do the following:

1. Send a test message that contains a mixture of US-ASCII text and binary data or non-US-ASCII text to an internal or external test account. Use an internal account to test organization settings, and an external account in the remote domain to test remote domain settings.
2. In an email client, view the **Content-Transfer-Encoding** header field in the message, and verify the content transfer encoding method that was used on the message matches the method you configured.

Message encoding options

Exchange Server 2013 > Mail flow > Content conversion >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-28*

The message encoding options that are available in Exchange specify message characteristics, such as MIME and non-MIME character sets, binary encoding, and attachment formats. You can specify message encoding options in the following locations:

- Remote domain settings
- Mail user and mail contact settings
- Microsoft Outlook settings
  - Message format
  - Internet message format
  - Internet recipient message format
  - Message character set encoding options

Content

Message encoding options for messages sent to remote domains

When you configure message encoding options for a remote domain, the specific settings are applied for all messages sent to that domain. For remote domains in your organization, you have the following configuration options for message encoding.

Setting	Available in EAC in Exchange Online Dedicated	Available in the Shell
MIME character set The character set that you specify will only be used for MIME messages that don't have their own character set specified. Setting this parameter won't overwrite character sets that are already specified in the outgoing mail. **Non-MIME character set** This setting is used if either of the following conditions are true: • Incoming messages from a remote domain are missing the value of the *charset*= setting in the MIME Content-Type: header field. • Outgoing messages to a remote domain are missing the value of	Yes	Yes

the MIME character set.		
Content type You can specify the content type for MIME messages sent to the recipients in the remote domain. You can use of the following settings: • **MimeHtmlText** All messages are converted to MIME messages that use HTML formatting, unless the original message is a text message. If the original message is a text message, the outgoing message will be a MIME message that uses text formatting. This is the default setting. • **MimeText** All messages are converted to MIME messages that use text formatting. • **MimeHtml** All messages are converted to MIME messages that use HTML formatting.	No	Yes
Line wrap size You can specify the maximum number of characters that can exist on a single line of text in the body of the e-mail message. Older email client applications may prefer 78 characters per line. This option is only available by using the Shell.	No	Yes

Return to top

Message encoding options for mail users and mail contacts

When you configure message encoding options for a mail contact or a mail user, that option is applied to all messages sent to that specific recipient. For mail contacts and mail users in your organization, you have the following configuration options for message encoding:

- **UsePreferMessageFormat** This parameter specifies whether the message format settings configured for the mail contact override the global settings configured for the remote domain. If you disable this setting, Exchange ignores other message encoding options for this recipient and the message encoding is determined by the configuration of the remote domain or the settings configured by the message sender.
- **MessageFormat** This parameter specifies the message format. You can either specify Text or Mime as the message format. The value of this setting is dependent on the *MessageBodyFormat* parameter. If the message body format is Html or TextAndHtml, you must set this parameter to Mime.
- **MessageBodyFormat** This parameter specifies the message body format. You can specify Text, Html, or TextAndHtml. The value of this setting is dependent on the *MessageFormat* parameter. If the message format is Text, you must also set this parameter to Text.
- **MacAttachmentFormat** This parameter specifies the Apple Macintosh operating system attachment format for messages. You can specify BinHex, UuEncode, AppleSingle, or AppleDouble. The value of this setting is dependent on the *MessageFormat* parameter. If the message format is set to Text, you must set this parameter to either BinHex or UuEncode. If the message format is set to Mime, you must set this parameter to BinHex, AppleSingle or AppleDouble.

You need to use these parameters in the Exchange Management Shell to set the message encoding options for mail users and mail contacts. For more information, see the following topics:

- Enable-MailContact
- New-MailContact
- Set-MailContact
- Enable-MailUser
- New-MailUser
- Set-MailUser

Return to top

Message encoding options available in Outlook

As a sender, you can specify message encoding options in Outlook at any of the following stages:

- By configuring the default message format to be either plain text or HTML.
- By setting the message format as you're composing it to either plain text or HTML using the **Format** area in the **Options** tab.
- By configuring the message encoding options for messages that are sent to all recipients outside

the Exchange organization. These options are called *Internet message format* options. The options only apply to remote recipients, and not to recipients in the Exchange organization.

- By configuring the message encoding options for messages that are sent to specific recipients outside the Exchange organization. These options are called *Internet recipient message format* options. The options only apply to remote recipients in your Contacts folder, and not to recipients in the Exchange organization.

By default, Outlook uses automatic character set message encoding by scanning the whole text of the outgoing message to determine the appropriate encoding to use for the message. This setting applies to messages that you send to Internet recipients and recipients in the Exchange organization. However, you can bypass this and specify a preferred encoding for outgoing messages.

Return to top

Message encoding options available in Outlook Web App

As a sender, you can specify message encoding options in Outlook Web App at any of the following stages:

- By configuring the default message format to be either plain text or HTMLin the **Message format** section of the **Settings** >**Options** > **Settings** page.
- By setting the message format as you're composing it to either plain text or HTML by using the **More options** (...) menu, and selecting **Switch to plain text** or **Switch to HTML**.

Return to top

Order of precedence for message encoding options

Exchange uses the order of precedence as described in the following list to determine the message encoding options for outgoing messages sent to recipients outside the Exchange organization:

1. Remote domain settings
2. Outlook or Outlook Web App settings
3. Mail user or mail contact settings

The list specifies the order of precedence from lowest to highest. A setting made at a higher level may override a setting made at a lower level.

The following table describes the order of precedence from lowest priority to highest priority for message character set encoding options.

Order of precedence from lowest priority to highest priority for message character set encoding options

Source	Parameter	Values
Remote domain entry setting,	*CharacterSet*	Specified

using the EAC or **Set-RemoteDomain**		
Remote domain entry setting, using the EAC or **Set-RemoteDomain**	*NonMimeCharacterSet*	Specified
Outlook setting	Message character set encoding	• Auto-select • Specified

When you set the non-MIME character set for a remote domain, the character set is assigned to the following types of messages:

- Outgoing messages to a configured remote domain that don't contain a specified character set.
- Incoming messages from a configured remote domain that don't contain a specified character set.

The value of the Windows ANSI code page for the transport server is used to assign a character set to the following types of messages:

- Internal messages that don't contain a specified character set.
- Internal messages that contain a specified character set, but don't contain a specified server code page.

If a message contains a specified but invalid character set, the transport server tries to replace the invalid character set with a valid character set.

The following table describes the order of precedence from lowest priority to highest priority for plain text message encoding options.

Order of precedence from lowest priority to highest priority for plain text message encoding options

Source	Parameter	Values
Set-RemoteDomain	*LineWrapSize*	• From 0 through 132 • unlimited
Outlook settings	Message format	Plain text
Outlook settings	Internet message format	Plain text options: • Encode attachments in UuEncode format when you send a plain text message • Automatically wrap text at *nn* characters

Outlook settings	Internet recipient message format	Plain text format: • Encode attachments in UuEncode attachment format • Use BinHex Mac attachment format
Set-MailUser **Set-MailContact**	*UsePreferMessageFormat*	• `$true` • `$false` If the value is `$false` or if the recipient isn't defined as a mail user or mail contact in the Exchange organization, the mail user or mail contact settings are ignored.
Set-MailUser **Set-MailContact**	*MessageFormat*	Text
Set-MailUser **Set-MailContact**	*MessageBodyFormat*	Text
Set-MailUser **Set-MailContact**	*MacAttachmentFormat*	• BinHex • UuEncode

The following table describes the order of precedence from lowest priority to highest priority for MIME message encoding options.

Order of precedence from lowest priority to highest priority for MIME message encoding options

Source	Parameter	Values
Set-RemoteDomain	*ContentType*	• `MimeHtmlText` • `MimeText` • `MimeHtml`
Outlook or Outlook Web App settings	Message format	• Plain text • HTML
Outlook settings	Internet recipient message format	MIME message format • Plain text • Include both plain text and

		HTML • HTML

Set-MailUser
Set-MailContact
	UsePreferMessageFormat	`$true`
		`$false`
		If the value is `$false` or if the recipient isn't defined as a mail user or mail contact in the Exchange organization, the mail user or mail contact settings are ignored.

Set-MailUser **Set-MailContact**	*MessageFormat*	• Text • Mime
Set-MailUser **Set-MailContact**	*MessageBodyFormat*	• `Text` • `Html` • `TextAndHtml`
Set-MailUser **Set-MailContact**	*MacAttachmentFormat*	• BinHex • AppleSingle • AppleDouble

Return to top

For more information

Message encoding options

TNEF conversion options

Remote domains

Remote domains in Exchange Online

Manage mail users

Manage mail contacts

Change the message format in Outlook

TNEF conversion options

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-28*

You can specify whether Transport Neutral Encapsulation Format (TNEF) should be preserved or removed from messages that leave the Exchange organization. TNEF, also known as Outlook Rich Text Format or Exchange Rich Text Format, is a Microsoft-specific format for encapsulating MAPI message properties. All versions of Microsoft Outlook fully understand TNEF. Outlook Web App translates TNEF into MAPI and displays the formatted messages. However, other email clients that don't understand TNEF typically display TNEF formatted messages as plain text messages with Winmail.dat or Win.dat attachments.

Contents

TNEF conversion options for remote domains

When you configure TNEF conversion options for a remote domain, those TNEF conversion options are applied to all messages sent to that domain.

- For Exchange Online Dedicated, you use the Exchange admin center (EAC) to set TNEF conversion options for a remote domain at **Mail flow** > **Remote domains** > **Edit** (✎) > **Use Exchange rich-text format**.
- For Exchange Online and Exchange 2013, you use the *TnefEnabled* parameter on the **Set-RemoteDomain** cmdlet to set TNEF conversion options for a remote domain.

For remote domains in your organization, you have the following configuration options for TNEF conversion:

Setting	In the EAC	In the Shell
Use TNEF for all messages sent to the remote domain.	**Always**	`$true`
Never use TNEF for any messages sent to the remote domain.	**Never**	`$false`
TNEF messages aren't specifically allowed or	**Follow user settings**	`$null` (blank)

prevented for recipients in the remote domain. Whether TNEF messages are sent to recipients in the remote domain depends on the specific setting on the mail contact or mail user, or the setting specified by the sender in Outlook. This is the default value.		

For more information about remote domains, see Remote domains or **Remote domains in Exchange Online**.

Return to top

TNEF conversion options for mail users and mail contacts

When you configure TNEF conversion options for a mail contact or a mail user, those TNEF conversion options are applied to all messages sent to that specific recipient. You use the *UseMapiRichTextFormat* parameter on the **Set-MailUser** and **Set-MailContact** cmdlets to configure the TNEF conversion options for mail users and mail contacts.

For mail users and mail contacts in your organization, you have the following configuration options for TNEF conversion:

- **Always** TNEF is used for all messages sent to the recipient. The corresponding value for the *UseMapiRichTextFormat* parameter is `Always`.
- **Never** TNEF is never used for any messages sent to the recipient. The corresponding value for the *UseMapiRichTextFormat* parameter is `Never`.
- **Use default settings** TNEF messages aren't specifically allowed or prevented for the mail user or mail contact. Whether TNEF messages are sent to the recipient depends on the specific setting for the corresponding remote domain or the setting specified by the sender in Outlook. The corresponding value for the *UseMapiRichTextFormat* parameter is `UseDefaultSettings`. This is the default setting.

Return to top

TNEF conversion options in Outlook

Senders can control the default TNEF message conversion options for TNEF messages sent to all recipients outside the Exchange organization. These options are called *Internet message format* options. The options only apply to remote recipients, and not to recipients in the Exchange

organization.

Note:

The following options define how messages containing Outlook rich text are handled when sent to external recipients. If the message format you're using is HTML or plain text, these settings don't apply.

You have the following TNEF conversion options in Outlook:

- **Convert to HTML format** This is the default option. Any TNEF messages sent to remote recipients are converted to HTML. Any formatting in the message should closely resemble the original message. MIME-encoded HTML messages are supported by many, but not all, email clients.
- **Convert to Plain Text format** Any TNEF messages sent to remote recipients are converted to plain text. Any formatting in the message is lost.
- **Send using Outlook Rich Text Format** Any TNEF messages sent to remote recipients remain TNEF messages.

You can configure these options in the following locations in Outlook:

- **Outlook 2010 or Outlook 2013** **File** > **Options** > **Mail** > **Message format**.
- **Outlook 2007** **Tools** > **Options** > **Mail Format** > **Internet Format**.

Senders can also control the default TNEF message conversion options for TNEF messages sent to specific recipients outside the Exchange organization. These options are called *Internet recipient message format* options. The options only apply to remote recipients stored in your Contacts folder, and not to recipients in the Exchange organization. You have the following TNEF conversion options for remote recipients in your Contacts folder:

- **Let Outlook decide the best sending format** This is the default setting. This setting forces Outlook to use the TNEF conversion option that's specified by the default Internet format. The possible values are **Convert to HTML format**, **Convert to Plain Text format**, or **Send using Outlook Rich Text Format**. Therefore, the TNEF message may be left as TNEF, converted to HTML, or converted to plain text. If you want to make sure that the TNEF message remains TNEF for this recipient, you should change this setting from **Let Outlook decide the best sending format** to **Send using Outlook Rich Text format**.
- **Send Plain Text only** Any TNEF messages sent to the recipient are converted to plain text. Any formatting in the message is lost.
- **Send using Outlook Rich Text format** Any TNEF messages sent to remote recipients remain TNEF messages.

You can configure these options for a contact in the following locations in Outlook:

- **Outlook 2010 or Outlook 2013** Open the contact card, double-click the email address, click the **View more options for interacting with this person** icon, and select **Outlook properties**. In the **E-mail Properties** dialog box, select **Internet format**.
- **Outlook 2007** Open the contact card, double-click the **E-mail** field and select **Internet format**.

Return to top

Order of precedence for TNEF conversion options

Exchange uses the order of precedence as described in the following list to determine the TNEF conversion options for outgoing messages sent to recipients outside the Exchange organization:

1. Remote domain settings
2. Mail user or mail contact settings
3. Outlook settings

The list specifies the order of precedence from highest to lowest. The TNEF setting on the remote domain overrides the TNEF settings on the mail user, mail contact or in Outlook. For example, suppose you send a Rich Text message in Outlook, but the recipient is in a domain where the remote domain setting specifically doesn't allow TNEF-formatted messages. The message received by the recipient will be plain text or HTML, but not TNEF.

Also, Exchange never sends Summary Transport Neutral Encoding Format (STNEF) messages to external recipients. Only TNEF messages can be sent to recipients outside the Exchange organization.

Return to top

Content conversion tracing

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-25*

Content conversion tracing captures failures in the MAPI content conversion that's performed by the Mailbox Transport service on inbound and outbound messages on a Microsoft Exchange Server 2013 Mailbox server.

The Mailbox Transport service on a Mailbox server is responsible for the content conversion of messages sent to and from mailbox recipients. Specifically, the Mailbox Transport Submission service converts outbound messages from mailbox users from MAPI to MIME. The Mailbox Transport Delivery service converts inbound messages for mailbox users from MIME to MAPI. Content conversion tracing is responsible for capturing these MAPI conversion failures.

The categorizer in the Transport service on a Mailbox server is responsible for the content conversion of all messages sent to external recipients. Content conversion tracing doesn't capture any content conversion failures that the categorizer in the Transport service encounters as it converts messages sent to external recipients.

Contents

Configure content conversion tracing

Content conversion tracing is controlled by the following parameters in the **Set-TransportService** and **Set-MailboxTransportService** cmdlets in the Exchange Management Shell:

- *ContentConversionTracingEnabled* This parameter enables or disables content conversion tracing in the Transport service on the Mailbox server, or in the Mailbox Transport service on the Mailbox server. Valid values for this parameter are `$true` and `$false`. The default value is `$false`. If your Exchange organization contains multiple Mailbox servers, you must enable content conversion tracing on each Mailbox server.
- *PipelineTracingPath* Although this parameter is associated with pipeline tracing, it also specifies the root location of the content conversion tracing files. The default location in the Transport service is `%ExchangeInstallPath%TransportRoles\Logs\Hub\PipelineTracing`. The default location in the Mailbox Transport service is `%ExchangeInstallPath%TransportRoles\Logs\Mailbox\PipelineTracing`.The path must be local to the Exchange computer.

Content conversion creates a folder named `ContentConversionTracing` in the path specified by the *PipelineTracingPath* parameter. In the `ContentConversionTracing` folder, content conversion creates two subfolders: `InboundFailures` and `OutboundFailures`. The `InboundFailures` folder contains the information from inbound message content conversion failures. The `OutboundFailures` folder contains the information from outbound message content conversion failures.

The maximum size for all the files in the `InboundFailures` folder or the `OutboundFailures` folder is 128 megabytes (MB). Content conversion tracing doesn't use circular logging to remove old files based on the age or size of the files. As soon as the maximum size for a folder is reached, content conversion tracing stops writing information to the folder. If you want to make sure that the maximum folder size limits aren't exceeded, you can create a scheduled task that periodically moves the content conversion tracing files to a different location.

The permissions required on the folders and subfolders used in content conversion tracing are as follows:

- Administrators: Full Control
- Network Service: Full Control
- System: Full Control

Caution:
Content conversion tracing copies the complete contents of email messages. To avoid the unwanted disclosure of confidential information, you need to set appropriate security permissions on the location of the content conversion tracing files.

Return to top

How content conversion tracing works

When the content conversion of an inbound message fails, a delivery status notification (DSN) that has the status code 5.6.0 is sent to the message sender. If content conversion tracing is enabled, the failure information is recorded at the time that the 5.6.0 DSN message is generated. Each content conversion error generates two separate files.

A content conversion error that occurs when an inbound message is converted from MIME to MAPI generates the following two files in the InboundFailures folder:

- **<GUID>.eml** This file contains the failed message in text format.
- **<GUID>.txt** This file contains the exception description, conversion results, conversion options, and message size limits imposed on all messages by the Mailbox Transport service.

A content conversion error that occurs when an outbound message is converted from MAPI to MIME generates the following two files in the OutboundFailures folder:

- **<GUID>.msg** This file contains the failed message in the Microsoft Outlook message format.
- **<GUID>.txt** This file contains the exception description, conversion results, conversion options, and message size limits imposed on all messages by the store driver.

The placeholder *<GUID>* is the same in both file names. Each content conversion error generates a different GUID that's used in the file names of the corresponding message and text files. An example of a GUID that's used in the file names is `038b930e-61fd-4bfd-b9b4-0374c18b73f7`.

Return to top

Considerations for content conversion tracing

You can leave content conversion tracing enabled for proactive monitoring. Or, you can enable content conversion tracing to troubleshoot a specific failure event. You can usually reproduce inbound content conversion failures by asking the recipient of the 5.6.0 DSN message to resend the original message.

Inbound content conversion failures are the most common. Some of the reasons for inbound content conversion errors include the following:

- **Violations of message size limits** These message size limits are imposed by the Mailbox Transport service to help prevent denial of service attacks (DoS). These message limits are listed in the *<GUID>*.txt file. These message limits include the following:
  - **MaxMimeTextHeaderLength** This limit specifies the maximum number of text characters that can be used in a MIME header. The value is 2000.
  - **MaxMimeSubjectLength** This limit specifies the maximum number of text characters that can be used in the subject line. The value is 255.
  - **MSize** This limit specifies the maximum message size. The value is 2147483647 bytes.
  - **MaxMimeRecipients** This limit specifies the total number of recipients allowed in the To, Cc, and Bcc fields. The value is 12288.

- **MaxRecipientPropertyLength** This limit specifies the maximum number of text characters that can be used in a recipient description. The value is 1000.
- **MaxBodyPartsTotal** This limit specifies the maximum number of message parts that can be used in a MIME multipart message. The value is 250.
- **MaxEmbeddedMessageDepth** This limit specifies the maximum number of forwarded messages that can exist in a message. The value is 30.

For more information about configurable message size limits used in the Transport service on Mailbox servers or on Edge Transport servers, see Message size limits.

- **Failure to convert an inbound iCalendar message to a meeting request** RFC 2445 defines iCalendar as a standard for calendar data exchange. Specific causes of the conversion failure include the following:
  - Incorrect use of iCalendar by the sending agent.
  - Constructs of iCalendar that can't be supported by the Outlook or Exchange calendar schema.

Conversion failures of iCalendar don't result in the sender receiving a 5.6.0 DSN message. Instead, the message is delivered with an attached .ics file that contains the iCalendar message body.

- **Failures caused by badly formatted MIME messages** Unsolicited commercial email or spam messages may have formatting errors in the message header, such as unmatched quotation marks in recipient descriptions. A much smaller number of failures caused by badly formatted MIME messages are considered bugs.

Outbound content conversion failures are much less common than inbound failures. When outbound failures occur, they are usually caused by Exchange code bugs or corrupted message content.

Return to top

DSNs and NDRs

Exchange Server 2013 > Mail flow >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-15*

This topic describes how to read and interpret non-delivery report (NDR) delivery status notification (DSN) messages in Microsoft Exchange Server 2013.

Contents

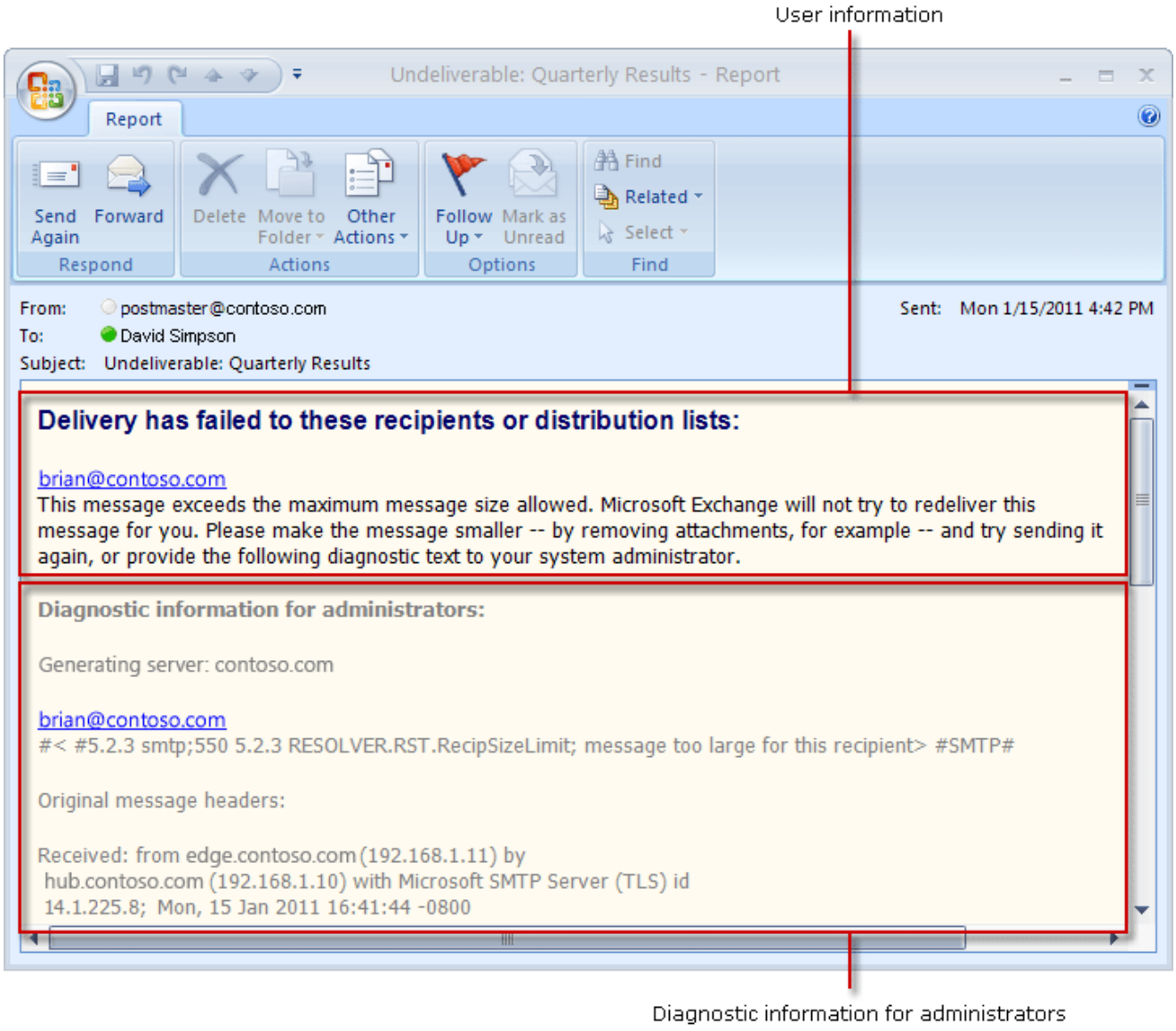
NDR sections

In Exchange 2013, NDRs are designed to be easy to read and understand by both end-users and administrators. Information that is displayed in an NDR is separated into the following two areas:

- A user information section
- A **Diagnostic information for administrators** section

The information in each section is targeted to the readers of that section. The user information section appears first and contains feedback to help the user understand in nontechnical terms why the delivery of the message failed. The **Diagnostic information for administrators** section provides deeper technical information such as the original message headers, to help email administrators troubleshoot a delivery issue that may exist. The following figure shows the user information section and **Diagnostic information for administrators** section of an NDR.

NDR Sections



User information section

The user information section of an NDR generated by Exchange contains information that you want to communicate to an end-user who has sent a message that is later returned with an NDR. The text that is displayed in this section is inserted by the Exchange server that generated the NDR.

The text in the user information section is designed to help end-users determine why the message was rejected and how to resend the message successfully if the message should be resent. When applicable, the fully qualified domain name (FQDN) of the server that rejected the message is included in the user information section. If delivery fails to more than one recipient, the email address of each recipient is listed and the reason for the failure is included in the space below the recipient's email address.

You can modify the text in the user information section by using the **New-SystemMessage** cmdlet. By creating a custom message, you can provide specific information to end-users, such as a telephone number to use to contact the helpdesk department or a hyperlink to use to obtain self-service support.

Return to top

Diagnostic information for administrators

The **Diagnostic information for administrators** section contains more detailed information about the specific error that occurred during delivery of the message, the server that generated the NDR, and the server that rejected the message. The following fields are present in most NDRs and are visible in the "NDR sections" figure earlier in this topic:

- **Generating server** The generating server is the SMTP server that created the NDR. The generating server takes the enhanced status code that is explained later in this topic. This code creates an easy-to-read NDR. If no remote server is listed below the email address of the sender in the **Diagnostic information for administrators** section, the generating server is also the server that rejected the original email message. If message delivery fails when the message is sent to another recipient in the Exchange organization, the same server typically rejects the original message and generates the NDR.
- **Rejected recipient** The rejected recipient is the email address of the recipient to which delivery of the original message failed. If delivery to more than one recipient has failed, the email address for each recipient is listed. The rejected recipient field also contains the following sub-fields for each email address listed:
  - **Remote server** The remote server field contains the FQDN of the server that rejects delivery of the message during the SMTP conversation. The remote server field is only populated when delivery has been attempted to a remote server, and that delivery attempt has been rejected before the receiving server successfully acknowledges the message after the message body is sent. If the original message is successfully acknowledged by the receiving server and is then rejected because of content restrictions, for example, the remote server field is not populated.
  - **Enhanced status code** The enhanced status code is the code returned by the server that rejected the original message. The enhanced status code indicates why the original message was rejected. The enhanced status code is not rewritten by Exchange but is used to determine

what text to display in the user information section. The enhanced status codes you're most likely to encounter are listed in "Common Enhanced Status Codes" later in this topic. For a detailed list of enhanced status codes, see RFC 3463.

  - **SMTP response** The SMTP response is the machine readable text returned by the server that rejected the original message. The SMTP response typically contains a short string that provides an explanation of the enhanced status code that is also returned. The SMTP response is not rewritten by Exchange. Additionally, this response is always presented in US-ASCII format.

- **Original message headers** The original message headers section contains the message headers of the rejected message. These headers can provide useful diagnostic information, such as information that can help you determine the path that the message took before it was rejected or whether the **To** field matches the email address that is specified in the rejected recipient field.

Return to top

Examples of NDR messages

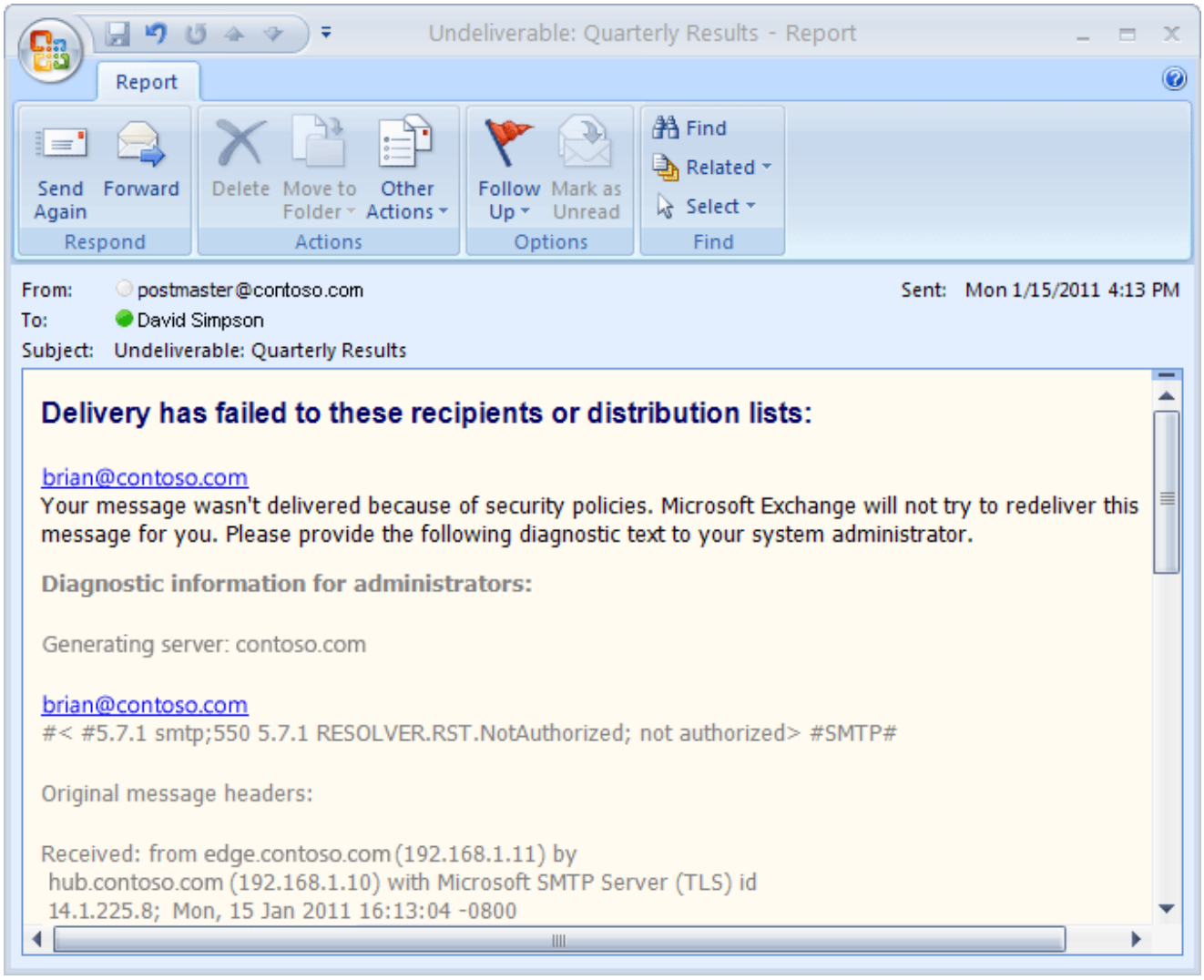
The following sections provide examples of two ways that NDR messages can be generated:

- By the same server
- By different servers

NDR generated and original message rejected by the same server

The following example shows what happens when a remote email organization accepts delivery of an email message through an Edge Transport server, and then rejects that message because of a policy restriction on the recipient's mailbox. In this case, the sender is not allowed to send messages to the recipient. Edge Transport servers do not perform message size validation so the Edge Transport server in this example accepts the message because it has a valid recipient address and the message does not violate another content restrictions. Because the remote email organization accepts the whole message, including the message contents, the remote email organization is responsible for rejecting the message and for generating the NDR message to be sent to the sender.

NDR generated and message rejected by the same server



Also, messages that are rejected when they are sent to recipients that are part of the same Exchange organization are typically rejected by the same email server that generates the NDR message. Messages sent to local recipients can be rejected for various reasons, such as mailboxes that have exceeded their quota, lack of authorization to send messages to the recipient address, or hardware failures that result in an extended loss of connectivity to other servers in the organization.

In both situations, no remote server is included under the email address of the recipients listed in the NDR message.

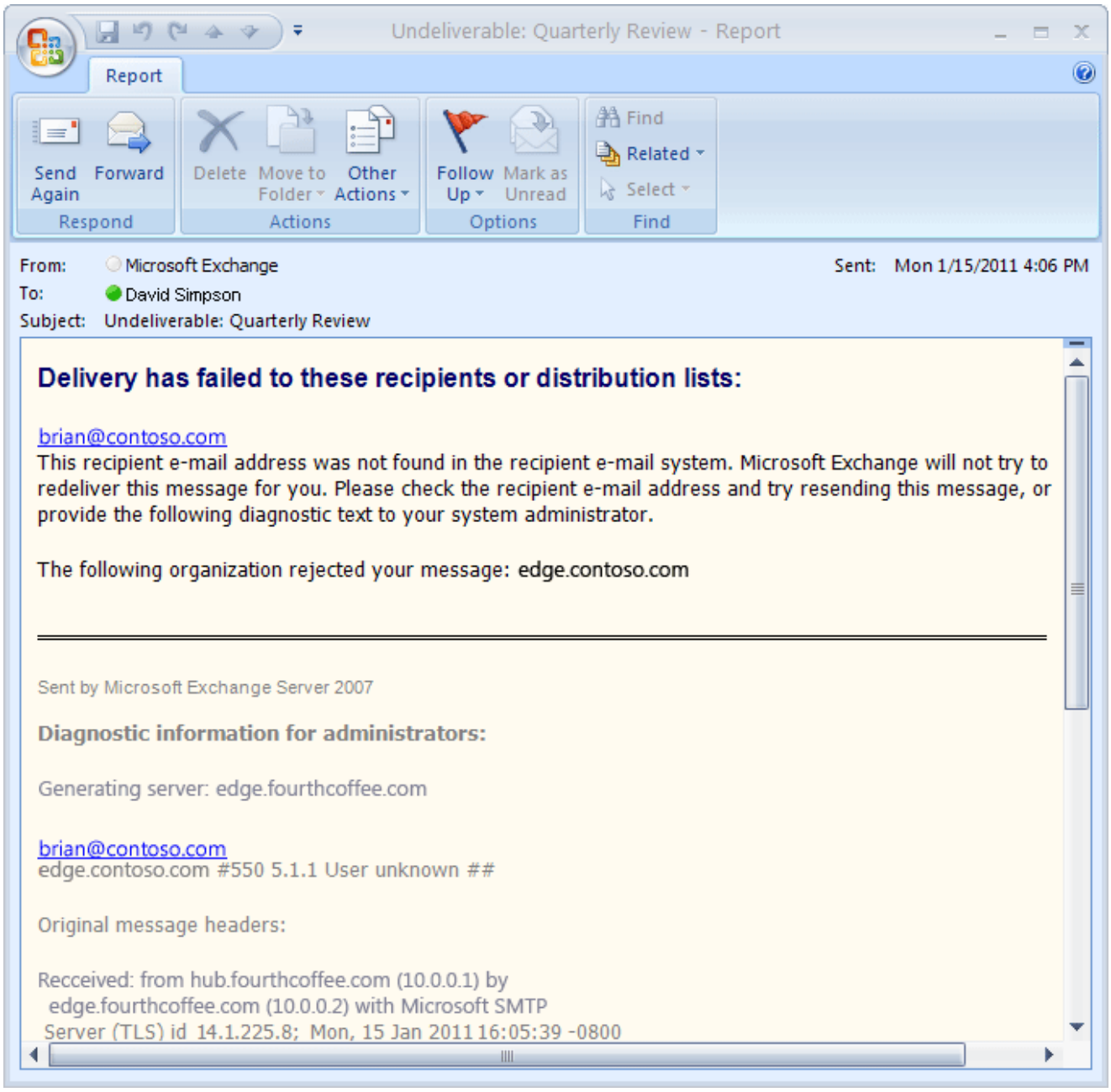
Return to top

NDR generated and original message rejected by different servers

The following example shows what happens when a remote email organization rejects delivery of an email message before it ever accepts the message. In this example, the remote server rejects the message and returns an enhanced status code to the local sending server because the specified recipient does not exist. The rejection happens before the receiving server ever acknowledges the message. Because the receiving server doesn't successfully acknowledge the message, the receiving server is not responsible for the message. Therefore, the local sending server generates the NDR

message and sends it to the sender of the original message.

NDR generated and message rejected by different servers



Return to top

Common enhanced status codes

The following table contains a list of the enhanced status codes that are returned in NDRs for the most common message delivery failures.

Enhanced status code	Description	Possible cause	Additional information
4.3.1	`Insufficient system resources`	An out-of-memory error occurred. A resource problem, such as a full	Ensure that your Exchange server has

		disk, can cause this problem. Instead of getting a disk full error, you might be getting an out-of-memory error.	enough disk storage.
4.3.2	`System not accepting network messages`	This NDR is generated when a queue has been frozen.	You can resolve this condition by unfreezing the queue.
4.4.1	`Connection timed out`	The destination server is not responding. Transient network conditions can cause this error. The Exchange server tries automatically to connect to the server again and deliver the mail. If delivery fails after multiple attempts, an NDR with a permanent failure code is generated.	Monitor the situation. This may be a transient problem that may correct itself.
4.4.2	`Connection dropped`	A connection dropped between the servers. Transient network conditions or a server that is experiencing problems can cause this error. The sending server will retry delivery of the message for a specific time period, and then	Monitor the situation as the server retries delivery. This may be a transient problem that may correct itself. This situation can also occur when the message size limit for the connection is reached, or if the message

		generate further status reports.	submission rate for the client IP address has exceeded the configured limit.
4.4.7	`Message expired`	The message in the queue has expired. The sending server tried to relay or deliver the message, but the action was not completed before the message expiration time occurred. This message can also indicate that a message header limit has been reached on a remote server, or some other protocol time-out occurred while communicating with the remote server.	This message usually indicates an issue on the receiving server. Check the validity of the recipient address, and determine if the receiving server is configured correctly to receive messages. You may have to reduce the number of recipients in the message header for the host about which you are receiving this error. If you send the message again, it is placed in the queue again. If the receiving server is available, the message is delivered.
5.0.0	`HELO / EHLO requires domain address`	This situation is a permanent failure. Possible causes include: • There is no route for the given address space; for example, an SMTP connector is configured, but this	Some potential resolutions include: • On one or more SMTP connectors, add an asterisk (*****) value as the SMTP address space. • Verify that DNS is working.

		address does not match. • DNS returned an authoritative host that was not found for the domain. • An SMTP error occurred.	
5.1.0	`Sender denied`	This NDR is caused by a general failure (bad address failure). An email address or another attribute could not be found in Active Directory. Contact entries without the **targetAddress** attribute set can cause this problem. Another possible cause could be that the **homeMDB** attribute of a user could not be determined. The **homeMDB** attribute corresponds to the Exchange server on which the user's mailbox resides. Another common cause of this NDR is if you used Microsoft Outlook to save your email message as a file, and then someone opened the	Either the recipient address is incorrectly formatted, or the recipient could not be correctly resolved. The first step in resolving this error is to check the recipient address and send the message again.

		message offline and replied to the message. The message property only preserves the **legacyExchangeDN** attribute when Outlook delivers the message, and therefore the lookup could fail.	
5.1.1	`Bad destination mailbox address`	This failure may be caused by the following conditions: • The recipient email address was entered incorrectly by the sender. • No recipient exists in the destination email system. • The recipient mailbox has been moved and the Outlook recipient cache on the sender's computer has not updated. • An invalid legacy domain name (DN) exists for the recipient mailbox Active Directory.	This error typically occurs when the sender of the message incorrectly enters the email address of the recipient. The sender should check the recipient's email address and send again. This error can also occur if the recipient email address was correct in the past but has changed or has been removed from the destination email system. If the sender of the message is in the same Exchange organization as the recipient, and the recipient mailbox still exists, determine whether the recipient mailbox has been relocated to a new email server. If this is the

			case, Outlook may not have updated the recipient cache correctly. Instruct the sender to remove the recipient address from sender's Outlook recipient cache and then create a new message. Resending the original message will result in the same failure. Other issues may cause this error, such as an invalid legacy distinguished name (DN) in Active Directory. Examine and correct the legacy DN of the recipient's mailbox. Then instruct the sender to remove the recipient address from sender's Outlook recipient cache and then create a new message. Resending the original message will result in the same failure.
5.1.2	`Invalid X.400 address`	The recipient has a non-SMTP address that can't be matched to a destination. The address does not appear to be	Verify that the recipient's address was entered correctly. If the recipient's address is in a non-SMTP email system that you

		local, and there are no connectors configured with address spaces that contain the recipient's address.	specifically want to provide mail delivery to, you will need to add the appropriate type of connector to your topology and configure it to provide service to the recipient's email system.
5.1.3	`Invalid recipient address`	This message indicates that the recipient address appears incorrectly on the message.	Either the recipient address is formatted incorrectly, or the recipient could not be correctly resolved. The first step in resolving this error is to check the recipient address and send the message again. Also, examine the SMTP recipient policy and ensure that each mail domain for which you want to accept mail appears correctly.
5.1.4	`Destination mailbox address ambiguous`	Two or more recipients in the Exchange organization have the same address.	This error typically occurs because of a misconfiguration in Active Directory. Possibly because of replication problems, two recipient objects in Active Directory have the same SMTP address or

			Exchange Server (EX) address.
5.1.7	`Invalid address`	The sender has a malformed or missing SMTP address, the **mail** attribute in the directory service. The mail item cannot be delivered without a valid **mail** attribute.	Check the sender directory structure, and determine if the **mail** attribute exists.
5.2.1	`Mailbox cannot be accessed`	The mailbox cannot be accessed. The mailbox may be offline, disabled, or the message has been quarantined by a rule.	Check to see if the recipient database is online, the recipient mailbox is disabled, or the message has been quarantined.
5.2.2	`Mailbox full`	The recipient's mailbox has exceeded its storage quota and is no longer able to accept new messages.	This error occurs when the recipient's mailbox has exceeded its storage quota. The recipient must reduce the size of the mailbox or the administrator must increase the storage quota before delivery can be successful.
5.2.3	`Message too large`	The message is too large, and the local quota is exceeded. For example, a remote Exchange user might have a restriction	Send the message again without attachments, or set the server or the client-side limit to allow a larger message size limit.

		on the maximum size of an incoming message.	
5.2.4	`Mailing list expansion problem`	The recipient is a misconfigured dynamic distribution list. Either the filter string or the base DN of the dynamic distribution list is invalid.	Set the categorizer event logging level to at least the minimum level, and send another message to the dynamic distribution list. Check the application event log for a 6025 event or a 6026 event detailing which attribute is misconfigured on the dynamic distribution list object.
5.3.3	`Unrecognized command`	When the Exchange remote server reaches capacity of its disk storage to hold mail, it could respond with this NDR. This error usually occurs when the sending server is sending mail with an ESMTP BDAT command. This error also indicates a possible SMTP protocol error.	Ensure that the remote server has enough storage capacity to hold mail. Check the SMTP log.
5.3.4	`Message too big for system`	The message exceeds a size limit configured on a transport or mailbox database and can't be accepted. This failure can	This error occurs when the size of the message that was sent by the sender exceeds the maximum allowed

		be generated by either the sending email system or the recipient email system.	message size when passing through a transport component or mailbox database. The sender must reduce the size of the message for the message to be successfully delivered. For more information about how to configure message size limits, see Message size limits.
5.3.5	`System incorrectly configured`	A mail-looping situation was detected, which means that the server is configured to loop mail back to itself.	Check the configuration of the server's connectors for loops, and ensure that each connector is defined by a unique incoming port. If there are multiple virtual servers, ensure that none are set to "All Unassigned."
5.4.4	`Invalid arguments`	This NDR occurs if no route exists for message delivery, or if the categorizer could not determine the next-hop destination.	Check that the domain name specified is valid, and that a mail exchanger (MX) record exists.
5.4.6	`Routing loop detected`	A configuration error has caused an email loop. By default, after 20	This error occurs when the delivery of a message generates another

		iterations of an email loop, Exchange interrupts the loop and generates an NDR to the sender of the message.	message in response. That message then generates a third message, and the process is repeated, creating a loop. To help protect against exhausting system resources, Exchange interrupts the mail loop after 20 iterations. Mail loops are typically created because of a configuration error on the sending mail server, the receiving mail server, or both. Check the mailbox rules configuration of the recipient and sender to determine whether automatic message forwarding is enabled.
5.5.2	`Send hello first`	A generic SMTP error occurs when SMTP commands are sent out of sequence. For example, a server attempts to send an AUTH (authorization) command before identifying itself with an EHLO command.	View the SMTP Log or a Netmon trace, and ensure that there is adequate disk storage and virtual memory available.

		It is possible that this error can also occur when the system disk is full.	
5.5.3	`Too many recipients`	The combined total of recipients on the To, Cc, and Bcc lines of the message exceeds the total number of recipients allowed in a single message.	This error occurs when the sender has included too many recipients on the message. The sender must reduce the number of recipient addresses in the message or the maximum number of recipients must be increased to allow the message to be successfully delivered.
5.5.4	`Invalid domain name`	The message contains either an invalid sender or an incorrect recipient address format. One possible cause is that the recipient address format might contain characters that are not conforming to Internet standards.	Check the recipient address for nonstandard characters.
5.5.6	`Invalid message content`	This message indicates a possible protocol error.	Check Event Log for possible failures.
5.7.1	`Delivery not authorized`	The sender of the message is not allowed to send messages to the	This error occurs when the sender tries to send a message to a recipient

		recipient.	but the sender is not authorized to do this. This frequently occurs when a sender tries to send messages to a distribution group that has been configured to accept messages only from members of that distribution group or other authorized senders. The sender must request permission to send messages to the recipient. This error can also occur if an Exchange transport rule rejects a message because the message matched conditions that are configured on the transport rule.
5.7.1	`Unable to relay`	The sending email system is not allowed to send a message to an email system where that email system is not the final destination of the message.	This error occurs when the sending email system tries to send an anonymous message to a receiving email system, and the receiving email system does not accept messages for the domain or domains specified in one or more of the

			recipients. The following are the most common reasons for this error: • A third party tries to use a receiving email system to send spam, and the receiving email system rejects the attempt. By the nature of spam, the sender's email address may have been forged and the resulting NDR could have been sent to the unsuspecting sender's email address. It is difficult to avoid this situation. • An MX record for a domain points to a receiving email system where that domain is not accepted. The administrator responsible for the specific domain name must correct the MX record or configure the receiving email system to accept messages sent to that domain, or both. • A sending email system

			or client that should use the receiving email system to relay messages does not have the correct permissions to do this.
5.7.1	`Client was not authenticated`	The sending email system did not authenticate with the receiving email system. The receiving email system requires authentication before message submission.	This error occurs when the receiving server must be authenticated before message submission, and the sending email system has not authenticated with the receiving email system. The sending email system administrator must configure the sending email system to authenticate with the receiving email system for delivery to be successful. This error can also occur if you try to accept anonymous messages from the Internet on a Mailbox server that has not been configured to do this.
5.7.3	`Not Authorized`	The sender prohibited reassignment to the alternate recipient.	

Return to top

Manage DSN messages

Exchange Server 2013 > Mail flow > DSNs and NDRs >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-20*

Microsoft Exchange Server 2013 uses delivery status notifications (DSN) to provide non-delivery reports (NDRs) and other status messages to message senders. You can use the built-in DSNs, or you can create custom DSN messages to meet the needs of your organization.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "DSNs" entry in the Mail flow permissions topic.
- You can't remove a built-in DSN message that's included with Exchange. To change a built-in DSN message, you need to create a custom DSN message for the DSN code that you want to customize. When you remove a custom DSN message, the DSN code associated with that message reverts to the built-in DSN message that's included with Exchange.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to view built-in and custom DSN messages

To view a summary list of all built-in DSN messages included with Exchange 2013, run the following command:

```
Get-SystemMessage -Original
```

To view a summary list of all custom DSN messages in your organization, run the following command:

```
Get-SystemMessage
```

To view detailed information for the custom DSN message for DSN code 5.1.2 that's sent to internal

senders in English, run the following command:

```
Get-SystemMessage En\Internal\5.1.2 | Format-List
```

Use the Shell to create a custom DSN message

Run the following command:

```
New-SystemMessage -Internal <$true | $false> -Language
<Locale> -DSNCode <x.y.z> -Text "<DSN text>"
```

This example creates a custom plain text DSN message for the DSN code 5.1.2 that's sent to internal senders in English.

```
New-SystemMessage -Internal $true -Language En -DSNCode
5.1.2 -Text "You tried to send a message to a disabled
mailbox that's no longer accepting messages. Please contact
the Help Desk at extension 123 for assistance."
```

This example creates a custom plain text DSN message for the DSN code 5.1.2 that's sent to external senders in English.

```
New-SystemMessage -Internal $false -Language En -DSNCode
5.1.2 -Text "You tried to send a message to a disabled
mailbox that's no longer accepting messages. Please contact
your System Administrator for more information."
```

This example creates a custom HTML DSN message for the DSN code 5.1.2 that's sent to internal senders in English.

```
New-SystemMessage -DSNCode 5.1.2 -Internal $true -Language
En -Text 'You tried to send a message to a <B>disabled</B>
mailbox. Please visit <A HREF="http://
it.contoso.com">Internal Support</A> or contact
&quot;InfoSec&quot; for more information.'
```

How do you know this worked?

To verify that you have successfully created a custom DNS message, do the following:
1. Run the following command:

```
Get-SystemMessge -DSNCode <x.y.z> | Format-List
Name,Internal,Text,Language
```

2. Verify the values you see are the values you configured.
3. Send a test message that will generate the custom DSN you configured.

Use the Shell to change the text of a custom DSN message

To change the text of a custom DSN message the following command:

```
Set-SystemMessage <Locale>\<Internal | External>\<DSNcode>
-Text "<DSN text>"
```

This example changes the text assigned to the custom DSN message for DSN code 5.1.2 that's sent to internal senders in English.

```
Set-SystemMessage En\Internal\5.1.2 -Text "The mailbox you
tried to send an e-mail message to is disabled and is no
longer accepting messages. Please contact the Help Desk at
extension 123 for assistance."
```

How do you know this worked?

To verify that you have successfully changed the text of a custom DNS message, do the following:
1. Run the following command: `Get-SystemMessage`.

```
Set-SystemMessage <Locale>\<Internal | External>\<DSNcode>
| Format-List -Text
```

2. Verify the value displayed is the value you configured.

Use the Shell to remove a custom DSN message

Run the following command:

```
Remove-SystemMessage <Local>\<Internal | External>
\<DSNcode>
```

This example removes the custom DSN message for the DSN code 5.1.2 that's sent to internal senders in English.

```
Remove-SystemMessage En\Internal\5.1.2
```

How do you know this worked?

To verify that you have successfully removed a custom DNS message, do the following:
1. Run the command: `Get-SystemMessage`.
2. Verify a DSN for the locale, internal or external recipients, and DSN code you deleted isn't listed.

Forward copies of DSN messages to the Exchange recipient mailbox

You can specify a list of DSN codes that you want to monitor by having the DSN messages copied to the mailbox of the Exchange recipient. However, by default, no mailbox is assigned to the Exchange recipient, so any messages sent to the Exchange recipient are discarded. To send copies of DSN messages to the Exchange recipient mailbox, you need to assign a mailbox to the Exchange recipient, and then specific the DSN codes you want to monitor. By default, the following DSN codes are monitored: `5.4.8`, `5.4.6`, `5.4.4`, `5.2.4`, `5.2.0`, and `5.1.4`.

Step 1: Use the Shell to assign a mailbox to the Exchange recipient

To assign a mailbox to the Exchange recipient, perform the following steps:

1. Due to the potentially high volume of email, consider creating a dedicated mailbox and Active Directory user account for the Exchange recipient. For more information, see Create user mailboxes. Otherwise, identify the existing mailbox you want to associate with the Exchange recipient.
2. Run the following command:

```
Set-OrganizationConfig -
MicrosoftExchangeRecipientReplyRecipient <MailboxIdentity>
```

For example, to assign the existing mailbox named "Contoso System Mailbox" to the Exchange recipient, run the following command:

```
Set-OrganizationConfig -
MicrosoftExchangeRecipientReplyRecipient "Contoso System
Mailbox"
```

Step 2: Specify the DSN codes you want to monitor

Use the EAC to specify the DSN codes

1. In the EAC, navigate to **Mail flow** > **Receive connectors** > **More options** ··· > **Organization transport settings** > **Delivery**.
2. In the **DNS codes** section, type the DSN codes you want to monitor using the format *<x.y.z>*, and click **Add** +. Select an existing entry and click **Edit** ✎ to modify it, or click **Remove** − to remove it. When you are finished, click **Save**.

Use the Shell to specify the DSN codes

To replace the existing values, run the following command:

```
Set-TransportConfig -GenerateCopyOfDSNFor
```

```
<x.y.z>,<x.y.z>...
```

This example configures the Exchange organization to forward all DSN messages that have the DSN codes 5.7.1, 5.7.2, and 5.7.3 to the Exchange recipient.

```
Set-TransportConfig -GenerateCopyOfDSNFor 5.7.1,5.7.2,5.7.3
```

To add or remove entries without modifying any existing values, run the following command:

```
Set-TransportConfig -GenerateCopyOfDSNFor
@{Add="<x.y.z>","<x.y.z>"...;
Remove="<x.y.z>","<x.y.z>"...}
```

This example adds the DSN code 5.7.5 and removes the DSN code 5.7.1 from the existing list of DSN messages that are forwarded to the Exchange recipient.

```
Set-TransportConfig -GenerateCopyOfDSNFor @{Add="5.7.5";
Remove="5.7.1"}
```

How do you know this worked?

To verify that you successfully configured copies of DNS messages to be sent to the mailbox of the Exchange recipient, monitor the mailbox that's associated with the Exchange recipient, and verify the DSN messages contain the DSN codes you specified.

DSN message identity

Exchange Server 2013 > Mail flow > DSNs and NDRs >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-25*

You can identify a customized delivery status notification (DSN) message based on its syntax. The identity is the customized DSN message's GUID or a string that consists of the following values:

- **Locale** This variable specifies the locale of the language that the DSN message is displayed in. For a list of locale codes that you can use with the **New-SystemMessage** command, see Supported languages for system messages.
- **Internal or External** This variable specifies whether the DSN message is sent only to senders who are part of the internal Microsoft Exchange Server 2013 organization or also to senders outside the Exchange organization. You can use the Internal option when you want to include a specific e-mail contact or resolution in DSN messages sent to internal senders, but don't want to expose that information to senders outside your organization.
- **DSN code** This variable specifies the DSN code of the customized DSN message.

The syntax of the DSN message identity is `<Locale>\<Internal or External>\<DSN code>`.

For each DSN code, you can create more than one customized DSN message, which can target internal senders or external senders, and different locales. For example, the following table shows some of the possible configurations for the DSN code 5.1.2 and the corresponding DSN message identities.

Example DSN configurations and identities

DSN configuration	DSN identity
Display DSN messages to internal senders with an English (en) locale	En\Internal\5.1.2
Display DSN messages to external senders with an English (en) locale	En\External\5.1.2
Display DSN messages to internal senders with a Japanese (ja) locale	Ja\Internal\5.1.2
Display DSN messages to external senders with a Japanese (ja) locale	Ja\External\5.1.2

DSN message text

Exchange Server 2013 > Mail flow > DSNs and NDRs >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-25*

You can include text in a customized delivery status notification (DSN) message in Microsoft Exchange Server 2013, and you can format that text in HTML.

You can include any information that you want to display to the recipient of the DSN message. For example, you can include a detailed description of the DSN, contact information for your help desk, and a link to your support department's Web site. Each DSN message can contain a maximum of 512 characters.

Because DSN messages can be displayed in HTML, you can embed HTML formatting tags in the DSN text. For example, if you want to make some text in your DSN message bold, enclose the text in <B> and </B> HTML tags. The following table provides some examples of valid HTML tags that can be used in DSN message text.

Valid HTML tags for use in DSN messages

HTML tag	Description
<B>	Bold begin
</B>	Bold end
<A HREF="url">	Hyperlink begin
</A>	Hyperlink end
 	Link break
<EM>	Italic begin
</EM>	Italic end
<P>	Paragraph begin
</P>	Paragraph end

Note:

By default, Exchange sends HTML DSN messages, but you can configure whether Exchange sends HTML DSN messages to internal senders, external senders, or both. To configure this behavior, modify the *InternalDsnSendHtml* parameter and the *ExternalDsnSendHtml* parameter with the **Set-TransportService** command.
If the *InternalDsnSendHtml* parameter is set to `$false`, Exchange suppresses HTML tags in DSN messages sent to internal senders. If the *ExternalDsnSendHtml* parameter is set to `$false`, Exchange suppresses HTML tags in DSN messages sent to external senders.

The following characters that Exchange uses in DSN message text have special meanings:

- Greater than sign (>)
- Less than sign (<)
- Ampersand (&)
- Quotation marks (")

These characters are used to determine where HTML tags begin and end, and where text that should be displayed to senders starts and stops. If you want to display these characters in your DSN messages, you must use the escape codes in the following table.

For example, if you want to display the message `"Please contact the Help Desk at <1234>."`, you must add `"Please contact the Help Desk at <1234>."` to the DSN message text.

DSN message character escape codes

Escape code	Character

&lt;	<
&gt;	>
&quot;	"
&amp;	&

Important:
If you include an HTML tag in your DSN message text that contains quotation marks ("), such as `<A HREF="url">`, you must use single quotation marks (') around the whole DSN message text. You will receive an error message if you use double quotation marks around the whole DSN message text and around an HTML tag.

Supported languages for system messages

Exchange Server 2013 > Mail flow > DSNs and NDRs >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-25*

The following table lists the supported language codes you can use with the **New-SystemMessage** cmdlet.

Language code	Language
af	Afrikaans
am-ET	Amharic (Ethiopia)
ar	Arabic
as-IN	Assamese (India)
bg	Bulgarian
bn-BD	Bengali (Bangladesh)
bn-IN	Bengali (India)

bs-Cyrl-BA	Bosnian (Cyrillic, Bosnia and Herzegovina)
bs-Latn-BA	Bosnian (Latin, Bosnia and Herzegovina)
ca	Catalan
cs	Czech
cy-GB	Welsh (Great Britain)
da	Danish
de	German
el	Greek
en	English
es	Spanish
et	Estonian
eu	Basque
fa	Persian
fi	Finnish
fil-PH	Filipino (Philippines)
fr	French
ga-IE	Irish (Ireland)
gl	Galician
gu	Gujarati
ha-Latn-NG	Hausa (Latin, Nigeria)
he	Hebrew
hi	Hindi

hr	Croatian
hu	Hungarian
hy	Armenian
id	Indonesian
ig-NG	Igbo (Nigeria)
is	Icelandic
it	Italian
iu-Latn-CA	Inuktitut (Latin, Canada)
ja	Japanese
ka	Georgian
kk	Kazakh
km-KH	Khmer (Cambodia)
kn	Kannada
ko	Korean
kok	Konkani
ky	Kyrgyz
lb-LU	Luxembourgish (Luxembourg)
lo-LA	Lao (Lao People's Democratic Republic)
lt	Lithuanian
lv	Latvian
mi-NZ	Maori (New Zealand)
mk	Macedonian

ml-IN	Malayalam (India)
mr	Marathi
ms	Malay
ms-BN	Malay (Brunei Darussalam)
mt-MT	Maltese (Malta)
ne-NP	Nepali (Nepal)
nl	Dutch
nn-NO	Norwegian (Nynorsk)
no	Norwegian
nso-ZA	Sesotho sa Leboa (South Africa)
or-IN	Oriya (India)
pa	Punjabi
pl	Polish
ps-AF	Pashto (Afghanistan)
pt	Portuguese
pt-PT	Portuguese (Portugal)
qut-GT	K'iche (Guatemala)
quz-PE	Quechua (Peru)
ro	Romanian
ru	Russian
rw-RW	Kinyarwanda (Rwanda)
si-LK	Sinhala (Sri Lanka)

sk	Slovak
sl	Slovenian
sq	Albanian
sr	Serbian
sr-Cyrl-CS	Serbian (Cyrillic, Serbia)
sv	Swedish
sw	Kiswahili
ta	Tamil
te	Telugu
th	Thai
tn-ZA	Setswana (South Africa)
tr	Turkish
tt	Tatar
uk	Ukrainian
ur	Urdu
uz	Uzbek
vi	Vietnamese
wo-SN	Wolof (Senegal)
xh-ZA	isiXhosa (South Africa)
yo-NG	Yoruba (Nigeria)
zh-Hans	Chinese (Simplified)
zh-Hant	Chinese (Traditional)

zh-HK	Chinese (Hong Kong)
zu-ZA	isiZulu (South Africa)

Message size limits

Exchange Server 2013 > Mail flow >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-30*

You can apply limits to messages that move through the Microsoft Exchange Server 2013 organization. You can restrict the total size of a message or the size of the individual components of a message, such as the message header, the message attachments, and the number of recipients. You can apply limits globally for the whole Exchange organization, or specifically to a connector or user object.

As you plan the message size limits for your Exchange organization, consider the following questions:

- What size limits should I impose on all incoming messages?
- What size limits should I impose on all outgoing messages?
- What is the mailbox quota for my Exchange organization?
- How do the message size limits that I have chosen relate to the mailbox quota size?
- Are there users in my Exchange organization who must send or receive messages that are larger than the specified allowed size?
- Does my Exchange network topology include other messaging systems or distinctly separate business units that have different message size limits?

This topic provides guidance to help you answer these questions.

Contents

Types of message size limits

Following are the basic categories of the size limits available for individual messages:

- **Message header size limits** These limits apply to the total size of all message header fields that

are present in a message. The size of the message body or attachments isn't considered. Because the header fields are plain text, the size of the header is determined by the number of characters in each header field and by the total number of header fields. Each character of text consumes 1 byte.

Note:

Some third-party firewalls or proxy servers apply their own message header size limits. These third-party firewalls or proxy servers may have difficulty processing messages that contain attachment file names that are greater than 50 characters or attachment file names that contain non-US-ASCII characters.

- **Message size limits** These limits apply to the total size of a message, which includes the message header, the message body, and any attachments. Message size limits may be imposed on incoming messages or outgoing messages. For internal message flow, Exchange uses the custom `X-MS-Exchange-Organization-OriginalSize:` message header to record the original message size of the message as it enters the Exchange organization. Whenever the message is checked against the specified message size limits, the lower value of the current message size or the original message size header is used. The size of the message can change because of content conversion, encoding, and agent processing.
- **Attachment size limits** These limits apply to the maximum allowed size of a single attachment within a message. The message may contain many attachments that greatly increase the overall size of the message. However, an attachment size limit applies to the size of an individual attachment only.
- **Recipient limits** These limits apply to the total number of message recipients. When a message is first composed, the recipients exist in the `To:`, `Cc:`, and `Bcc:` header fields. When the message is submitted for delivery, the message recipients are converted into `RCPT TO:` entries in the message envelope. A distribution group is counted as a single recipient during message submission.

Return to top

Scope of limits

Following are the basic categories for the scope of the limits available for individual messages:

- **Organizational limits** These limits apply to all Exchange 2013 Mailbox servers and Exchange 2010 and Exchange 2007 Hub Transport servers that exist in the organization. If you have an Edge Transport server installed in the perimeter network, the specified limits apply to the specific server.
- **Connector limits** These limits apply to any messages that use the specified Send connector, Receive connector, Delivery Agent connector, or Foreign connector for message delivery. Send connectors are defined in the Transport service on Mailbox servers and on Edge Transport servers. Receive connectors are defined in the Transport service on Mailbox servers, in the Front End Transport service on Client Access servers, and on Edge Transport servers.
- **Active Directory site links** The Transport service on Mailbox servers use Active Directory sites and the costs that are assigned to the Active Directory IP site links as one of the factors to

determine the least-cost routing path between Mailbox servers in the organization. You can assign specific message size limits to the Active Directory site links in your organization.

- **Server limits** These limits apply to a specific Mailbox server or Edge Transport server. You can set the specified message limits independently on each Mailbox server or Edge Transport server.

In Outlook Web App, the maximum HTTP request size limit setting on the Client Access servers also controls the size of messages that Outlook Web App users can send.

- **User limits** These limits apply to a specific user object, such as a mailbox, contact, distribution group, or public folder.

The following tables show the message limits, including information about how to configure the limits in the Exchange Management Shell or the Exchange Administrator Center (EAC).

Organizational limits

Size limit	Default value	Shell configuration	EAC configuration
Maximum size for messages received	10 MB	Cmdlet: **Set-TransportConfig** Parameter: *MaxReceiveSize*	**Mail flow** > **Receive connectors** > **More options** ··· > **Organization transport settings** > **Limits** tab > **Maximum receive message size**
Maximum size for messages sent	10 MB	Cmdlet: **Set-TransportConfig** Parameter: *MaxSendSize*	**Mail flow** > **Receive connectors** > **More options** ··· > **Organization transport settings** > **Limits** tab > **Maximum send message size**
Maximum number of recipients per message **Note:**	500	Cmdlet: **Set-TransportConfig** Parameter: *MaxRecipientEnvelopeLimit*	**Mail flow** > **Receive connectors** > **More options** ··· > **Organization transport settings** > **Limits** tab > **Maximum number of recipients**

Maximum attachment size in Transport rules that apply to all Mailbox servers in the organization	Not configured	Cmdlets: **New-TransportRule**, **Set-TransportRule** Parameter: *AttachmentSizeOver*	**Mail flow** > **Rules** > **Add +** or **Edit** . Use the predicate **Apply this rule if** > **Any attachment** > **is greater than or equal to** Use the predicate **Apply this rule if** > **The message** > **size is greater than or equal to**
Maximum message size in Transport rules that apply to all Mailbox servers in the organization		Cmdlets: **New-TransportRule**, **Set-TransportRule** Parameter: *MessageSizeOver*	**Mail flow** > **Rules** > **Add +** or **Edit** . Use the predicate **Apply this rule if** > **The message** > **size is greater than or equal to**

Return to top

Connector limits

Size limit	Default value	Shell configuration	EAC configuration
Maximum header size through a Receive connector	128 KB	Cmdlets: **New-ReceiveConnector**, **Set-ReceiveConnector** Parameter: *MaxHeaderSize*	N/A
Maximum message size through a Receive	**Transport service on Mailbox servers**	Cmdlets: **New-ReceiveConnector**,	**Mail flow** > **Receive connectors** > **Edit** >

connector **Note:** The actual message size may be smaller due to message encoding and content conversion.	35 MB for the Default and Client Proxy Receive connectors **Front End Transport service on Client Access servers** 36 MB for the Default Frontend and Outbound Proxy Frontend Receive connectors. 35 MB for the Client Frontend Receive connector.	**Set-ReceiveConnector** Parameter: *MaxMessageSize*	**General** tab > **Maximum receive message size**
Maximum number of recipients per message through a Receive connector	**Transport service on Mailbox servers** 5,000 for the Default Receive connector 200 for the Client Proxy Receive connector **Front End Transport service on Client Access servers** 200 for the Default Frontend, Client Frontend, and Client Proxy Frontend Receive connectors. **Note:** If the number of recipients is exceeded	Cmdlets: **New-ReceiveConnector**, **Set-ReceiveConnector** Parameter: *MaxRecipientsPerMessage*	N/A

	for an anonymous sender, the message is accepted for the first 200 recipients. Most SMTP messaging servers detect that a recipient limit is in effect. The SMTP messaging server continues to resend the message in groups of 200 recipients until the message is delivered to all recipients.		
Maximum message size through a Send connector	10 MB	Cmdlets: **New-SendConnector**, **Set-SendConnector** Parameter: *MaxMessageSize*	**Mail flow** > **Send connectors** > **Edit** > **General** tab > **Maximum send message size**
Maximum message size through an Active Directory site link	Unlimited	Cmdlet: **Set-AdSiteLink** Parameter: *MaxMessageSize*	N/A
Maximum message size through a delivery agent connector	Unlimited	Cmdlets: **New-DeliveryAgentConnector**, **Set-DeliveryAgentConnector** Parameter: *MaxMessageSize*	N/A
Maximum message size through a foreign connector	Unlimited	**Cmdlet: Set-ForeignConnector** Parameter: *MaxMessageSize*	N/A

Return to top

Server limits

Size limit	Default value	Shell configuration	EAC configuration
Maximum header size for messages in the pickup directory	64 KB	Cmdlet: **Set-TransportService** Parameter: *PickupDirectoryMaxHeaderSize*	N/A
Maximum number of recipients per message for messages in the pickup directory	100	Cmdlet: **Set-TransportService** Parameter: *PickupDirectoryMaxRecipientsPerMessage*	N/A
Client-specific maximum messages size limits for Outlook Web App, Exchange ActiveSync, and Exchange Web Services clients	Outlook Web App 35 MB Exchange ActiveSync 10 MB Exchange Web Services 64 MB **Note:** These values are approximately 33% larger than the actual usable maximum message size because of the overhead that's associated with Base64 encoding.	You configure these values in the appropriate web.config XML application configuration file on Client Access servers. For more information, see Configure client-specific message size limits.	N/A

Return to top

User limits

Size Limit	Default value	Shell configuration	EAC configuration
Maximum message	Unlimited	Cmdlets:	For mailboxes:

size that can be sent by this recipient		**Set-DistributionGroup** **Set-DynamicDistributionGroup** **Set-Mailbox** **Set-MailContact** **Set-MailUser** **Set-MailPublicFolder** **Set-RemoteMailbox** Parameter: *MaxSendSize*	**Recipients** > **Mailboxes** > **Edit** > **Mailbox features** > **Mail flow** > **Message size restrictions** > **View details** > **Sent messages** **Note:** This setting isn't configurable using the EAC for other recipient types.
Maximum message size that can be sent to this recipient	Unlimited For site mailbox provisioning policies: 36 MB	Cmdlets: **Set-DistributionGroup** **Set-DynamicDistributionGroup** **Set-Mailbox** **Set-MailContact** **Set-MailUser** **Set-MailPublicFolder** **New-SiteMailboxProvisioningPolicy** **Set-SiteMailboxProvisioningPolicy**	For mailboxes: **Recipients** > **Mailboxes** > **Edit** > **Mailbox features** > **Mail flow** > **Message size restrictions** > **View details** > **Received messages** **Note:** This setting isn't configurable using the EAC for other recipient types.

		Parameter: *MaxReceiveSize*	
Maximum number of recipients per message sent by this recipient	Unlimited	Cmdlets: **Set-Mailbox**, **Set-MailUser** Parameter: *RecipientLimits*	N/A

Return to top

Order of precedence for message size limits

You can set different message size limits at different levels in the Exchange organization. As a message is routed through your Transport infrastructure, it may be subjected to several different message size restrictions. You should plan your message size restrictions in a way that makes sure that messages in the transport pipeline are rejected as early as possible if they violate message size limits. Generally speaking, you should set more restrictive limits at the points where messages enter your infrastructure. For example, any message size restrictions on your Receive connectors that receive messages from the Internet should be less than or equal to the message size restrictions you configure for your internal Exchange organization. It would be a waste of system resources for the Exchange server to accept and process a message from the Internet that would be rejected by the Transport service on your Mailbox servers. Make sure that your organization, server, and connector limits are configured in a way that minimizes any unnecessary processing of messages.

One exception to this approach is the user limits. User level limits take precedence over other message size restrictions. Therefore, you can configure a user to exceed the default message size limits for your organization. For example, you can allow a specific group of user mailboxes to send larger messages than the rest of the organization by configuring custom send and receive limits for those mailboxes.

The exceptions for the user limits only apply to message exchanges between authenticated users. If a message is sent to or received by a recipient on the Internet, the organizational limits will be applied. For example, assume that you have an organizational message size restriction of 10 MB, but you have configured the users in your marketing department to send and receive messages up to 50 MB. These users will be able to exchange large messages with each other, but they still won't be able to receive large messages from Internet users because such messages will be coming from unauthenticated senders.

Return to top

Messages exempt from size limits

The following list shows the types of messages generated by a Mailbox server or an Edge Transport server and exempted from all message size limits:

- System messages
- Agent-generated message
- Delivery status notification (DSN) messages
- Journal report messages
- Quarantined messages

However, these messages are still subject to the organizational value for maximum number of recipients in a message.

Return to top

Message throttling

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

Message throttling refers to a group of limits that are set on the number of messages and connections that can be processed by a Microsoft Exchange Server 2013 computer. These limits prevent the accidental or intentional exhaustion of system resources on the Exchange server.

Contents

Message throttling scope

Message throttling involves a variety of limits on message processing rates, SMTP connection rates, and SMTP session time-out values. These limits work together to protect an Exchange server from being overwhelmed by accepting and delivering messages. Although a large backlog of messages

and connections may be waiting to be processed, the message throttling limits enable the Exchange server to process the messages and connections in an orderly manner.

In addition to message throttling, you can also put size limits on the individual components of messages, such as the number of recipients, the size of the message header, or the size of individual attachments. For more information about message size limits, see Message size limits.

Another feature that helps avoid overwhelming the system resources of an Exchange transport server is *back pressure*. Back pressure is a system resource monitoring feature in the Transport service on Mailbox servers and on Edge Transport servers. When a monitored system resource, such as hard disk utilization or memory utilization, exceeds the specified threshold, the server reduces the rate at which it accepts new connections and messages, and focuses on delivering existing messages. When the utilization of the monitored system resources returns to normal levels, the server slowly increases the rate at which it accepts new connections and then establishes a normal level.

Return to top

Message cost and mail flow throttling

To provide more consistent message throughput and predictable message delivery latency, Exchange 2013 establishes an accumulated cost for messages. This quality of service (QoS) feature was added in Microsoft Exchange Server 2010 SP1This cost is based on the following criteria:

- Message size
- Number of recipients
- Frequency of transmission

Exchange 2013 transport servers track the average delivery cost of messages that are sent by individual users. By using message costs, Exchange 2013 provides a group of settings that can control the effect that a user or connection has on an Exchange organization. This group of settings is known as a *throttling policy*. When a user repeatedly sends costly messages, such as messages that have large attachments or messages that are sent to many recipients, the Exchange 2013-based transport servers use a throttling policy to assign a lower priority to higher-cost messages from the user while continuing to deliver lower-cost messages. This new behavior adds a "quality of service" aspect to the message throttling functionality in Exchange 2013.

Note:

Message throttling doesn't affect the message priority from a user's perspective. Messages still retain the original priority set by the user. For example, messages retain a setting of Important or Urgent, and so on.

To support this functionality, Exchange 2013 uses the following mechanisms:

- **Internal prioritization agent** This agent is triggered on the **OnResolvedMessage** event and assigns a lower priority to messages from the same sender that have a high accumulated cost. This cost is measured over a period of one minute and affects messages that have more than 500 recipients or that are larger than 1 megabyte (MB).

- **Quota-based priority queuing for the MapiDelivery queue type** This mechanism causes Exchange to deliver messages in a normal-priority queue more frequently than messages in a low-priority queue. By default, the normal-to-low message ratio is 20:1. However, new messages from a lower priority queue are never delivered sooner than new items from a higher priority queue. For example, consider the following scenario:
  1. Twenty normal priority messages are delivered. By default, the next delivered message is a lower priority message.
  2. Two new messages are received by the transport server: One message from a higher priority queue and one message from a lower priority queue.

In this scenario, the message from the higher priority queue is delivered first. Then, the message from the lower priority queue is delivered.

- **Throttle concurrent connections based on messaging database health** This mechanism monitors the health of the Exchange messaging database (MDB) health and throttles concurrent connections to Exchange transport servers based on an assigned Health Measure value. The MDB is monitored by the Resource Health Monitor API in the Transport service on the Mailbox server and is assigned a health value from -1 through 100. This value is based on the RPC performance statistics that are included with each RPC response from the Store.exe process in the Mailbox Transport service. The Resource Health framework uses both the **Requests/Second** rate performance counter and the **Average RPC Latency** performance counter to calculate a health value for the database. To help maintain a consistent interactive user experience, Exchange reduces the number of concurrent connections as the health value decreases. The following health value ranges are available:
  1. **-1:** This value indicates that the MDB health state is unknown. This value is assigned when the database starts. In this scenario, the database is considered healthy.
  2. **0:** This value is assigned if the database is in an unhealthy state. In this state, the database should not be contacted.
  3. **1 through 99:** These values represent a fair health state. A lower value represents a less healthy database.
  4. **100:** This value represents a healthy database.

The Microsoft Exchange Throttling service provides the framework for mail flow throttling. The Microsoft Exchange Throttling service keeps track of mail flow throttling settings for a specific user and caches the throttling information in memory. Mail flow throttling settings are also known as a *budget*. Restarting the Microsoft Exchange Throttling service also resets mail flow throttling budgets.

You can use the throttling policy cmdlets that are available in Exchange 2013 to configure individual budget settings for a throttling policy. A budget is the amount of access that a user or application may have for a specific setting. A budget represents how many connections a user may have or how much activity a user may be permitted for each one-minute period. For example, a budget may be configured to set the amount of time that a user may spend using a specific feature in Exchange, such as ActiveSync, Outlook Web App, or Exchange Web Services. This threshold is stored in a throttling policy and defines the budget.

Time settings for a budget are set as a percentage of one minute. Therefore, a threshold of 100 percent represents 60 seconds. For example, assume that you want to specify Outlook Web App policy settings that limit the amount of time during which a user may run Outlook Web App code on a Client Access server and the amount of time the user may communicate with the Client Access server to 600 milliseconds over a one-minute period. To accomplish this, you need to set the value to 1 percent of one minute (600 milliseconds) for both of the following parameters:

- **OWAPercentTimeInCAS:** 1
- **OWAPercentTimeInMailboxRPC:** 1

A user who has this policy applied has a budget of OWAPercentTimeInCAS of 600 milliseconds and of OWAPercentageTimeInMailboxRPC of 600 milliseconds. In this scenario, when the user is logged into Outlook Web App, the user can run Client Access code for up to 600 milliseconds. After the 600 millisecond-period, the connection is considered over budget and the Exchange server doesn't allow any further Outlook Web App action until one minute after the budget limit is reached. After the one-minute period, the user can run Outlook Web App client access code for another 600 milliseconds.

Return to top

Message throttling on servers

You can set the message throttling options at the following locations:

- In the transport service
- On a Send connector
- On a Receive connector

You can set all the message throttling options that are available in the Transport service on Mailbox servers, in the Mailbox Transport service on Mailbox servers, or in the Front End Transport service on Client Access servers using the Exchange Management Shell. You can also set some of the same options by configuring the transport server properties in the Exchange Administration Center (EAC).

The following table shows the message throttling options that are available on transport servers.

Message throttling options on transport servers

Source	Parameter	Default value	Description
Set-TransportService **Set-MailboxTransportService**	*MaxConcurrentMailboxDeliveries*	20	This parameter specifies the maximum number of delivery threads that the transport service can have open at the same time to deliver

			messages to mailboxes. The valid input range for this parameter is from 1 through 256. We recommend that you don't modify the default value unless Microsoft Customer Service and Support advises you to do this.
Set-TransportService **Set-MailboxTransportService**	*MaxConcurrentMailboxSubmissions*	20	This parameter specifies the maximum number of submission threads that the transport service can have open at the same time to send messages from mailboxes. The valid input range for this parameter is from 1 through 256.
Set-TransportService	*MaxConnectionRatePerMinute*	1200	This parameter specifies the maximum rate that connections are allowed to be opened with the transport service. If many connections are attempted with the transport service at the same time, the *MaxConnectionRatePer*

			Minute parameter limits the rate that the connections are opened so that the server's resources aren't overwhelmed.
Set-TransportService or Transport server properties	*MaxOutboundConnections*	1000	This parameter specifies the maximum number of outbound connections that can be open at a time. If you enter a value of `unlimited`, no limit is imposed on the number of outbound connections. The value of this parameter must be greater than or equal to the value of the *MaxPerDomainOutboundConnections* parameter. This value can also be configured using the EAC at **Servers** > **Servers** > **Properties** > **Transport limits** > **Outbound connection restrictions**.
Set-TransportService	*MaxPerDomainOutbou*	20	This parameter

or Transport server properties	*ndConnections*		specifies the maximum number of concurrent connections to any single domain. If you enter a value of `unlimited`, no limit is imposed on the number of outbound connections per domain. The value of this parameter must be greater than or equal to the value of the *MaxOutboundConnections* parameter. This value can also be configured using the EAC at **Servers** > **Servers** > **Properties** > **Transport limits** > **Outbound connection restrictions**.
Set-TransportService	*PickupDirectoryMaxMessagesPerMinute*	100	This parameter specifies the maximum number of messages processed per minute by the Pickup directory and by the Replay directory. Each directory can independently process

			message files at the rate specified by this parameter.

Return to top

Message throttling on Send connectors

The following table shows the message throttling option that's available on Send connectors. You need to use the Exchange Management Shell to configure this option.

Message throttling option available on Send connectors

Source	Parameter	Default value	Description
Set-SendConnector	*ConnectionInactivityTimeOut*	10 minutes	This parameter specifies the maximum time that an open SMTP connection with a destination messaging server can remain idle before the connection is closed.

Return to top

Message throttling on Receive connectors

The following table shows the message throttling options that are available on Receive connectors that are configured in the Transport service on a Mailbox server or on an Edge Transport server. You need to use the Exchange Management Shell to configure these options.

Message throttling options available on Receive connectors

Source	Parameter	Default value	Description
Set-ReceiveConnector	*ConnectionInactivityTimeOut*	5 minutes in the Transport service on Mailbox servers 5 minutes in the Front End Transport service	This parameter specifies the maximum time that an open SMTP connection with a source messaging

		on Client Access servers. 1 minute on Edge Transport servers.	server can remain idle before the connection is closed. The value of this parameter must be smaller than the value specified by the *ConnectionTimeout* parameter.
Set-ReceiveConnector	*ConnectionTimeOut*	10 minutes in the Transport service on Mailbox servers 10 minutes in the Front End Transport service on Client Access servers. 5 minutes on Edge Transport servers.	This parameter specifies the maximum time that an SMTP connection with a source messaging server can remain open, even if the source messaging server is transmitting data. The value of this parameter must be larger than the value specified by the *ConnectionInactivityTimeout* parameter.
Set-ReceiveConnector	*MaxInboundConnection*	5000	This parameter specifies the maximum number of inbound SMTP connections that this Receive connector allows at the same time.
Set-ReceiveConnector	*MaxInboundConnectionPercentagePerSource*	100 percent on the Default Receive connector in the	This parameter specifies the maximum

		Transport service on a mailbox server 2 percent on the other Receive connectors in the Transport service on Mailbox servers, and in the Front End Transport service on Client Access servers.	number of SMTP connections that a Receive connector allows at the same time from a single source messaging server. The value is expressed as the percentage of available remaining connections on a Receive connector. The maximum number of connections that are permitted by the Receive connector is defined by the *MaxInboundConnection* parameter.
Set-ReceiveConnector	*MaxInboundConnectionPerSource*	**unlimited** on the Default Receive connector in the Transport service on a mailbox server 20 on the other Receive connectors in the Transport service on Mailbox servers, and in the Front End Transport service on Client Access servers.	This parameter specifies the maximum number of SMTP connections that a Receive connector allows at the same time from a single source messaging server.
Set-ReceiveConnector	*MaxProtocolErrors*	5	This parameter specifies the maximum

			number of SMTP protocol errors that a Receive connector allows before the Receive connector closes the connection with the source messaging server.
Set-ReceiveConnector	*TarpitInterval*	5 seconds	This parameter specifies the delay that's used in *tarpitting*. Tarpitting is the practice of artificially delaying SMTP responses for specific SMTP communication patterns that indicate a directory harvest attack or other unwelcome messages. A *directory harvest attack* is an attempt to collect valid e-mail addresses from a particular organization to use as a target for unsolicited commercial e-mail. The delay that's specified by the *TarpitInterval* parameter only applies to anonymous

			connections.

Return to top

Message throttling policies

In Exchange 2013, each mailbox has a *ThrottlingPolicy* setting. The default value for this setting is blank (`$null`). You can use the *ThrottlingPolicy* paramenter on the **Set-Mailbox** cmdlet to configure a throttling policy for a mailbox.

A default throttling policy exists to provide a default set budget configuration for users who connect to Exchange. To configure customized budget settings for one or more users, create a new throttling policy. Then, apply the policy to the appropriate user or group.

Important:
We recommend that you don't modify the default throttling policy.

You can set all the message throttling options that are available on Mailbox servers in the Exchange Management Shell. The following cmdlets are available to manage throttling policies:

- **Get-ThrottlingPolicy**
- **Remove-ThrottlingPolicy**
- **New-ThrottlingPolicy**
- **Set-ThrottlingPolicy**

You can use the **New-ThrottlingPolicy** and **Set-ThrottlingPolicy** cmdlets to configure how much activity a user can perform against Exchange over a specific connection or time period. These settings make up a user's budget. You can establish throttling policies to control access to the following Exchange features:

- Exchange ActiveSync
- Exchange Web Services
- Outlook Web App
- Unified Messaging
- IMAP4
- POP3
- Outlook client connections (MAPI or RPC connections)
- Mail flow settings
- PowerShell commands
- CPU usages

Return to top

Back pressure

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-20*

Back pressure is a system resource monitoring feature of the Microsoft Exchange Transport service that exists on Microsoft Exchange 2013 Mailbox servers and Edge Transport servers.

Exchange can detect when vital resources, such as available hard drive space and memory, are under pressure, and take action in an attempt to prevent service unavailability. Back pressure prevents the system resources from being completely overwhelmed, and the Exchange server tries to process the existing messages before accepting any new messages. When utilization of the system resource returns to a normal level, the Exchange server gradually resumes normal operation and starts accepting new messages again.

In Exchange 2013, when the Transport service on a Mailbox server or an Edge Transport server is under resource pressure, incoming connections are accepted, but incoming messages over those connections are either accepted at a slower rate or are rejected. When an SMTP host attempts to connect to an Exchange server that's under resource pressure, the connection will succeed. However, when the host issues the **MAIL FROM** command to submit a message, depending on the resource that's under pressure, the Transport service either delays the acknowledgement of the **MAIL FROM** command or rejects the connection.

Contents

Resources monitored

The following system resources are monitored as part of the back pressure feature:

- Free space on the hard drive that stores the message queue database.
- Free space on the hard drive that stores the message queue database transaction logs.
- The number of uncommitted message queue database transactions that exist in memory.
- The memory that's used by the EdgeTransport.exe process.
- The memory that's used by all other processes.
- The number of messages in the Submission queue.

For each monitored system resource on a Mailbox server or Edge Transport server, the following three levels of resource utilization are applied:

- **Normal** The resource isn't overused. The server accepts new connections and messages.
- **Medium** The resource is slightly overused. Back pressure is applied to the server in a limited

manner. Mail from senders in the authoritative domain can flow. However, depending on the specific resource under pressure, the server uses tarpitting to delay server response or rejects incoming **MAIL FROM** commands from other sources.

- **High** The resource is severely overused. Full back pressure is applied. All message flow stops, and the server rejects all new incoming **MAIL FROM** commands.

The following sections explain how Exchange handles the situation when a specific resource is under pressure.

Free hard drive space for the message queue database

By default, the message queue database is stored at %ExchangeInstallPath%TransportRoles\data\Queue. Exchange monitors the hard drive space utilization for this location. The high level of hard drive space utilization is calculated by using the following formula:

100 * (*hard disk size* - *fixed constant*) / *hard drive size*

The value of *fixed constant* is 500 megabytes (MB).

The results of this formula are expressed as a percentage of the total hard drive space that's being used. The results of the formula are always rounded down to the nearest integer. By default, the medium level of hard drive utilization is 2 percent less than the high level. By default, the normal level of hard drive utilization is 4 percent less than the high level.

Return to top

Free hard drive space for the message queue database transaction logs

By default, the message queue database transaction logs are stored at %ExchangeInstallPath%TransportRoles\data\Queue. Exchange monitors the hard drive space utilization for this location. The %ExchangeInstallPath%Bin\EdgeTransport.exe.config application configuration file contains a *DatabaseCheckPointDepthMax* key that has a default value of 384 MB. This key controls the total allowed size of all uncommitted transaction logs that exist on the hard drive. This key is used in the formula that calculates hard drive utilization.

Note:

The value of the *DatabaseCheckPointDepthMax* key applies to all transport-related Extensible Storage Engine (ESE) databases that exist on the Mailbox server or Edge Transport server. This would include the message queue database and the IP filter database.

By default, the high level of disk utilization is calculated by using the following formula:

100 * (*hard drive size* - Min(5 GB, 3**DatabaseCheckPointDepthMax*)) / *hard drive size*

The results of the formula are always rounded down to the nearest integer. By default, the medium level of hard drive utilization is 2 percent less than the high level. The normal level of hard drive

utilization is 4 percent less than the high level.

Return to top

Number of uncommitted message queue database transactions in memory

A list of changes that are made to the message queue database is kept in memory until those changes can be committed to a transaction log. Then the list is committed to the message queue database itself. These outstanding message queue database transactions that are kept in memory are known as *version buckets*. The number of version buckets may increase to unacceptably high levels because of an unexpectedly high volume of incoming messages, spam attacks, problems with the message queue database integrity, or hard drive performance.

When Exchange starts receiving messages, these messages are grouped together in batches and then prepared as version buckets. If an incoming message has a large attachment, it can be separated into multiple batches. These batches that are being processed are known as *batch points*. The number of outstanding batch points can exceed the set thresholds, especially when there's an unexpectedly high volume of incoming messages with large attachments.

When version buckets or batch points are under pressure, the Exchange server will start throttling incoming connections by delaying acknowledgement to incoming messages. Exchange will reduce the rate of inbound message flow by tarpitting, which introduces a delay to the **MAIL FROM** commands. If the resource pressure condition continues, Exchange will gradually increase the tarpitting delay. After the resource utilization returns to normal, Exchange will gradually start reducing the acknowledgement delay and ease into normal operation. By default, Exchange will start delaying message acknowledgements 10 seconds when under resource pressure. If the resources continue to be under pressure, the delay is increased in 5-second increments up to 55 seconds.

Exchange keeps a history of version bucket and batch point resource utilization. If the resource utilization doesn't go down to normal level for a specific number of polling intervals, known as the history depth, Exchange will stop the tarpitting delay and start rejecting incoming messages until the resource utilization goes back to normal. By default, the history depths for version buckets and batch points are in 10 and 300 polling intervals respectively.

Return to top

Memory used by the EdgeTransport.exe process

By default, the high level of memory utilization by the EdgeTransport.exe process is calculated by using the following formula:

75 percent of the total physical memory or 1 terabyte, whichever is less

This calculation doesn't include virtual memory that's available on the hard drive in the paging file, or the memory that's used by other processes. The results of this formula are expressed as a percentage of the total memory that's used by the EdgeTransport.exe process. The results of the formula are always rounded down to the nearest integer.

By default, the medium level of memory utilization by the EdgeTransport.exe file is calculated as 73 percent of the total physical memory or 2 percent less than the value of the high level, whichever is less. By default, the normal level of memory utilization by the EdgeTransport.exe file is calculated as 71 percent of the total physical memory or 4 percent less than the value of the high level, whichever is less.

If the memory utilization of the EdgeTransport.exe process is higher than the specified normal level, *garbage collection* is forced. Garbage collection is a process that checks for unused objects that exist in memory, and reclaims the memory that's used by those unused objects.

Exchange keeps a history of the memory utilization of the EdgeTransport.exe process. If the utilization doesn't go down to normal level for a specific number of polling intervals, known as the history depth, Exchange will start rejecting incoming messages until the resource utilization goes back to normal. By default, the history depth for EdgeTransport.exe memory utilization is 30 polling intervals.

Return to top

Memory used by all processes

By default, the high level of memory utilization by all processes is 94 percent of total physical memory. This value doesn't include virtual memory that's available on the hard drive in the paging file.

When the specified memory utilization level is reached, *message dehydration* occurs. Message dehydration is the act of removing unnecessary elements of queued messages that are cached in memory. Complete messages are cached in memory for enhanced performance. Removal of the MIME content of queued messages from memory reduces the memory that's used at the expense of higher latency because the messages are read directly from the message queue database. By default, message dehydration is enabled.

Return to top

Number of messages in the Submission queue

The Submission queue is associated with the Transport service on Exchange 2013 Mailbox servers and on Edge Transport servers. The categorizer processes each message in the Submission queue. This categorization results in the message being put in a delivery queue. For more information, see Mail flow and Queues. A large number of messages in the Submission queue indicates the categorizer is having difficulty processing messages.

When the Submission queue is under pressure, the Exchange server will start throttling incoming connections by delaying acknowledgement to incoming messages. Exchange will reduce the rate of inbound message flow by tarpitting, which introduces a delay to the **MAIL FROM** commands. If the Submission queue pressure condition continues, Exchange will gradually increase the tarpitting delay. After the Submission queue utilization returns to normal, Exchange will gradually start reducing the acknowledgement delay and ease into normal operation. By default, Exchange will start delaying message acknowledgements 10 seconds when under Submission queue pressure. If the Submission queue continues to be under pressure, the delay is increased in 5-second increments up to 55 seconds.

Exchange keeps a history of Submission queue utilization. If the Submission queue utilization doesn't go down to normal level for a specific number of polling intervals, known as the history depth, Exchange will stop the tarpitting delay and start rejecting incoming messages until the Submission utilization goes back to normal. By default, the history depth for the Submission queue is in 300 polling intervals.

Actions taken by Exchange Transport when under resource pressure

The following table summarizes the actions taken by Exchange transport when a specific resource is under pressure.

Back pressure actions taken by Mailbox and Edge Transport servers when responding to resource pressure

Resource under pressure	Utilization level	Actions taken
Hard drive space for message queue database	Medium	• Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories
Hard drive space for message queue database	High	• Reject incoming messages from other Exchange servers • Reject message submissions from mailbox databases by the Mailbox Transport Submission service on Mailbox servers • Reject incoming messages from non-Exchange servers

		• Reject message submissions from Pickup and Replay directories
Hard drive space for message queue database transaction logs	Medium	• Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories
Hard drive space for message queue database transaction logs	High	• Reject incoming messages from other Exchange servers • Reject message submissions from mailbox databases by the Mailbox Transport Submission service on Mailbox servers • Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories
Version buckets	Medium	Introduce or increment the tarpitting delay to incoming messages. If normal level isn't reached for the entire version bucket history depth, take the following actions: • Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories
Version buckets	High	Introduce or increment the tarpitting delay to incoming messages. If normal level isn't

		reached for the entire version bucket history depth, take the following actions: • Reject incoming messages from other Exchange servers • Reject message submissions from mailbox databases by the Mailbox Transport Submission service on Mailbox servers • Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories
Batch point	Medium	Introduce or increment the tarpitting delay to incoming messages. If normal level isn't reached for the entire batch point history depth, take the following actions: • Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories
Batch point	High	Introduce or increment the tarpitting delay to incoming messages. If normal level isn't reached for the entire batch point history depth, take the following actions: • Reject incoming messages from other Exchange servers

		• Reject message submissions from mailbox databases by the Mailbox Transport Submission service on Mailbox servers • Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories
Memory used by EdgeTransport.exe process	Medium	• Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories • Force garbage collection
Memory used by EdgeTransport.exe process	High	• Reject incoming messages from other Exchange servers • Reject message submissions from mailbox databases by the Mailbox Transport Submission service on Mailbox servers • Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories
Memory used by all processes	Medium	• Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories • Force garbage collection
Memory used by all processes	High	• Reject incoming messages from other Exchange servers

		• Reject message submissions from mailbox databases by the Mailbox Transport Submission service on Mailbox servers • Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories • Flush enhanced DNS cache from memory • Start message dehydration
Number of messages in the Submission queue	Medium	Introduce or increment the tarpitting delay to incoming messages. If normal level isn't reached for the entire Submission queue history depth, take the following actions: • Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories • Force garbage collection
Number of messages in the Submission queue	High	Introduce or increment the tarpitting delay to incoming messages. If normal level isn't reached for the entire Submission queue history depth, take the following actions: • Reject incoming messages from other Exchange servers • Reject message submissions

		from mailbox databases by the Mailbox Transport Submissions service on Mailbox servers • Reject incoming messages from non-Exchange servers • Reject message submissions from Pickup and Replay directories • Flush enhanced DNS cache from memory • Start message dehydration

Return to top

Back pressure configuration options in the EdgeTransport.exe.config file

All configuration options for back pressure are available in the %ExchangeInstallPath%Bin\EdgeTransport.exe.config XML application configuration file.

Caution:

These settings are listed as a reference only. We strongly discourage any modifications to the back pressure settings in the EdgeTransport.exe.config file. Modifications to the back pressure settings may result in poor performance or data loss. We recommend that you investigate and correct the root cause of any back pressure events that you may encounter.

Back pressure configuration options

Key name	Default value
EnableResourceMonitoring	true
ResourceMonitoringInterval	`00:00:02` (2 seconds)
PercentageDatabaseDiskSpaceUsedHighThreshold	0. This value indicates that the default formula will be used.
PercentageDatabaseDiskSpaceUsedMediumThreshold	0. This value indicates that the actual value is 2 percent less than the value of *PercentageDatabaseDiskSpaceUsedHighThresh*

	old.
PercentageDatabaseDiskSpaceUsedNormalThreshold	0. This value indicates that the actual value is 2 percent less than the value of *PercentageDatabaseDiskSpaceUsedMediumThreshold*.
PercentageDatabaseLoggingDiskSpaceUsedHighThreshold	0. This value indicates that the default formula will be used.
PercentageDatabaseLoggingDiskSpaceUsedMediumThreshold	0. This value indicates that the actual value is 2 percent less than the value of *PercentageDatabaseLoggingDiskSpaceUsedHighThreshold*.
PercentageDatabaseLoggingDiskSpaceUsedNormalThreshold	0. This value indicates that the actual value is 2 percent less than the value of *PercentageDatabaseLoggingDiskSpaceUsedMediumThreshold*.
PercentagePrivateBytesUsedHighThreshold	0. This value indicates that the default calculation will be used.
PercentagePrivateBytesUsedMediumThreshold	0. This value indicates that the actual value is 2 percent less than the value of *PercentagePrivateBytesUsedHighThreshold*.
PercentagePrivateBytesUsedNormalThreshold	0. This value indicates that the actual value is 2 percent less than the value of *PercentagePrivateBytesUsedMediumThreshold*.
VersionBucketsHighThreshold	200
VersionBucketsMediumThreshold	120
VersionBucketsNormalThreshold	80
VersionBucketsHistoryDepth	10

BatchPointHighThreshold	4000
BatchPointMediumThreshold	2000
BatchPointNormalThreshold	1000
BatchPointHistoryDepth	300
BatchPointUseCostForPressure	true
BatchPointBatchSize	40
BatchPointBatchTimeout	`00:00:00.100` (0.1 seconds)
BatchPointItemExpiryInterval	`00:05:00` (5 minutes)
SMTPBaseThrottlingDelayInterval	`00:00:00`
SMTPMaxThrottlingDelayInterval	`00:00:55` (55 seconds)
SMTPStepThrottlingDelayInterval	`00:00:05` (5 seconds)
SMTPStartThrottlingDelayInterval	`00:00:10` (10 seconds)
PercentagePhysicalMemoryUsedLimit	94
DehydrateMessagesUnderMemoryPressure	true
PrivateBytesHistoryDepth	30
SubmissionQueueHighThreshold	10000
SubmissionQueueMediumThreshold	4000
SubmissionQueueNormalThreshold	2000
SubmissionQueueHistoryDepth	300

Return to top

Back pressure logging information

The following list describes the event log entries that are generated by specific back pressure

events in Exchange:

- **Event log entry for an increase in any resource utilization level**

Event Type: Error

Event Source: MSExchangeTransport

Event Category: Resource Manager

Event ID: 15004

Description: Resource pressure increased from *Previous Utilization Level* to *Current Utilization Level*.

- **Event log entry for a decrease in any resource utilization level**

Event Type: Information

Event Source: MSExchangeTransport

Event Category: Resource Manager

Event ID: 15005

Description: Resource pressure decreased from *Previous Utilization Level* to *Current Utilization Level*.

- **Event log entry for critically low available disk space**

Event Type: Error

Event Source: MSExchangeTransport

Event Category: Resource Manager

Event ID: 15006

Description: The Microsoft Exchange Transport service is rejecting messages because available disk space is below the configured threshold. Administrative action may be required to free disk space for the service to continue operations.

- **Event log entry for critically low available memory**

Event Type: Error

Event Source: MSExchangeTransport

Event Category: Resource Manager

Event ID: 15007

Description: The Microsoft Exchange Transport service is rejecting message submissions because the service continues to consume more memory than the configured threshold. This may require that this service be restarted to continue normal operation.

Return to top

Queues

Exchange Server 2013 > Mail flow >

Topic Last Modified: *2014-01-31*

A *queue* is a temporary holding location for messages that are waiting to enter the next stage of

processing or delivery to a destination. Each queue represents a logical set of messages that the Exchange server processes in a specific order. In Microsoft Exchange Server 2013, queues hold messages before, during and after delivery. Queues exist on Mailbox servers and Edge Transport servers. Mailbox servers and Edge Transport servers are called *transport servers* throughout this topic.

Like the previous versions of Exchange, Exchange 2013 uses a single Extensible Storage Engine (ESE) database for queue storage.

You can manage queues and the messages in queues using the Exchange Management Shell and Queue Viewer in the Exchange Toolbox. You can use these interfaces to view the status and contents of queues and detailed message properties. You can also use these interfaces to perform actions that modify queues or the messages in the queues.

Contents

Types of queues

The following types of queues are used in Exchange 2013:

- **Persistent queues** *Perisistent queues* are queues that exist on every transport server in every Exchange organization. Like previous versions of Exchange, there are three persistent queues in Exchange 2013:
  - **Submission queue** The Submission queue is used by the categorizer to gather all messages that have to be resolved, routed, and processed by transport agents on the transport server. All messages that are received by a transport server enter processing in the Submission queue. On Mailbox servers, messages are submitted through a Receive connector, the Pickup or Replay directories, or the Mailbox Transport Submission service. On Edge Transport servers, messages are typically submitted through a Receive connector, but the Pickup and Replay directories are also available.

The categorizer retrieves messages from this queue and, among other things, determines the location of the recipient and the route to that location. After categorization, the message is moved to a delivery queue or to the Unreachable queue. Each transport server has only one Submission queue. Messages that are in the Submission queue can't be in other queues at the same time. For

more information about the categorizer and the transport pipeline, see Mail flow.

- **Unreachable queue** The Unreachable queue contains messages that can't be routed to their destinations. Typically, an unreachable destination is caused by configuration changes that have modified the routing path for delivery. Regardless of destination, all messages that have unreachable recipients reside in this queue. Each transport server has only one Unreachable queue.

Messages in the Unreachable queue are automatically resubmitted when a routing change is detected. So, after the condition or configuration error caused the messages to enter the Unreachable queue is repaired, you don't need to take additional action to move the messages out of the Unreachable queue for delivery.

The Unreachable queue is typically empty. If the Unreachable queue contains no messages it doesn't appear in Queue Viewer or **Get-Queue** results.

- **Poison message queue** The poison message queue is a special queue that's used to isolate messages that are determined to be harmful to the Exchange 2013 system after a transport server or service failure. The messages may be genuinely harmful in their content and format. Alternatively, they may be the results of a poorly written agent that has caused the Exchange server to fail when it processed the supposedly bad messages.

The poison message queue is typically empty. If the poison message queue contains no messages it doesn't appear in Queue Viewer or **Get-Queue** results. The messages in the poison message queue are never automatically resumed or expired. Messages remain in the poison message queue until they're manually resumed or removed by an administrator.

- **Delivery queues** Delivery queues hold messages that are being delivered to any local or remote destinations by using SMTP. All messages are transmitted between Exchange servers by using SMTP. Non-SMTP destinations also use delivery queues if the destination is serviced by a Delivery Agent connector. . Each delivery queue contains messages that are being routed to the same destination. It's practically inevitable that multiple delivery queues will exist on a transport server. Delivery queues are dynamically created when they're required and are automatically deleted when the queue is empty and the expiration time has passed. The queue expiration time is controlled by the *QueueMaxIdleTime* parameter on the **Set-TransportService** cmdlet. The default value is three minutes.
- **Shadow queues** Shadow queues hold redundant copies of a message while the message is in transit. For more information, see Shadow redundancy.
- **Safety Net** Safety Net retains copies of messages that were successfully delivered by the transport server. Although it's not accessible by queue management tools, Safety Net is just another queue in the queue database. For more information, see Safety Net.

Return to top

Queue database files

All the different queues are stored in a single ESE database. By default, this queue database is located on the transport server at `%ExchangeInstallPath%TransportRoles\data\Queue`.

Like any ESE database, the queue database uses log files to accept, track, and maintain data. To enhance performance, all message transactions are written first to log files and memory, and then to the database file. The checkpoint file tracks the transaction log entries that have been committed to the database. During an ordinary shutdown of the Microsoft Exchange Transport service, uncommitted database changes that are found in the transaction logs are always committed to the database.

Circular logging is used for the queue database. This means that the history of committed transactions that are found in the transaction logs isn't maintained. Any transaction logs that are older than the current checkpoint are immediately and automatically deleted. Therefore, the transaction logs can't be replayed for queue database recovery from backup.

Exchange 2013 uses *generation tables* for storage and clean-up of messages in the queue database. Instead of processing and deleting individual message records from one large table, the queue database stores messages in time-based tables, and only deletes the entire table after all the messages in the table have been successfully processed. For example, all messages queued from 1:00 PM to 2:00 PM, regardless of the queue or destination, are stored in the `1p-2p_msgs` table. At 2:00 PM, new messages are stored in the `2p-3p_msgs` table. At 4:00 PM, a new table named `4p-5p_msgs` is created, and the entire `1p-2p_msgs` table is deleted, but only if all messages in the table have been successfully processed. This approach of deleting entire messages tables instead of individual messages helps improves the I/O performance of the drive that holds the queue database.

The following table lists the files that constitute the queue database.

Files that constitute the queue database

File	Description
Mail.que	This queue database file stores all the queued messages.
Tmp.edb	This temporary database file is used to verify the queue database schema on startup.
Trn*.log	This transaction log records all changes to the queue database. Changes to the database are first written to the transaction log and then committed to the database. Trn.log is the current active transaction log file. Trntmp.log is the next provisioned transaction log file that's created in advance. If the existing Trn.log transaction log file reaches its maximum size,

	Trn.log is renamed to Trn*nnnn*.log, where *nnnn* is a sequence number. Trntmp.log is then renamed Trn.log and becomes the current active transaction log file.
Trn.chk	This checkpoint file tracks the transaction log entries that have been committed to the database. This file is always in the same location as the mail.que file.
Trnres00001.jrs Trnres00002.jrs	These reserve transaction log files act as placeholders. They're only used when the hard disk that contains the transaction log runs out of space to stop the queue database cleanly.

Return to top

Options for configuring the queue database

You configure the queue database by adding or modifying keys in the `%ExchangeInstallPath%Bin\EdgeTransport.exe.config` XML application configuration file. This file is associated with the Microsoft Exchange Transport service. Changes you make to the EdgeTransport.exe.config file take effect after you restart the Microsoft Exchange Transport service.

The `<appSettings>` section of the EdgeTransport.exe.config file is where you can add new keys or modify existing keys. If a specific key doesn't exist, you can add it manually to change its value.

The keys for the queue database that are available in the EdgeTransport.exe.config file are described in the following table.

Message queue database keys that are available in the EdgeTransport.exe.config file

Key	Default value	Description
QueueDatabaseBatchSize	40	This key specifies the number of database I/O operations that can be grouped together before they're executed. By default, this key doesn't exist in the EdgeTransport.exe.config file.

QueueDatabaseBatchTimeout	100	This key specifies the maximum time in milliseconds that the database will wait for multiple database I/O operations to group before it executes them. The database I/O operations are executed without waiting for any more if the following conditions are true: • The number of database I/O operations that's specified by the *QueueDatabaseBatchSize* key hasn't been reached. • The time specified by the *QueueDatabaseBatchTimeout* key has passed. By default, this key doesn't exist in the EdgeTransport.exe.config file.
QueueDatabaseMaxConnections	4	This key specifies the number of ESE database connections that can be open.
QueueDatabaseLoggingBufferSize	5 MB	This key specifies the memory that's used to cache the transaction records before they're written to the transaction log file.
QueueDatabaseLoggingFileSize	5 MB	This key specifies the maximum size of a transaction log file. When the maximum log file size is reached, a new log file is opened.
QueueDatabaseLoggingPath	`%ExchangeInstallPath%TransportRoles\data\Queue`	This key specifies the default directory for the queue database

		log files. For instructions on how to change the location of the queue database, see Change the location of the queue database.
QueueDatabaseMaxBackgroundCleanupTasks	32	This key specifies the maximum number of background cleanup work items that can be queued to the database engine thread pool at any time.
QueueDatabaseOnlineDefragEnabled	True	The key enables or disables scheduled online defragmentation of the mail queue database. By default, this key doesn't exist in the EdgeTransport.exe.config file.
QueueDatabaseOnlineDefragSchedule	`1:00:00` or 1:00 A.M.	This key specifies the time of day in 24 hour format to start the online defragmentation of the mail queue database. To specify a value, enter the value as a time: *hh:mm:ss*, where *h* = hours, *m* = minutes, and *s* = seconds.
QueueDatabaseOnlineDefragTimeToRun	`3:00:00` or 3 hours	This key specifies the length of time the online defragmentation task is allowed to run. Even if the defragmentation task doesn't finish in the time specified, the queue database is left in a consistent state. To specify a value, enter the value as a time span: *hh:mm:ss*, where *h* = hours,

		m = minutes, and *s* = seconds.
QueueDatabasePath	`%ExchangeInstallPath%TransportRoles\data\Queue`	This key specifies the default directory for the queue database files. For instructions on how to change the location of the queue database, see Change the location of the queue database.

Note:

Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.

Return to top

Queue properties

A queue has many properties that describe the purpose and status of the queue. Some queue properties are applied to the queue when the queue is created, and don't change. Other properties contain status size, time, or other indicators that are updated frequently.

Return to top

NextHopSolutionKey

The routing component of the categorizer in the Microsoft Exchange Transport service selects the destination for a message, and this destination is used to create the delivery queue. The destination is stamped on every recipient as the **NextHopSolutionKey** attribute. Every unique value of the **NextHopSolutionKey** attribute corresponds to a separate delivery queue.

The **NextHopSolutionKey** attribute contains the following fields:

- **DeliveryType** The value of this field represents the results of the categorization of the message, and how the Transport service intends to transmit the message to the next hop, which could be the ultimate destination of the message, or an intermediate hop along the way. The Transport service uses a predefined list of values for **DeliveryType** based on the target routing destination or delivery group.
- **NextHopDomain** This field uses specific values based on the value of the **DeliveryType** field. For delivery queues, the value of this field is effectively the name of the queue. The value of **NextHopDomain** isn't always a domain name. For example, the value could be the name of the

target Active Directory site or database availability group (DAG). Think of this field as the next hop name, where the value is the name of the routing destination or the target delivery group.

- **NextHopConnector** This field uses specific values based on the value of the **DeliveryType** field. The value is always expressed as a GUID. If this field isn't used, the value is a GUID with all zeroes. The value of **NextHopConnector** isn't always the GUID of a connector. For example, the value could be the GUID of the target Active Directory site or DAG. Think of this field as the next hop GUID, where the value is the GUID of the routing destination or the target delivery group.

Exchange 2013 also adds the **NextHopCategory** property to the queue based on the value of **DeliveryType**. The value of **NextHopCategory** is `External` or `Internal`. The value `External` indicates the next hop of the queue is outside the Exchange organization. The value `Internal` indicates the next hop of the queue is inside the Exchange organization. Note that a message for an external recipient may require one or more internal hops before the message is delivered externally.

The values of **DeliveryType**, **NextHopCategory**, **NextHopDomain** and **NextHopConnector** are described in the following table.

Delivery Type in Queue Viewer	DeliveryType in the Shell	Description	NextHopCategory	NextHopDomain	NextHopConnector
Delivery Agent	**DeliveryAgent**	The queue holds messages for delivery to recipients in a non-SMTP address space. The messages are delivered by using a Delivery Agent connector that's configured on the local server.	External	This value is the destination address space that's configured on the Delivery Agent connector.	This value is he GUID of the Delivery Agent connector. For example, `4520e633-d83d-411a-bbe4-6a84648674ee`.
DnsConnectorDelivery	**DnsConnectorDelivery**	The queue holds messages for delivery to recipients in an SMTP address	External	This value is the destination address space that's configured on the Send	This value is the GUID of the Send connector. For example,

		space. The messages are delivered by using a Send connector that's configured on the local server. The Send connector is configured to use DNS routing.		connector. For example, `contoso.com`.	`4520e633-d83d-411a-bbe4-6a84648674ee`.
NonSmtpGatewayDelivery	**NonSmtpGatewayDelivery**	The queue holds messages for delivery to recipients in a non-SMTP address space. The messages are delivered by using a Foreign connector that's configured on the local server.	External	This value is the destination address space that's configured on the Foreign connector.	This value is the GUID of the Foreign connector. For example, `4520e633-d83d-411a-bbe4-6a84648674ee`.
SmartHostConnectorDelivery	**SmartHostConnectorDelivery**	The queue holds messages for delivery to recipients in an SMTP address space. The messages are delivered by using a Send connector that's	External	This value is the list of smart hosts that are configured on the Send connector. Smart hosts can be configured as FQDNs, IP addresses or both. The values	This value is the GUID of the Send connector. For example, `4520e633-d83d-411a-bbe4-6a84648674ee`.

		configured on the local server. The Send connector is configured to use smart host routing.		can be one of the following: • **FQDN** The syntax is `<FQDN1,FQDN2,...>`. For example, `smarthost01.contoso.com` or `smarthost01.contoso.com,smarthost02.fabrikam.com`. • **IP address** The syntax is `<[IPAddress1],[IPAddress2],...>`. For example, `[10.10.10.100]` or `[10.10.10.100],[10.10.10.101]`. • **FQDN and IP address** The syntax is `<[IPAddress1],FQDN1,...>`, and depends on how the smart hosts are listed on the Send connector. For example, `[172.17.17.7],relay.tailspintoys.com` or	

				`mail.contoso.com, [192.168.1.50]`.	
SMTP Delivery to Mailbox	**SmtpDeliveryToMailbox**	The queue holds messages for delivery to Exchange 2013 mailbox recipients. The destination mailbox database is in one of the following locations: • The local Exchange 2013 Mailbox server. • An Exchange 2013 Mailbox server in the same DAG. • An Exchange 2013 Mailbox server in the same Active Directory site in non-DAG environments.	Internal	This value is the name of the destination mailbox database. For example, `Mailbox Database 0471695037`.	This value is the GUID of the target mailbox database. For example, `6dcb5a1e-0a88-4fc9-b8f9-634c34b1a123`.
SMTP Relay to Send Connector Source Servers	**SmtpRelayToConnectorSourceServers**	The queue holds messages for delivery to SMTP or non-SMTP recipients. The	Internal	This value is the name of the destination Send connector, Delivery Agent	This value is the GUID of the destination Send

		messages are delivered by using a Send connector, Delivery Agent connector, or Foreign connector that's configured on a remote transport server. The remote transport server could be an Exchange 2013 Mailbox server, or an Exchange 2007 or Exchange 2010 Hub Transport server from a previous version of Exchange. The remote server could be located in the local Active Directory site, or in a remote Active Directory site.		connector, or Foreign connector. For example, `Contoso.com Send Connector`.	connector, Delivery Agent connector, or Foreign connector. For example, `4520e633-d83d-411a-bbe4-6a84648674ee`.
SMTP Relay to Database Availability Group	**SmtpRelayToDag**	The queue holds messages for delivery to Exchange 2013 mailbox	Internal	This value is the name of the destination DAG. For example, `DAG1`.	This value is the GUID of the destination DAG. For

		recipients, where the destination mailbox database is located in a remote DAG. The remote DAG could be in the local Active Directory site, or a remote Active Directory site.			example, `6dcb5a1e-0a88-4fc9-b8f9-634c34b1a123`
SMTP Relay to Mailbox Delivery Group	**SmtpRelayToMailboxDeliveryGroup**	The queue holds messages for delivery to legacy mailbox recipients, where the destination mailbox is on an Exchange 2007 or Exchange 2010 Mailbox server. The message is related to a Hub Transport server that's running the same version of Exchange as the destination mailbox. The destination Hub Transport server could be in the	Internal	The queue name uses the syntax: `Site:<ADSiteName>;Version:<ExchangeVersion>`, where *<ADSiteName>* is the name of the destination Active Directory site, and *<ExchangeVersion>* is the version of Exchange on the Mailbox server.	This value is blank.

		local Active Directory site, or a remote Active Directory site.			
SMTP Relay to Remote Active Directory Site	**SmtpRelayToRemoteActiveDirectorySite**	The queue holds messages for delivery to a remote destination, and the routing topology requires the message to be routed through a specific Active Directory site. The site is an intermediate hop on the way to the final destination. This situation occurs under the following circumstances: • The message needs to be routed through a hub site. • The message requires delivery through a Send connector that's configured on	Internal	This value is the target Active Directory site name. For example, `NorthAmericanSite`.	This value is the GUID of the target Active Directory site. For example, `bfd6c3df-5b65-8bfb-53f1f2c0d55c`.

		an Edge Transport server that's subscribed to a remote Active Directory site.			
SMTP Relay to Specified Exchange Servers	**SmtpRelayToServers**	The queue holds messages for delivery to a distribution group that's configured for a specific expansion server. The expansion could be an Exchange 2013 Mailbox server, or an Exchange 2007 or Exchange 2010 Hub Transport server. The server could be in the local Active Directory site, or in a remote Active Directory site.	Internal	This value is the FQDN of the target expansion server. For example, `mailbox01.contoso.com`.	This value is `00000000-0000-0000-0000-000000000000`.
SMTP Relay in Active Directory Site to Edge Transport Server	**SmtpRelayWithinAdSiteToEdge**	The queue holds messages for delivery to an SMTP address space. The messages are	Internal	This value is the name of the Send connector that sends outbound Internet mail from the organization	This value is the GUID of the Send connector. For example, `4520e633-d83d-411a-bbe4-`

		delivered by using a Send connector that's configured on an Edge Transport server that's subscribed to the local Active Directory site.		to the Internet. This Send connector is automatically created by the Edge subscription, and is named `EdgeSync - <ADSiteName> to Internet`. *<ADSiteName>* is the name of the local Active Directory site to which the Edge Transport server is subscribed.	`6a84648674ee.`
Heartbeat	**Heartbeat**	This value is reserved for internal Microsoft use. For more information about heartbeat, see Shadow redundancy.	n/a	n/a	n/a
Shadow Redundancy	**ShadowRedundancy**	The queue holds messages in a shadow queue. A shadow queue holds redundant copies messages in transit in case the primary	Internal	This value is the FQDN of the primary server for which the shadow queue is holding redundant copies of the primary messages. For	This value is `00000000-0000-0000-0000-000000000000.`

		messages aren't successfully delivered. For more information, see Shadow redundancy.		example, `mailbox01.contoso.com`.	
Undefined	**Undefined**	This value is used only on the Submission queue and the poison message queue.	Internal	For the Submission queue, this value is `Submisssion`. For the poison message queue, this value is `Poison Message`.	This value is `00000000-0000-0000-0000-000000000000`.
Undreachable	**Unreachable**	This value is used only on the Unreachable queue.	Internal	This value is `Unreachable Domain`.	This value is `00000000-0000-0000-0000-000000000000`.

Note that Exchange 2013 supports legacy values of **DeliveryType** for backwards compatibility with previous versions of Exchange. These values are available in Queue Viewer and the Shell, but they aren't used by Exchange 2013. These legacy **DeliveryType** values are:

- **MapiDelivery** The queue holds messages for delivery by an Exchange 2007 or Exchange 2010 Hub Transport server to a mailbox on an Exchange 2007 or Exchange 2010 Mailbox server in the local Active Directory site.
- **SmtpRelayWithinAdSite** The queue holds messages for delivery by an Exchange 2007 or Exchange 2010 Hub Transport server to another Hub Transport server in the same Active Directory site. The destination Hub Transport server can be the source server for a connector, or a distribution group expansion server.
- **SmtpRelaytoTiRg** The queue holds messages for delivery by an Exchange 2007 or Exchange 2010 Hub Transport server to an Exchange Server 2003 routing group. The destination server can be the source server for a connector, a distribution group expansion server, or an Exchange 2003 bridgehead server.

Return to top

IncomingRate, OutgoingRate, and Velocity

Exchange 2013 measures the rate of messages entering and leaving a queue and stores these values in queue properties. You can use these rates as an indicator of queue and transport server health. The properties are:

- **IncomingRate** This property is the rate that messages are entering the queue.

This value is calculated from the number of messages entering the queue every 5 seconds averaged over the last 60 seconds. The formula can be expressed as `(i1+i2+i3+i4+i5+i6)/6`, where i*n* = the number of incoming messages in 5 seconds.

- **OutgoingRate** This property is the rate that messages are leaving the queue.

This value is calculated from the number of messages leaving the queue every 5 seconds averaged over the last 60 seconds. The formula can be expressed as `(o1+o2+o3+o4+o5+o6)/6`, where o*n* = the number of outgoing messages in 5 seconds.

- **Velocity** This property is the drain rate of the queue, and is calculated by subtracting the value of **IncomingRate** from the value of **OutgoingRate**.

If the value of **Velocity** is greater than 0, messages are leaving the queue faster than they are entering the queue.

If the value of **Velocity** is equals 0, messages are leaving the queue as fast as they are entering the queue. This is also the value you'll see when the queue is inactive.

If the value of **Velocity** is less than 0, messages are entering the queue faster than they are leaving the queue.

At a basic level, a positive value of **Velocity** indicates a healthy queue that's efficiently draining, and a negative value of **Velocity** indicates a queue that isn't efficiently draining. However, you also need to consider the values of the **IncomingRate**, **OutgoingRate**, and **MessageCount** properties, as well as the magnitude of the **Velocity** value for the queue. For example, a queue that has a large negative value of **Velocity**, a large **MessageCount** value, a small **OutgoingRate** value, and a large **IncominRate** value are accurate indicators that the queue isn't draining properly. However, a queue with a negative **Velocity** value that's very close to zero that also has very small values for **IncomingRate**, **OutgoingRate**, and **MessageCount** doesn't indicate a problem with the queue.

Return to top

Queue status

The current status of a queue is stored in the **Status** property of the queue. A queue can have one of the following status values:

- **Active** The queue is actively transmitting messages.
- **Connecting** The queue is in the process of connecting to the next hop.
- **Ready** The queue recently transmitted messages, but the queue is now empty.
- **Retry** The last automatic or manual connection attempt failed, and the queue is waiting to retry the connection.

- **Suspended** The queue has been manually suspended by an administrator to prevent message delivery. New messages can enter the queue, and messages that are in the act of being transmitted to the next hop will finish delivery and leave the queue. Otherwise, messages won't leave the queue until the queue is manually resumed by an administrator. Note that suspending a queue doesn't change the status of the individual messages in the queue.

You can suspend a queue that has a status of Active or Retry. You can also suspend the Unreachable queue and the Submission queue.

If you suspend the Unreachable queue, messages won't be automatically resubmitted to the categorizer when configuration updates are detected. To automatically resubmit these messages, you need to manually resume the Unreachable queue. If you suspend the Submission queue, messages won't be picked up by the categorizer until the queue is resumed.

Return to top

Other queue properties

There are other queue properties that are self-explanatory. You use most of the queue properties as filter options. By specifying filter criteria, you can quickly locate queues and take action on them. For a complete description of the filterable queue properties, see Queue filters.

An important queue property that's also worth mentioning here is the **MessageCount** property that shows how many messages are in a queue. This property is an important indicator of queue health. For example, a delivery queue that contains a large number of messages that continues to grow and never decreases could indicate a routing or transport pipeline issue that requires your attention.

Return to top

Message properties

A message in a queue has many properties. Many of the properties reflect the information that was used to create the message. Some of the messages status and information properties are heavily influenced by corresponding properties on the queue. However, an individual message may have a different value than the corresponding property of the queue. Other properties contains status, time, or other indicators that are updated frequently.

Return to top

Message status

The current status of a message is stored in the **Status** property of the message. A message can have one of the following status values:

- **Active** If the message is in a delivery queue, the message is being delivered to its destination. If the message is in the Submission queue, the message is being processed by the categorizer.

- **Locked** This value is reserved for internal Microsoft use, and isn't used in on-premises Exchange organizations.
- **PendingRemove** The message was deleted by the administrator, but the message was already in the act of being transmitted to the next hop. The message will be deleted if the delivery ends in an error that causes the message to reenter the queue. Otherwise, delivery will continue.
- **PendingSuspend** The message was suspended by the administrator, but the message was already in the act of being transmitted to the next hop.. The message will be suspended if the delivery ends in an error that causes the message to reenter the queue. Otherwise, delivery will continue.
- **Ready** The message is waiting in the queue and is ready to be processed.
- **Retry** The last automatic or manual connection attempt for the queue in which this message is located failed. The message is waiting for the next automatic queue connection retry.
- **Suspended** The message was manually suspended by the administrator. All messages in the poison message queue are in a permanently suspended state.

Return to top

Other message properties

There are other message properties that are self-explanatory. You can use most of the message properties as filter options. By specifying filter criteria, you can quickly locate messages and take action on them. For a complete description of the filterable message properties, see Message filters.

Return to top

Manage queues and messages in queues

Queue Viewer and virtually all of the queue and message management cmdlets are restricted to a single Exchange server. You can view or operate on individual queues or messages, or multiple queues or messages, but only on a specific server.

Exchange 2013 introduces the **Get-QueueDigest** cmdlet that provides a high-level, aggregate view of the state of queues on all servers within a specific scope, for example, a DAG, an Active Directory site, a list of servers, or the entire Active Directory forest. Note that queues on a subscribed Edge Transport server in the perimeter network aren't included in the results. Also, **Get-QueueDigest** is available on an Edge Transport server, but the results are restricted to queues on the Edge Transport server.

Note:
By default, the **Get-QueueDigest** cmdlet displays delivery queues that contain ten or more messages, and the results are between one and two minutes old. For instructions on how to change these default values, see Configure Get-QueueDigest.

The following table describes the management tasks you can perform on queues or messages in

queues.

Task	Description	Tool to use	Instructions
View and filter queues on a server	This action displays one or more queues on a transport server. You can use the results to take action on the queues.	Queue Viewer or the **Get-Queue** cmdlet.	Manage queues
View and filter queues on specific servers in specific DAGs, specific Active Directory sites, or in the whole Active Directory forest.	This action displays a summary view of queues across a defined scope (servers, DAGs, Active Directory sites, or the entire Active Directory forest).	**Get-QueueDigest** cmdlet only	Manage queues
Suspend queues	This action temporarily prevents delivery of messages that are currently in the queue. The queue continues to accept new messages, but no messages leave the queue.	Queue Viewer or the **Suspend-Queue** cmdlet.	Manage queues
Resume queues	This action reverses the effect of the suspend queue action and enables delivery of queued messages to resume.	Queue Viewer or the **Resume-Queue** cmdlet.	Manage queues
Retry queues	This action immediately tries to	Queue Viewer or the **Retry-Queue** cmdlet.	Manage queues

	connect to the next hop. Without manual intervention, when the connection to the next hop fails, the connection is attempted a specific number of times after a specific time interval between each attempt. Whether the connection attempt is manual or automatic, any connection attempt resets the next retry time. For more information, see Message retry, resubmit, and expiration intervals.		
Resubmit messages in queues	This action causes the messages in the queue to be resubmitted to the Submission queue and to go back through the categorization process.	**Retry-Queue** with the *Resubmit* parameter Note that you can use Queue Viewer to resubmit messages, but only from the poison message queue. To resubmit a message in poison message, you resume the message in Queue Viewer, or by using the **Resume-**	Manage queues

		Message cmdlet.	
Suspend messages in queues	This action temporarily prevents delivery of a message. You can use the suspend message action to prevent delivery of a message to all the recipients in a specific queue or to all recipients in all queues.	Queue Viewer or the **Suspend-Message** cmdlet.	Manage messages in queues
Resume messages in queues	This action reverses the effect of the suspend message action and enables delivery of queued messages to resume. You can use the resume message action to resume delivery of a message to all the recipients in a specific queue or to all recipients in all queues.	Queue Viewer or the **Resume-Message** cmdlet.	Manage messages in queues
Remove messages from queues	This action permanently prevents delivery of a message. You can use the remove message action to prevent delivery of a message to any recipients in a specified queue or to	Queue Viewer or the **Remove-Message** cmdlet.	Manage messages in queues

	all recipients in all queues. You can also configure the remove message action to send a non-delivery report (NDR) to the sender when the message is removed.		
Export messages from queues	This action copies a message to the file path that you specify. The messages aren't deleted from the queue, but a copy of the message is saved to a file location. This enables administrators or officials in an organization to later examine the messages. Before you export a message, you need to suspend the message in the queue so that typical delivery doesn't continue during the export process.	**Export-Message** cmdlet only.	Export messages from queues

Return to top

Manage queues

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-31*

In Microsoft Exchange Server 2013, you can use the Queue Viewer in the Exchange Toolbox or the Exchange Management Shell to manage queues. For more information about using the queue management cmdlets in the Exchange Management Shell, see Use the Exchange Management Shell to manage queues.

What do you need to know before you begin?

- Estimated time to complete each procedure: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

View queues

Use Queue Viewer in the Exchange Toolbox to view queues

1. Click **Start** > **All Programs** > **Microsoft Exchange 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer** to open the tool in a new window.
3. In Queue Viewer, click the **Queues** tab. A list of all queues on the server to which you're connected is displayed.
4. You can use the **Export List** link in the action pane to export the list of queues. For more information, see Export lists from Queue Viewer.

Use the Shell to view queues

To view queues, use the following syntax.

```
Get-Queue [-Filter <Filter> -Server <ServerIdentity> -
Include <Internal | External | Empty | DeliveryType> -
Exclude <Internal | External | Empty | DeliveryType>]
```

This example displays basic information about all non-empty queues on the Exchange 2013 Mailbox server named Mailbox01.

```
Get-Queue -Server Mailbox01 -Exclude Empty
```

This example displays detailed information for all queues that contain more than 100 messages on the Mailbox server on which the command is run.

```
Get-Queue -Filter {MessageCount -gt 100} | Format-List
```

Use the Shell to view queue summary information on multiple Exchange servers

The **Get-QueueDigest** cmdlet provides a high-level, aggregate view of the state of queues on all servers within a specific scope, for example, a DAG, an Active Directory site, a list of servers, or the entire Active Directory forest. Note that queues on a subscribed Edge Transport server in the perimeter network aren't included in the results. Also, **Get-QueueDigest** is available on an Edge Transport server, but the results are restricted to queues on the Edge Transport server.

Note:

By default, the **Get-QueueDigest** cmdlet displays delivery queues that contain ten or more messages, and the results are between one and two minutes old. For instructions on how to change these default values, see Configure Get-QueueDigest.

To view summary information about queues on multiple Exchange servers, run the following command:

```
Get-QueueDigest <-Server
<ServerIdentity1,ServerIdentity2,..> | -Dag
<DagIdentity1,DagIdentity2...> | -Site
<ADSiteIdentity1,ADSiteIdentity2...> | -Forest> [-Filter
<Filter>]
```

This example displays summary information about the queues on all Exchange 2013 Mailbox servers in the Active Directory site named FirstSite where the message count is greater than 100.

```
Get-QueueDigest -Site FirstSite -Filter {MessageCount -gt
100}
```

This example displays summary information about the queues on all Exchange 2013 Mailbox servers in the database availability group (DAG) named DAG01 where the queue status has the value **Retry**.

```
Get-QueueDigest -Dag DAG01 -Filter {Status -eq "Retry"}
```

Resume queues

By resuming a queue, you restart outgoing activities on a queue that has a status of Suspended. The queue must have a status of Suspended for this action to have any effect. When you resume a

queue, the status of messages in the queue doesn't change. Messages that have a status of Suspended remain suspended and don't leave the queue.

Use Queue Viewer in the Exchange Toolbox to resume queues

1. Click **Start** > **All Programs** > **Microsoft Exchange 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer** to open the tool in a new window.
3. In Queue Viewer, click the **Queues** tab. A list of all queues on the server to which you're connected is displayed.
4. Click **Create Filter**, and enter your filter expression as follows:
   a. Select **Status** from the queue property drop-down list.
   b. Select **Equals** from the comparison operator drop-down list.
   c. Select **Suspended** from the value drop-down list.
5. Click **Apply Filter**. All queues on the server that are currently suspended are displayed.
6. Select one or more queues from the list, right-click, and then select **Resume**.

Use the Shell to resume queues

To resume queues, use the following syntax.

```
Resume-Queue <-Identity QueueIdentity | -Filter
{QueueFilter} [-Server ServerIdentity]>
```

This example resumes all queues on the local server that have a status of Suspended.

```
Resume-Queue -Filter {Status -eq "Suspended"}
```

This example resumes the suspended delivery queue named contoso.com on the server named Mailbox01.

```
Resume-Queue -Identity Mailbox01\contoso.com
```

How do you know this worked?

To verify that you have successfully resumed a queue, do the following:
1. Use the Queue Viewer or the **Get-Queue** cmdlet to find the queue you attempted to resume.
2. Verify the queue **Status** property doesn't have the value Suspended.

Retry queues

When a transport server can't connect to the next hop, the delivery queue is put in a status of Retry. When you retry a delivery queue by using Queue Viewer or the Shell, you force an immediate connection attempt and override the next scheduled retry time. If the connection isn't successful, the retry interval timer is reset. The delivery queue must be in a status of Retry for this action to have any effect.

Use Queue Viewer in the Exchange Toolbox to retry a queue

1. Click **Start** > **All Programs** > **Microsoft Exchange 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer** to open the tool in a new window.
3. In Queue Viewer, click the **Queues** tab. A list of all queues on the server to which you're connected is displayed.
4. Click **Create Filter**, and enter your filter expression as follows:
   a. Select **Status** from the queue property drop-down list.
   b. Select **Equals** from the comparison operator drop-down list.
   c. Select **Retry** from the value drop-down list.
5. Click **Apply Filter**. All queues that currently have a **Retry** status are displayed.
6. Select one or more queues from the list. Right-click, and then select **Retry Queue**. If the connection attempt is successful, the queue status changes to **Active**. If no connection can be made, the queue remains in a status of **Retry** and the next retry time is updated.

Use the Shell to retry a queue

To retry queues, use the following syntax.

```
Retry-Queue <-Identity QueueIdentity | -Filter QueueFilter
[-Server ServerIdentity]>
```

This example retries all queues on the local server with the status of Retry.

```
Retry-Queue -Filter {status -eq "retry"}
```

This example retries the queue named contoso.com that's in the `Retry` state on the server named Mailbox01.

```
Retry-Queue -Identity Mailbox01\contoso.com
```

How do you know this worked?

To verify that you have successfully retried a queue, do the following:
1. Use the Queue Viewer or the **Get-Queue** cmdlet to find the queue you attempted to retry.
2. Verify the queue **LastRetryTime** property matches the time you attempted to retry the queue.

Resubmit messages in queues

Resubmitting a queue is similar to retrying a queue, except the messages are sent back to the Submission queue for the categorizer to reprocess. You can resubmit messages that have the following status:

- Delivery queues that have the status of Retry. The messages in the queues can't be in the Suspended state.
- Messages in the Unreachable queue that aren't in the Suspended state.
- Messages in the poison message queue.

Use the Shell to resubmit messages

To resubmit messages, use the following syntax.

```
Retry-Queue <-Identity QueueIdentity | -Filter {Status -eq "Retry"} -Server ServerIdentity> -Resubmit $true
```

This example resubmits all messages located in any delivery queues with the status of Retry on the server named Mailbox01.

```
Retry-Queue -Filter {Status -eq "Retry"} -Server Mailbox01 -Resubmit $true
```

This example resubmits all messages located in the Unreachable queue on the server Mailbox01.

```
Retry-Queue -Identity Mailbox01\Unreachable -Resubmit $true
```

Resubmit messages in the poison message queue

You resubmit messages in the poison message queue by resuming the message. You can use the Queue Viewer or the Shell to resubmit messages from the poison message queue. Note that the poison message queue is only visible in Queue Viewer when there are messages in the poison message queue.

Note:

The poison message queue contains messages that are determined to be harmful to the Exchange system after a server failure. The messages may be genuinely harmful in their content or format. Alternatively, they may be victims of a poorly written agent that crashed the Exchange server while it was processing the supposedly bad messages. If you're unsure of the safety of the messages in the poison message queue, you should export the messages to files so that you can examine them. For more information, see Export messages from queues.

Use Queue Viewer in the Exchange Toolbox to resubmit messages in the poison message queue

1. Click **Start** > **All Programs** > **Microsoft Exchange 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer** to open the tool in a new window.
3. In Queue Viewer, click the **Queues** tab. A list of all queues on the server to which you're connected is displayed.
4. Click the poison message queue. In the action pane, select **View Messages**.
5. Select one or more messages from the list, right-click, and select **Resume**.

Use the Shell to resubmit messages in the poison message queue

To resubmit a message from the poison message queue, perform the following steps.

1. Find the identity of the message by running the following command.

```
Get-Message -Queue Poison | Format-Table Identity
```

2. Use the identity of the message from the previous step in the following command.

```
Resume-Message <PoisonMessageIdentity>
```

This example resumes a message from the poison message queue that has the message Identity value of 222.

```
Resume-Message 222
```

How do you know this worked?

To verify that you have successfully resubmitted a message from the poison message queue, do the following:

1. Use the Queue Viewer or the **Get-Queue** cmdlet to view the poison message queue where you attempted to resubmit the message.
2. Verify the message is no longer in the poison message queue. Note that an empty poison message queue doesn't appear in the Queue Viewer or the **Get-Queue** cmdlet. Therefore, if the message you resubmitted was the only message in the poison message queue, and the poison message queue is no longer visible, that is also an indication of a successful message resubmission.

Suspend queues

When you suspend a queue, you prevent messages from leaving the queue, but you don't change the status of messages in the queue. Messages that are in delivery through SMTP-send will finish operations. You can suspend a queue to stop mail flow, and then suspend one or more messages in the queue. When you resume the queue, the messages that were suspended won't leave the queue.

You can suspend a queue that has a status of Active or Retry. You can also suspend the Unreachable queue and the Submission queue.

If you suspend the Unreachable queue, items won't be resubmitted to the categorizer when configuration updates are received by the transport server until the queue is resumed. If you suspend the Submission queue, messages won't be picked up by the categorizer until the queue is resumed.

Use Queue Viewer in the Exchange Toolbox to suspend a queue

1. Click **Start** > **All Programs** > **Microsoft Exchange 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer** to open the tool in a new window.
3. In Queue Viewer, click the **Queues** tab. A list of all queues on the server to which you're connected is displayed. You can create a filter to display only queues that meet specific criteria.
4. Select one or more queues, right-click, and then select **Suspend**.

Use the Shell to suspend a queue

To suspend a queue, use the following syntax.

```
Suspend-Queue <-Identity QueueIdentity | -Filter
```

```
{QueueFilter} [-Server ServerIdentity]>
```

This example suspends all queues on the local server that have a message count equal to or greater than 1,000 and that have a status of Retry.

```
Suspend-Queue -Filter {MessageCount -ge 1000 -and Status -
eq "Retry"}
```

This example suspends the queue named contoso.com on the server named Mailbox01.

```
Suspend-Queue -Identity Mailbox01\contoso.com
```

How do you know this worked?

To verify that you have successfully suspended a queue, do the following:
1. Use the Queue Viewer or the **Get-Queue** cmdlet to find the queue you attempted to suspend.
2. Verify the queue **Status** property has the value Suspended.

Use the Exchange Management Shell to manage queues

Exchange Server 2013 > Mail flow > Queues >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-31*

As in previous versions of Exchange, you can use the Exchange Management Shell in Microsoft Exchange Server 2013 to view information about queues and the messages in those queues, and to perform management actions on queues and messages. In Exchange 2013, queues exist on Mailbox servers and Edge Transport servers. This topic refers to these servers as *transport servers*.

When you use the Shell to view and manage queues and messages in queues on transport servers, it's important to understand how to identify the queues or messages you want to manage. Typically, transport servers contain a large number of queues and messages to be delivered. You use the filtering parameters that are available on the queue and message management cmdlets to identify the queues or messages that you want to view or manage.

Note that you can also use Queue Viewer in the Exchange Toolbox to manage queues and messages in queues. However, the queue and message viewing cmdlets support more filterable properties and filter options than Queue Viewer. For more information about using Queue Viewer, see Queue Viewer.

Contents

- Queue filtering parameters
  - Queue identity
  - Queue Filter parameter
  - Include and Exclude parameters
- Get-QueueDigest
- Message filtering parameters
  - Message identity
  - Message Filter parameter
  - Queue parameter
- Comparison operators to use when filtering queues or messages
- Advanced paging parameters

Queue filtering parameters

The following table describes the filtering parameters that are available on the queue management cmdlets.

Cmdlet	Filtering parameters	Comments
Get-Queue	*Identity* *Filter* *Include* *Exclude*	You can't use the *Identity* parameter in the same command with the *Filter* parameters. You can use the *Include* and *Exclude* parameters with the *Filter* parameter in the same command.
Resume-Queue **Retry-Queue** **Suspend-Queue**	*Identity* *Filter*	You need to use either the *Identity* parameter or the *Filter* parameter, but you can't use both in the same command.
Get-QueueDigest	*Server* *Dag* *Site* *Forest* *Filter*	You need to use the *Server*, *Dag*, *Site*, or *Forest* parameter, but you can't use any of them together in the same command. You can use the *Filter* parameter with any of the

		other filtering parameters.

Note that a *Server* parameter is available on all queue management cmdlets. On the **Get-QueueDigest** cmdlet, the *Server* parameter is a scope parameter that specifies the server or servers where you want to view summary information about queues. On all other queue management cmdlets, you use the *Server* parameter to connect to a specific server, and run the queue management commands on that server. You can use the *Server* parameter with or without the *Filter* parameter, but you can't use the *Server* parameter with the *Identity* parameter. You use the transport server's hostname or FQDN with the *Server* parameter.

Return to top

Queue identity

The *Identity* parameter on the queue management cmdlets identifies a specific queue. When you use the *Identity* parameter, you can't specify any other queue filtering parameters, because you've already uniquely identified the queue. The *Identity* parameter uses the basic syntax <*Server*>\<*Queue*>.

The <*Server*> placeholder is the hostname or FQDN of the Exchange server, for example `mailbox01` or `mailbox01.contoso.com`. If you omit the <*Server*> qualifier, the local server is implied.

The <*Queue*> placeholder accepts one of the following values:

- **Persistent queue name** Persistent queues have unique, consistent names on all Mailbox or Edge Transport servers. The persistent queue names are:
  - **Submission** This queue contains messages waiting to be processed by the categorizer.
  - **Unreachable** This queue contains messages that can't be routed. This queue doesn't exist until messages are placed in it.
  - **Poison** This queue contains messages that are determined to be harmful to the Exchange server. This queue doesn't exist until messages are placed in it.
- **Delivery queue name** The name of a delivery queue is the value of the **NextHopDomain** property of the queue. For example, the queue name could be the address space of a Send connector, the name of an Active Directory site, or the name of a DAG. For more information, see the "NextHopSolutionKey" section in the Queues topic.
- **Queue integer** Delivery queues and shadow queues are assigned a unique integer value in the queue database. However, you need to run the **Get-Queue** cmdlet to find the integer value for the queue in the **Identity** or **QueueIdentity** properties.
- **Shadow queue name** A shadow queue uses the syntax `Shadow\`<*QueueInteger*>

The following table summarizes the syntax you can use with *Identity* parameter on the queue management cmdlets. In all values, <*Server*> is the hostname or FQDN of the server.

Queue identity formats

Identity parameter value	Description

`<Server>\<PersistentQueueName>` or `<PersistentQueueName>`	A persistent queue on the specified server or the local server. *<PersistentQueueName>* is `Submission`, `Unreachable`, or `Poison`.
`<Server>\<NextHopDomain>` or `<NextHopDomain>`	A delivery queue on the specified server or the local server. *<NextHopDomain>* is a routing destination or delivery group for the messages in the queue. For more information, see the "NextHopSolutionKey" section in the Queues topic.
`<Server>\<QueueInteger>` or `<QueueInteger>`	A delivery queue on the specified server or the local server. *<QueueInteger>* is the unique integer value of the queue that's displayed in the **Identity** property of the **Get-Queue** cmdlet.
`<Server>\Shadow\<QueueInteger>` or `Shadow\<QueueInteger>`	A shadow queue on the specified server or the local server.
`<Server>\*` or `*`	All queues on the specified server or the local server. Note that these values can only be used with the **Get-Queue** cmdlet.

Return to top

Queue Filter parameter

You can use the *Filter* parameter on the all of the queue management cmdlets to specify the queues you want to view or manage based on the properties of the queues. The *Filter* parameter creates an expression with comparison operators that restricts the queue operation to queues that meet the filter criteria. You can use the `-and` logical operator to specify multiple conditions that the results must match.

For a complete list of queue properties you can use with the *Filter* parameter, see Queues.

For a list of comparison operators you can use with the *Filter* parameter, see the Comparison operators to use when filtering queues or messages section in this topic.

For examples of procedures that use the *Filter* parameter to view and manage queues, see Manage queues.

Return to top

Include and Exclude parameters

Exchange 2013 has the *Include* and *Exclude* parameters available on the `Get-Queue` cmdlet. You can use these parameters individually, together, and with the *Filter* parameter to fine-tune your queue results on the local or specified transport server. For example, you can:

- Exclude empty queues from the results.
- Exclude queues to external destinations from the results.
- Include queues that have a specific value of **DeliveryType** in the results.

The *Include* and *Exclude* parameters use the following queue properties to filter queues:

Value	Description	Shell code example
`DeliveryType`	This value includes or excludes queues based on the **DeliveryType** property. You can specify multiple values separated by commas. The valid values for **DeliveryType** are explained in the "NextHopSolutionKey" section in the topic Queues topic.	This example returns all delivery queues on the local server where the next hop is a Send connector on the local server that's configured for smart host routing: `Get-Queue -Include SmartHostConnectorDelivery`
`Empty`	This value includes or excludes empty queues. Empty queues have the value `0` in the **MessageCount** property.	This example returns all queues on the local server that contain messages `Get-Queue -Exclude Empty`
`External`	This value includes or excludes queues that have the value `External` in the **NextHopCategory** property. External queues always have one of the following values for	This example returns all internal queues on the local server `Get-Queue -Exclude External`

	DeliveryType: • `DeliveryAgent` • `DnsConnectorDelivery` • `NonSmtpGatewayDelivery` • `SmartHostConnectorDelivery` For more information, see the "NextHopSolutionKey" section in the Queues topic.	
`Internal`	This value includes or excludes queues that have the value `Internal` in the **NextHopCategory** property. For more information, see the "NextHopSolutionKey" section in the Queues topic.	This example returns all internal queues on the local server. `Get-Queue -Include Internal`

Note that you can duplicate the functionality of the *Include* and *Exclude* parameters by using the *Filter* parameter. For example, the command `Get-Queue -Exclude Empty` yields the same result as `Get-Queue -Filter {MessageCount -gt 0}`. However, the syntax of the *Include* and *Exclude* parameters is simpler and easier to remember.

Return to top

Get-QueueDigest

Exchange 2013 adds a new queue cmdlet named **Get-QueueDigest**. This cmdlet allows you to view information about some or all of the queues in your Exchange organization by using a single command. Specifically, the **Get-QueueDigest** cmdlet allows you to view information about queues based on their location on servers, in DAGs, in Active Directory sites, or in the whole Active Directory forest. Note that queues on a subscribed Edge Transport server in the perimeter network aren't included in the results. Also, **Get-QueueDigest** is available on an Edge Transport server, but the results are restricted to queues on the Edge Transport server.

Note:

By default, the **Get-QueueDigest** cmdlet displays delivery queues that contain ten or more messages, and the results are between one and two minutes old. For instructions on how to change these default values, see Configure Get-QueueDigest.

The filtering and sorting parameters that are available with the **Get-QueueDigest** cmdlet are described in the following table.

Parameter	Description

Dag, *Server*, or *Site*	These parameters are mutually exclusive, and set the scope for the cmdlet. You need to specify one of these parameters or the *Forest* switch. Typically, you would use the name of the server, DAG or Active Directory site, but you can use any value that uniquely identifies the server, DAG, or site. You can specify multiple servers, DAGs, or sites separated by commas.
Forest	This switch is required if you aren't using the *Dag*, *Server*, or *Site* parameters. You don't specify a value with this switch. By using this switch, you get queues from all Exchange 2013 Mailbox servers in the Active Directory forest. You can't use the *Forest* switch to view queues in remote Active Directory forests.
DetailsLevel	This parameter accepts the values `None`, `Normal`, and `Verbose`. The default value is `Normal`. When you use the value `None`, the queue name is omitted from the **Details** column in the results.
Filter	This parameter allows you to filter queues based on the queue properties. You can use any of the filterable queue properties as described in the Queue filters topic.
GroupBy	This parameter groups the queue results. You can group the results by one of the following properties: • DeliveryType • LastError • NextHopCategory • NextHopDomain • NextHopKey

	• Status • ServerName By default, the results are grouped by `NextHopDomain`. For information about these queue properties, see Queue filters.
ResultSize	This parameter limits the queue results to the value you specify. The queues are sorted in descending order based on the number of messages in the queue, and grouped by the value specified by the *GroupBy* parameter. The default value is 1000. This means that by default, the command displays the top 1000 queues grouped by **NextHopDomain**, and sorted by the queues containing the most messages to the queues containing the least messages.
Timeout	The parameter specifies the number of seconds before the operation times out. The default value is `00:00:10` or 10 seconds.

This example returns all non-empty external queues on the Exchange 2013 Mailbox servers named Mailbox01,Mailbox02, and Mailbox03.

```
Get-QueueDigest -Server Mailbox01,Mailbox02,Mailbox03 -
Include External -Exclude Empty
```

Return to top

Message filtering parameters

The following table describes the filtering parameters that are available on the message management cmdlets.

Cmdlet	Filtering parameters	Comments
Get-Message	*Identity* *Filter*	All filtering parameters are mutually exclusive, and you can use them together in the same

	Queue	command.
Remove-Message **Resume-Message** **Suspend-Message**	*Identity* *Filter*	You need to use either the *Identity* parameter or the *Filter* parameter, but you can't use both in the same command.
Export-Message	*Identity*	The *Identity* parameter is required.

Note that a *Server* parameter is available on all message management cmdlets except for the **Export-Message** cmdlet. You use the *Server* parameter to connect to a specific server, and run the message management commands on that server. You can use the *Server* parameter with or without the *Filter* parameter, but you can't use the *Server* parameter with the *Identity* parameter. You use the transport server's hostname or FQDN with the *Server* parameter.

Return to top

Message identity

The *Identity* parameter on the message management cmdlets identifies a specific message in one or more queues. When you use the *Identity* parameter, you can't specify any other message filtering parameters, because you've already uniquely identified the message. The *Identity* parameter uses the basic syntax <*Server*>\<*Queue*>\<*MessageInteger*>.

The <*Server*> placeholder is the hostname or FQDN of the Exchange server, for example `mailbox01` or `mailbox01.contoso.com`. If you omit the <*Server*> qualifier, the local server is implied.

The <*Queue*> placeholder accepts the identity of the queue as described in the "Queue identity" section in this topic. For example, you can use the persistent queue name, the **NextHopDomain** value, or the unique integer value of the queue in the queue database.

The <*MessageInteger*> placeholder represents the unique integer value that's assigned to the message when it first enters the queue database on the server. If the message is sent to multiple recipients that require multiple queues, all copies of the message in all queues in the queue database have the same integer value. However, you need to run the **Get-Message** cmdlet to find the integer value for the message in the **Identity** or **MessageIdentity** properties.

The following table summarizes the syntax you can use with *Identity* parameter on the message management cmdlets. In all values, <*Server*> is the hostname or FQDN of the server.

Message identity formats

Identity parameter value	Description
`<Server>\<Queue>\<MessageInteger>` or	A message in a specific queue on the specified

`<Queue>\<MessageInteger>`	server or the local server. *<MessageInteger>* is the unique integer value of the message that's displayed in the **Identity** property of the **Get-Message** cmdlet. *<Queue>* represents one of the following values: • **Persistent queue name** The value `Submission`, `Unreachable`, or `Poison`. • **Delivery queue name** The value of the **NextHopDomain** property of the queue, which is effectively the name of the queue. This value could be a routing destination or a delivery group. For more information, see the "NextHopSolutionKey" section in the Queues topic. • **Queue integer** The unique integer value of the delivery queue or shadow queue that's displayed in the **Identity** property of the **Get-Message** or **Get-Queue** cmdlets. • **Shadow queue identity** The shadow queue identity uses the syntax `Shadow \<QueueInteger>`.
`<Server>\*\<MessageInteger>` or `*\<MessageInteger>` or `<MessageInteger>`	All copies of the message in all queues in the queue database on the specified server or the local server.

Return to top

Message Filter parameter

You can use the *Filter* parameter on the **Get-Message**, **Remove-Message**, **Resume-Message**, and **Suspend-Message** cmdlets to specify the messages you want to view or manage based on the properties of the messages. The *Filter* parameter creates an expression with comparison operators that restricts the message operation to messages that meet the filter criteria. You can use the `-and` logical operator to specify multiple conditions that the results must match.

For a complete list of message properties you can use with the *Filter* parameter, see Queues.

For a list of comparison operators you can use with the *Filter* parameter, see the Comparison operators to use when filtering queues or messages section in this topic.

For examples of procedures that use the *Filter* parameter to view and manage messages, see Manage queues.

Return to top

Queue parameter

The *Queue* parameter is used only with the **Get-Message** cmdlet. You can use this parameter to get all messages in a specific queue, or all messages from multiple queues by using the wildcard character (*).When you use the *Queue* parameter, use the queue identity format <*Server*>\<*Queue*> as described in the "Queue identity" section in this topic.

Return to top

Comparison operators to use when filtering queues or messages

When you create a queue or message filter expression by using the *Filter* parameter, you need to include an comparison operator for the property value to match. The following table shows the comparison operators that you can use in a filter expression and how each operator functions. For all operators, the values compared aren't case sensitive.

Comparison operators

Operator	Function	Shell code example

`-eq`	This operator is used to specify that the results must exactly match the property value that's supplied in the expression.	To display a list of all queues that have a status of Retry:
		`Get-Queue -Filter {Status -eq "Retry"}`
		To display a list of all messages that have a status of Retry:
		`Get-Message -Filter {Status -eq "Retry"}`

`-ne`	This operator is used to specify that the results shouldn't match the property value that's supplied in the expression.	To display a list of all queues that don't have a status of Active: `Get-Queue -Filter {Status -ne "Active"}`

		To display a list of all messages that don't have a status of Active: `Get-Message -Filter {Status -ne "Active"}`

`-gt`	This operator is used with properties where the value is expressed as an integer or date/time. The filter results only include queues or messages where the value of the specified property is greater than the value that's supplied in the expression.	To display a list of queues that currently contain more than 1,000 messages:
		`Get-Queue -Filter {MessageCount -gt 1000}`
		To display a list of messages that currently have a retry count that's more than 3:
		`Get-Message -Filter {RetryCount -gt 3}`

`-ge`	This operator is used with properties where the value is expressed as an integer or date/time. The filter results only include queues or messages where the value of the specified property is greater than or equal to the value that's supplied in the expression.	To display a list of queues that currently contain 1,000 or more messages:
		`Get-Queue -Filter {MessageCount -ge 1000}`
		To display a list of messages that currently have a retry count that's 3 or more:
		`Get-Message -Filter {RetryCount -ge 3}`

`-lt`	This operator is used with properties where the value is expressed as an integer or date/time. The filter results only include queues or messages where the value of the specified property is less than the value that's supplied	To display a list of queues that currently contain less than 1,000 messages:
		`Get-Queue -Filter {MessageCount -lt 1000}`
		To display a list of messages that have an SCL that's less than 6:
		`Get-Message -Filter {SCL -lt 6}`

	in the expression.	

`-le`	This operator is used with properties where the value is expressed as an integer or date/time. The filter results only include queues or messages where the value of the specified property is less than or equal to the value supplied in the expression.	To display a list of queues that currently contain 1,000 or fewer messages:
		`Get-Queue -Filter {MessageCount -le 1000}`
		To display a list of messages that have an SCL that's 6 or less:
		`Get-Message -Filter {SCL -le 6}`

`-like`	This operator is used with properties where the value is expressed as a text string. The filter results only include queues or messages where the value of the specified property contains the text string that's supplied in the expression. You can include the wildcard character (*) in a **-like** expression that's applied to a text string field, but not with a field that has the enumeration type.	To display a list of delivery queues that have a destination to any SMTP domain that ends in Contoso.com:
		`Get-Queue -Filter {Identity -like "*contoso.com"}`
		To display a list of messages that have a subject that contains the text "payday loan":
		`Get-Messages -Filter {Subject -like "*payday loan*"}`

You can specify a filter that evaluates multiple expressions by using the **-and** comparison operator. The queues or messages must meet all conditions of the filter to be included in the results.

This example displays a list of queues that have a destination to any SMTP domain name that ends in Contoso.com and that currently contain more than 500 messages.

```
Get-Queue -Filter {Identity -like "*contoso.com*" -and MessageCount -gt 500}
```

This example displays a list of messages that are sent from any email address in the contoso.com domain that have an SCL that's greater than 5.

```
Get-Message -Filter {FromAddress -like "*Contoso.com*" -and
SCL -gt 5}
```

Return to top

Advanced paging parameters

Depending on current mail flow, queries against queues and messages can return a large set of objects. You can use the advanced paging parameters to control how query results are retrieved and displayed.

When you use the Shell to view queues and the messages in the queues, your query retrieves one page of information at a time. The advanced paging parameters control the size of the result set and can also be used to sort the results. All advanced paging parameters are optional and can be combined with any one of the parameter sets that can be used with the **Get-Queue** and **Get-Message** cmdlets. If you don't specify any advanced paging parameters, the query returns the results in ascending order of identity.

By default, when a sort order is specified, the message identity property is always included and is sorted in an ascending order. This is the default ordering relationship. The message identity property is included because the other properties that can be included in a sort order aren't unique. By explicitly including the message identity property in the sort order, you can specify that the results display the message identity sorted in descending order.

You can use the *BookmarkIndex* and *BookmarkObject* parameters to mark a position in the sorted result set. If the bookmark object no longer exists when the next page of results is retrieved, the default ordering relationship makes sure that the result set starts with the closest object to the bookmark. The closest object depends on the specified sort order.

The following table describes the advanced paging parameters.

Advanced paging parameters

Parameter	Description
BookmarkIndex	This parameter specifies the position in the result set where the displayed results start. The value of this parameter is a 1-based index in the total result set. If the value is less than or equal to zero, the first complete page of results is returned. If the value is set to `Int.MaxValue`, the last complete page of results is returned.
BookmarkObject	This parameter specifies the object in the result

	set where the displayed results start. If you specify a bookmark object, that object is used as the point to start the search. The rows before or after that object, depending on the value of the *SearchForward* parameter, are retrieved. You can't combine the *BookmarkObject* parameter and the *BookmarkIndex* parameter in a single query.
IncludeBookmark	This parameter specifies whether to include the bookmark object in the result set. By default, the value is set to `$true` and the bookmark object is included. You may run a query for a limited result size, and then specify the last item in that result set as the bookmark for the next query. In this case, you may want to set *IncludeBookmark* to `$false` so that the object isn't included in both result sets.
ResultSize	This parameter specifies the number of results to display per page. If you don't specify a value, the default result size of 1,000 objects is used. Exchange limits the result set to 250,000.
ReturnPageInfo	This parameter is a hidden parameter. It returns information about the total number of results and the index of the first object of the current page. The default value is `$false`.
SearchForward	This parameter specifies whether to search forward or backward in the result set. This parameter doesn't affect the order in which the result set is returned. It determines the direction of search relative to the bookmark index or object. If no bookmark index or object

	is specified, the *SearchForward* parameter determines whether the search starts from the first or last object in the result set. The default value for this parameter is `$true`. If the this parameter is set to `$true` and a bookmark is specified, the query searches forward from that bookmark. If you use this configuration and there are no results beyond the bookmark, the query returns the last full page of results. If the *SearchForward* parameter is set to `$false` and a bookmark is specified, the query searches backward from that bookmark. If you use this configuration and there is less than a full page of results beyond the bookmark, the query returns the first full page of results.
SortOrder	This parameter specifies an array of message properties used to control the sort order of the result set. The sort order properties are specified in descending order of precedence. Each property is separated by a comma and appended with a plus sign (+) to sort in ascending order, or a minus sign (-) to sort in descending order. If an explicit sort order isn't specified by using this parameter, the records that match the query are displayed and sorted by the Identity field for the respective object type. The results are always sorted by identity in ascending order when a sort order isn't explicitly specified.

The following code example shows how to use the advanced paging parameters in a query. In this

example, the command connects to the specified server and retrieves a result set that contains 500 objects. The results are displayed in a sorted order, first in ascending order by sender address, and then in descending order of message size.

```
Get-Message -Server mailbox01.contoso.com -ResultSize 500 -SortOrder +FromAddress,-Size
```

If you want to view successive pages, you can set a bookmark for the last object retrieved in a result set and run an additional query. You need to use the scripting capabilities of the Shell to perform this procedure.

The following example uses scripting to retrieve the first page of results, sets the bookmark object, excludes the bookmark object from the result set, and then retrieves the next 500 objects on the specified server.

1. Open the Shell and type the following command to retrieve the first page of results.

```
$Results=Get-message -Server mailbox01.contoso.com -
ResultSize 500 -SortOrder +FromAddress,-Size
```

2. To set the bookmark object, type the following command to save the last element of the first page to a variable.

```
$temp=$results[$results.length-1]
```

3. To retrieve the next 500 objects on the specified server and to exclude the bookmark object, type the following command.

```
Get-message -Server mailbox01.contoso.com -
BookmarkObject:$temp -IncludeBookmark $False -ResultSize
500 -SortOrder +FromAddress,-Size
```

Return to top

Configure Get-QueueDigest

Topic Last Modified: *2013-10-18*

The **Get-QueueDigest** cmdlet allows you to view information about some or all of the queues in your Exchange organization by using a single command.

By default, the results returned by the **Get-QueueDigest** cmdlet are between one and two minutes old. These values are controlled by the following settings:

- **QueueLoggingInterval key in EdgeTransport.exe.config** This key specifies how frequently queue data is logged and is available to **Get-QueueDigest**. The default value is `00:01:00` (one minute). To specify a value, enter it as a time span: *hh:mm:ss* where *h* = hours, *m* = minutes, and *s*

= seconds. By default, this key isn't present in the EdgeTransport.exe.config file.

- **QueueDiagnosticsAggregationInterval parameter on Set-TransportConfig** This parameter specifies how frequently queue data is shared between Mailbox servers. The default value is `00:01:00` (one minute). To specify a value, enter it as a time span: *hh:mm:ss* where *h* = hours, *m* = minutes, and *s* = seconds.

The sum of the **QueueLoggingInterval** key and *QueueDiagnosticsAggregationInterval* parameter values determine the maximum age of the results returned by **Get-QueueDigest**.

Also, **Get-QueueDigest** returns results differently based on the type of queue and the status of the queue. For example, the following queues are displayed in the results as long as they contain at least one message:

- The Submission queue, the Unreachable queue, and the poison message queue (persistent queues).
- Delivery queues in the Suspended state (queues manually suspended by an administrator).

By default, delivery queues that have the status Active, Connecting, Ready, or Retry are returned in the results only if the queue contains 10 or more messages. This value is controlled by the **QueueLoggingThreshold** key in the EdgeTransport.exe.config file. You can specify a smaller or larger integer value. By default, this key isn't present in the EdgeTransport.exe.config file.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- To see the Exchange permissions you need to run **Set-TransportConfig** in the Exchange Management Shell, see the "Transport configuration" entry in the Mail flow permissions topic.
- Exchange permissions don't apply to modifying the EdgeTransport.exe.config file and restarting the Microsoft Exchange Transport service. These procedures are performed in the operating system of the Exchange Server.
- Changes you save to the EdgeTransport.exe.config file are applied after you restart the Microsoft Exchange Transport service. When you restart the service, mail flow on the server is temporarily interrupted.
- Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.
- Changes you make using **Set-TransportConfig** affect all Mailbox servers in your organization. Changes you make in the EdgeTransport.exe.config file affect the local Mailbox server only.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Configure Get-QueueDigest

1. In a Command Prompt window, open the EdgeTransport.exe.config file in Notepad by running the following command:

```
Notepad %ExchangeInstallPath%Bin\EdgeTransport.exe.config
```

2. Add one or both of the following keys in the `<appSettings>` section.

```
<add key="QueueLoggingThreshold" value="<integer>" />
<add key="QueueLoggingInterval" value="<hh:mm:ss>" />
```

For example, to set the **QueueLoggingThreshold** value to 1 and the **QueueLoggingInterval** value to 30 seconds, use the following values:

```
<add key="QueueLoggingThreshold" value="1" />
<add key="QueueLoggingInterval" value="00:00:30" />
```

3. When you are finished, save and close the EdgeTransport.exe.config file.
4. Restart the Microsoft Exchange Transport service by running the following command:

```
net stop MSExchangeTransport && net start
MSExchangeTransport
```

5. To change the value of the *QueueDiagnosticsAggregationInterval* parameter in the Exchange Management Shell, use the following syntax:

```
Set-TransportConfig -QueueDiagnosticsAggregationInterval
<hh:mm:ss>
```

For example, to change the value to 30 seconds, run the following command:

```
Set-TransportConfig -QueueDiagnosticsAggregationInterval
00:00:30
```

How do you know this worked?

To verify that you have successfully configured **Get-QueueDigest**, do the following:

1. Verify the values of the **QueueLoggingThreshold** and **QueueLoggingInterval** keys in the EdgeTransport.exe.config file. If the keys aren't present, the default values are used.
2. Verify the value of the *QueueDiagnosticsAggregationInterval* parameter by running the following command:

```
Get-TransportConfig | Format-List *queue*
```

Queue filters

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-31*

Filtering generates different views of queues. You use the queue properties as filter options. By specifying filter criteria, you can quickly locate queues and take action on them. The following scenarios are examples of how you might use queue filtering to manage mail flow:

- You receive a message from the Microsoft System Center Operations Manager that indicates that a queue length has exceeded the established threshold. You want to investigate whether a server-wide mail flow problem exists.

You can create a filter to view all the queues that have a message count that exceeds what you consider typical. If a mail flow problem is indicated, you can select all the queues in the filter results and suspend the queues while you continue to investigate.

- You suspend several queues to investigate the cause of mail flow problems. You determine that the problem was caused by an incorrect connector configuration and is now fixed.

You can create a filter to view all the queues that have a status of Suspended, and then select all the queues in the filter results and resume the queues.

Queue properties to use when filtering queues

You can use the queue properties to create a filter and locate queues that meet specified criteria. You can create filters in Queue Viewer, or by using the *Filter* parameter on the queue management cmdlets. Note that the queue management cmdlets support more filterable properties than the Queue Viewer. The following table lists the queue properties by which you can filter and the valid values for those properties.

Queue Viewer queue property	Shell queue property	Description
n/a	`DeferredMessageCount`	This property identifies the number of messages that were returned to the Submission queue because of transient errors that were encountered during recipient resolution. For more information about deferred messages, see

		Recipient resolution.
Delivery Type	`DeliveryType`	The valid values for **DeliveryType** are explained in the "NextHopSolutionKey" section in the Queues topic.
n/a	`Identity`	This property is the identity of the queue in the form of *<Server>**<Queue>*. For more information see the "Queue identity" section in the Queues topic.
n/a	`IncomingRate`	This property is a calculated number that indicates how quickly messages are entering the queue. For more information, see "IncomingRate, OutgoingRate, and Velocity" section in the Queues topic.
Last Error	`LastError`	This property indicates the text string of the last error that was recorded for the queue.
Last Retry Time	`LastRetryTime`	This property indicates the date/time of the last connection attempt for a queue that has a status of `Retry`.
n/a	`LockedMessageCount`	This property is reserved for internal Microsoft use, and isn't used in on-premises Exchange 2013 organizations.

Message Count	`MessageCount`	This property indicates the number of messages in the queue.
n/a	`NextHopCategory`	This property designates the next hop of the queue as `Internal` or `External` and is based on the value of the **DeliveryType** property of the queue. For more information, see the "NextHopSolutionKey" section in the Queues topic.
n/a	`NextHopConnector`	This property is the GUID of the next hop and is based on the value of the **DeliveryType** property of the queue. For more information, see the "NextHopSolutionKey" section in the Queues topic.
Next Hop Domain	`NextHopDomain`	This property is the name of next hop and is based on the value of the **DeliveryType** property of the queue . For more information, see the "NextHopSolutionKey" section in the Queues topic.
Next Retry Time	`NextRetryTime`	This property indicates the date/time of the next connection attempt for a queue that has a status of `Retry`.
n/a	`OutgoingRate`	This property is a calculated

		number that indicates how quickly messages are leaving the queue. For more information, see "IncomingRate, OutgoingRate, and Velocity" section in the Queues topic.
n/a	`RiskLevel`	This property is reserved for internal Microsoft use, and isn't used in on-premises Exchange 2013 organizations.
Status	`Status`	This property indicates the current queue status. A queue can have one of the following status values Active, Connecting, None, Suspended, Ready, or Retry. For more information, see the "Queue status" section in the Queues topic.
n/a	`TlsDomain`	This property contains the FQDN of the destination domain if the domain is configured for Domain Security.
n/a	`Velocity`	This property contains a calculated number that indicates how effectively the queue is draining. For more information, see "IncomingRate, OutgoingRate,

		and Velocity" section in the Queues topic.

Manage messages in queues

Exchange Server 2013 > Mail flow > Queues >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-07*

In Microsoft Exchange Server 2013, you can use the Queue Viewer in the Exchange Toolbox or the Exchange Management Shell to manage messages in queues. For more information about using the message management cmdlets in the Exchange Management Shell, see Use the Exchange Management Shell to manage queues.

What do you need to know before you begin?

- Estimated time to complete each procedure: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Remove messages from queues

A message that's being sent to multiple recipients might be located in more than one queue. To remove a message from more than one queue in a single operation, you need to use a filter. You can select whether to send a non-delivery report (NDR) when you remove messages from a queue. You can't remove a message from the Submission queue.

Use Queue Viewer in the Exchange Toolbox to remove messages

1. Click **Start** > **All Programs** > **Microsoft Exchange 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer** to open the tool in a new window.
3. In Queue Viewer, click the **Messages** tab. A list of all messages on the server to which you're

connected is displayed. To adjust the action to a single queue, click the **Queues** tab, double-click the queue name, and then click the *Server**Queue* tab that appears.

4. Select one or more messages from the list, right-click, and then select **Remove Messages (with NDR)** or **Remove Messages (without NDR)**. A dialog box appears that confirms the selected action and displays, **Do you want to continue?** Click **Yes**.
5. To remove all messages from a particular queue, click the **Queues** tab. Select a queue, right-click, and then select **Remove Messages (with NDR)** or **Remove Messages (without NDR)**. A dialog box appears that confirms the selected action and displays, **Do you want to continue?** Click **Yes**.

Note:

If you're working with a filtered list, the displayed page may not include all items in the filter. In this case, a prompt appears that displays: **This action will affect all items on this page. To expand the scope of this action to include all items in this filter, check the following box before you click OK.**

Use the Shell to remove messages

To remove messages from queues, use the following syntax.

```
Remove-Message <-Identity MessageIdentity | -Filter
{MessageFilter}> -WithNDR <$true | $false>
```

This example removes messages in the queues that have a subject of "Win Big" without sending an NDR.

```
Remove-Message -Filter {Subject -eq "Win Big"} -WithNDR
$false
```

This example removes the message with the message ID 3 from the unreachable queue on server named Mailbox01 and sends an NDR.

```
Remove-Message -Identity Mailbox01\Unreachable\3 -WithNDR
$true
```

How do you know this worked?

To verify that you have successfully removed messages from queues, do one of the following:

- In Queue Viewer, select the queue or create a filter to verify the messages no longer exist.
- Use the **Get-Message** cmdlet with the *Queue* or *Filter* parameters to verify the messages no longer exist. For more information, see Get-Message.

Resume messages in queues

You can resume a message that currently has a status of Suspended. By resuming a message, you enable delivery of the message. If you resume a message located in the poison message queue, the message will be sent to the categorizer for processing. A message being sent to multiple recipients might be located in multiple queues. To resume a message in more than one queue in a single

operation, you must use a filter.

Use Queue Viewer in the Exchange Toolbox to resume messages

1. Click **Start** > **All Programs** > **Microsoft Exchange 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer** to open the tool in a new window.
3. In Queue Viewer, click the **Messages** tab. A list of all messages on the server to which you're connected is displayed. To adjust the action to focus on a single queue, click the **Queues** tab, double-click the queue name, and then click the *Server\Queue* tab that appears.
4. Click **Create Filter**, and enter your filter expression as follows:
   a. Select **Status** from the message property drop-down list.
   b. Select **Equals** from the comparison operator drop-down list.
   c. Select **Suspended** from the value drop-down list.
5. Click **Apply Filter**. All messages that have a status of Suspended are displayed.
6. Select one or more messages from the list, right-click, and select **Resume**.

Use the Shell to resume messages

To resume messages, use the following syntax:

```
Resume-Message <-Identity MessageIdentity | -Filter {MessageFilter}>
```

This example resumes all messages being sent from any sender in the Contoso.com domain.

```
Resume-Message -Filter {FromAddress -eq "*contoso.com"}
```

This example resumes the message with the message ID 3 in the unreachable queue on server Hub01.

```
Resume-Message -Identity Hub01\Unreachable\3
```

To resubmit messages from the poison message queue, perform the following steps:

How do you know this worked?

To verify that you have successfully resume messages in queues, do one of the following:

- In Queue Viewer, select the queue or create a filter to verify the messages are no longer suspended.
- Use the **Get-Message** cmdlet with the *Queue* or *Filter* parameters to verify the messages are no longer suspended. For more information, see Get-Message.

Note that if you can't find the message in any queues on the server, that probably indicates the message was successfully delivered to the next hop.

Suspend messages in queues

When you suspend a message, you prevent delivery of the message. A message that appears in the queue but is already in delivery won't be suspended. Delivery will continue, and the message status will be **PendingSuspend**. If the delivery fails, the message will re-enter the queue, and the message will then be suspended. You can't suspend a message in the Submission queue or in the poison message queue.

A message being sent to multiple recipients might be located in multiple queues. To suspend a message in more than one queue in a single operation, you need to use a filter.

Use Queue Viewer in the Exchange Toolbox to suspend messages

1. Click **Start** > **All Programs** > **Microsoft Exchange 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer** to open the tool in a new window.
3. In Queue Viewer, click the **Messages** tab. A list of all messages on the server to which you're connected is displayed. To limit the view to a single queue, click the **Queues** tab, double-click the queue name, and then click the *Server**Queue* tab that appears.
4. Select one or more messages, right-click, and then select **Suspend**.

Use the Shell to suspend messages

To suspend messages, use the following syntax:

```
Suspend-Message <-Identity MessageIdentity | -Filter {MessageFilter}>
```

This example suspends all messages in the queues that are from any sender in the domain contoso.com.

```
Suspend-Message -Filter {FromAddress -eq "*contoso.com"}
```

This example suspends the message with the message ID 3 in the unreachable queue on server named Mailbox01:

```
Suspend-Message -Identity Mailbox01\Unreachable\3
```

How do you know this worked?

To verify that you have successfully suspended messages in queues, do one of the following:

- In Queue Viewer, select the queue or create a filter to verify messages are suspended.
- Use the **Get-Message** cmdlet with the *Queue* or *Filter* parameters to verify the messages are suspended. For more information, see Get-Message.

Queue Viewer

Exchange Server 2013 > Mail flow > Queues >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-25*

Queue Viewer is a Microsoft Management Console snap-in that's installed on a Mailbox server or the Edge Transport server. Queue Viewer is located in the **Mail flow tools** section of the Exchange Toolbox console. You can use this tool to view information about queues on a transport server and the messages that are present in those queues and to perform management actions on queues and mail items. Queue Viewer is useful for troubleshooting mail flow and identifying spam.

When you're using Queue Viewer to manage queues, consider the following:

- You must connect to a transport server. By default, Queue Viewer opens the queue database on the server where you opened Queue Viewer. However, you can also connect to different servers. For more information, see Connect to a server in Queue Viewer.
- The list of queues and messages can be large, depending on current mail flow, and the list of queues and messages changes when messages enter and leave the server. You can configure the options for Queue Viewer to control the interval at which the list of queues and messages is refreshed and the number of items displayed on each page. For more information, see Set Queue Viewer options.
- You can create a filter to display the specific set of queues or messages that you want to monitor. After you locate the queues and messages that you want to monitor, you can view the property information for these queues and messages. This information is helpful when you troubleshoot the cause of mail flow problems. For more information, see Queue filters and Message filters.
- You can use the **Export List** link in the action pane to export the list of queues or a list of messages. For more information, see Export lists from Queue Viewer.

Connect to a server in Queue Viewer

Mail flow > Queues > Queue Viewer >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

When you use Queue Viewer in the Exchange Toolbox on a Microsoft Exchange Server 2013 server that's located inside the Exchange organization, you can connect to other Mailbox servers. By default, when you open Queue View on a Mailbox server, Queue Viewer connects to the queue database on the local server. However, you can start more than one instance of Queue Viewer so that each instance focuses on a different server. You can also tile Queue Viewer windows so you can easily monitor more than one Mailbox server at a time.

You can also specify the server that Remote PowerShell uses to perform the specified tasks in Queue Viewer. This server doesn’t need to match the remote server you're managing in Queue Viewer.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.
- The procedures in this topic don't apply to Edge Transport servers. When you use Queue Viewer on an Edge Transport server, you can't change the focus of the tool.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Exchange Toolbox to specify the server you want to manage in Queue Viewer

1. Click **Start** > **All Programs** > **Microsoft Exchange Server 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer**.
3. In the action pane, click **Connect to server**.
4. In the **Connect to Server** window, click **Browse** to view a list of the available Mailbox servers.
5. In the **Select Exchange Server** window, select a Mailbox server. To search for a Mailbox server, use one of the following procedures:
   - Enter the exact server name or the first few letters of the server name in the **Search** field, and then click **Find Now**. Select a server from the result pane.
   - Select the **View** menu, and then click **Show Filter**. In the **Name** column or **Version** column, click the filter icon, and then select the filter operator. Type the filter criteria in the **Enter text here** field. Press ENTER. Select a server from the result pane.
6. Click **OK** to close the **Select Exchange Server** window.
7. After you select a server, in the **Connect to server** window, select the **Set as default server** check box if you want Queue Viewer to focus on this server first whenever Queue Viewer is opened.
8. In the **Connect to server** window, click **Connect**.

Use the Exchange Toolbox to specify the server that Queue Viewer uses to run Remote PowerShell

1. Click **Start** > **All Programs** > **Microsoft Exchange Server 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools section**, double-click **Queue Viewer**.

3. In the action pane, click **Properties**.
4. In the **Queue Viewer - <server name> Properties** dialog box, select one of the following options:
   - **Connect to the automatically selected server** Select this option to automatically connect to the server where you're managing queues to run Remote PowerShell.
   - **Specify a server to connect to** Select this option to specify a server to run Remote PowerShell. If you select this option, click **Browse** to open the **Select Exchange Server** dialog box. Select the server where you want to run Remote PowerShell, and then click **OK**.

How do you know this worked?

You should be able to manage the queues on the Mailbox server you specified.

Set Queue Viewer options

Mail flow > Queues > Queue Viewer >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

You can set options in Queue Viewer to adjust the number of items that are displayed on the page and adjust the auto-refresh interval. The auto-refresh interval determines how frequently the results in Queue Viewer are updated.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

⚠**Warning:**
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Exchange Toolbox to set Queue Viewer options

1. Click **Start** > **All Programs** > **Microsoft Exchange Server 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer**.
3. In Queue Viewer, click **View** > **Options** to configure the following settings in the **Queue Viewer Options** dialog box:

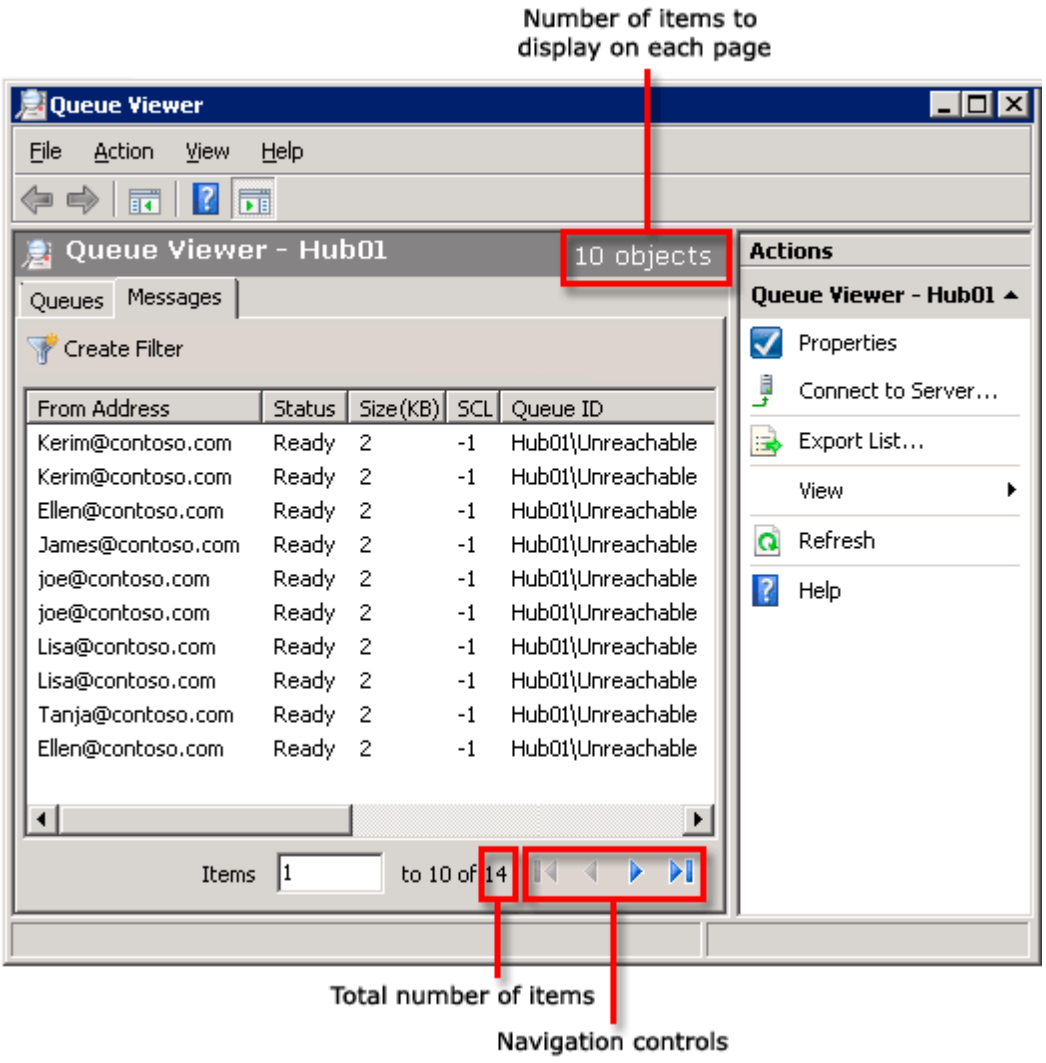
a. In the **Refresh interval (seconds)** field, enter the frequency at which Queue Viewer should update the display.

Note:
The default auto-refresh interval is 30 seconds and can't be set for a shorter time. If you disable auto-refresh functionality by clearing the **Auto-refresh screen** check box on the **Queue Viewer Options** page, you must manually update the results that are displayed in Queue Viewer by clicking **Refresh**.

b. In the **Number of items to display on each page** field, enter the maximum number of items to display in Queue Viewer. This number must be from 1 through 10,000. If you have more items than the limit, you will see the items in groups of the maximum number of items. For example, the following figure shows a queue with 14 messages with Queue Viewer configured to display 10 items on each page. The number of objects on the page is displayed on the upper right. At the bottom of the page, you can see the total number of items in the queue. You can use the navigation controls to see the additional items in the queue.

4. When you are finished, click **OK**.

Queue Viewer



How do you know this worked?

You'll know this procedure worked if Queue Viewer uses the refresh interval and number of items per page settings.

View queued message properties in Queue Viewer

Mail flow > Queues > Queue Viewer >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-17*

You can use the Queue Viewer in the Exchange Toolbox to view the properties of a message that is queued for delivery.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.
- You can also use the Get-Message cmdlet in the Exchange Management Shell to view additional message properties that aren't visible in Queue Viewer. For more information, see Message filters and Use the Exchange Management Shell to manage queues.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What Do You Want to Do?

Use Queue Viewer in the Exchange Toolbox to view the properties of a message

1. Click **Start** > **All Programs** > **Microsoft Exchange 2013** > **Exchange Toolbox**.
2. In the **Mail flow tools** section, double-click **Queue Viewer** to open the tool in a new window.
3. In Queue Viewer, select the **Messages** tab to see the list of messages that are currently queued for delivery in your organization.
4. Right-click the message whose properties you want to view and then select **Properties**.

5. The **General** tab displays the following detailed information about the message:
   - **Identity** This field shows the integer that represents a particular message. The message identity is assigned by the queuing database when the message is received for processing. You can include an optional server and queue identity to identify a unique instance of the message.
   - **Subject** This field shows the subject of a message and is expressed as a text string. The value is taken from the `Subject:` header field.
   - **Internet Message ID** This field shows the value of the `MessageID:` header field. The value of this property is expressed as a GUID followed by the SMTP address of the sending server, as in this example: 67D754D6103DC4FB3BA6BC7205DACABA61231@exchange.contoso.com
   - **From Address** This field shows the SMTP address of the sender of the message. This value is taken from MAIL FROM: in the message envelope.
   - **Status** This field shows the current message status. A message can have one of the following status values:
     - **Active** If the message is in a delivery queue, the message is being delivered to its destination. If the message is in the Submission queue, the message is being processed by the categorizer.
     - **Pending Remove** The message was deleted by the administrator but was already in delivery. The message will be deleted if the delivery ends in an error that causes the message to re-enter the queue. Otherwise, delivery will continue.
     - **Pending Suspend** The message was suspended by the administrator but was already in delivery. The message will be suspended if the delivery ends in an error that causes the message to re-enter the queue. Otherwise, delivery will continue.
     - **Ready** The message is waiting in the queue and is ready to be processed.
     - **Retry** The last connection attempt failed for the queue in which this message is located. The message is waiting for the next queue retry.
     - **Suspended** The message was suspended by the administrator.
   - **Size (KB)** This field shows the size of the message rounded up to the nearest kilobyte (KB).
   - **Message Source Name** This field shows the name of the component that submitted this message to the queue.
   - **Source IP** This field shows the IP address of the external server that submitted the message to the Exchange organization.
   - **SCL** This field shows the spam confidence Level (SCL) rating of the message. Valid SCL entries are integers 0 through 9 or -1. An empty SCL entry indicates that the message hasn't been processed by the Content Filter agent.
   - **Date Received** This field shows the date-time when the message was received by the server that holds the queue in which the message is located.
   - **Expiration Time** This field shows the date-time when the message will expire and will be deleted from the queue if the message can't be delivered.
   - **Last Error** This field shows the last error that was recorded for a message.
   - **Queue ID** This field shows the identity of the queue that holds the message. The queue identity is expressed in the form *Server**destination*, where *destination* is a delivery group, routing destination, persistent queue name, or the queue database identifier. The queue

database identifier is represented as an integer and can be determined by viewing the message properties.

- **Recipients** This field shows the list of recipients to which the message is addressed.
- **Retry Count** This field shows the number of times that delivery of a message to a destination was tried.

6. The **Recipient Information** tab displays the following information about the message recipients:
   - **Address** This field shows the SMTP address of the recipient of the message. This value is taken from `RCPT TO:` in the message envelope.
   - **Status** This field shows the current message status. A message can have one of the following status values:
     - **Active** If the message is in a delivery queue, the message is being delivered to its destination. If the message is in the Submission queue, the message is being processed by the categorizer.
     - **Pending Remove** The message has been deleted by the administrator but was already in delivery. The message will be deleted if the delivery ends in an error that causes the message to re-enter the queue. Otherwise, delivery will continue.
     - **Pending Suspend** The message has been suspended by the administrator but was already in delivery. The message will be suspended if the delivery ends in an error that causes the message to re-enter the queue. Otherwise, delivery will continue.
     - **Ready** The message is waiting in the queue and is ready to be processed.
     - **Retry** The last connection attempt failed for the queue in which this message is located. The message is waiting for the next queue retry.
     - **Suspended** The message has been suspended by the administrator.
   - **Last Error** This field shows the last error that was recorded for a message.

Use the Shell to view the properties of a message

You use the **Get-Message** cmdlet to view the properties of a message that is currently queued for delivery. The following example tabulates the sender address, recipients, subject, and received date information for all messages that are currently in retry state:

```
Get-Message -IncludeRecipientInfo -Filter {Status -eq
"Retry"} | Format-Table
FromAddress,Recipients,Subject,DateReceived
```

For detailed syntax and parameter information, see Get-Message.

Export lists from Queue Viewer

Mail flow > Queues > Queue Viewer >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-25*

This topic explains how to use Queue Viewer in the Exchange Toolbox to export lists of messages or queues.

You can export lists to the following file formats:

- Text (tab delimited)
- Text (comma delimited)
- Unicode text (tab delimited)
- Unicode text (comma delimited)

What do you need to know before you begin?

- Estimated time to complete: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Queues" entry in the Mail flow permissions.
- By default, the result pane in Queue Viewer displays only the first 1,000 objects. To change this value, see Set Queue Viewer options.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Export a list from the result pane in Queue Viewer

1. Click **Start** > **All Programs** > **Microsoft Exchange 2013** > **Exchange Toolbox**.
2. In the **Mail flow** section, double-click **Queue Viewer**.
3. In Queue Viewer, select the **Queues** tab or the **Messages** tab. On either tab, you can click **Create Filter** to restrict the results.

Note:

If the result pane doesn't refresh, in the action pane, click **Refresh**. Long lists may take several minutes to refresh.

4. In the action pane, click **Export List**. The **Export List** dialog box appears.
5. In **Export List**, type the name of the file in the **File name** box, and then select the file format from the **Save as type** list.
6. Click **Save**.

Message filters

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-14*

Filtering generates different views of the messages in queues. By specifying filter criteria, you can quickly locate messages and take action on them. When an email message is sent to multiple recipients, the message may be located in multiple queues. When you filter by message properties, you can locate messages across all queues. The following scenarios are examples of how you might use message filtering to manage mail flow:

- The Submission queue on the Mailbox server or Edge Transport server that receives email from the Internet has a high volume of messages that are queued for delivery. Many of the messages have the same subject. Therefore, you suspect that spam is being sent to your organization. You can create a filter to view all the messages that meet the subject criteria. If you determine that the messages are spam, you can select them all and delete them from the delivery queue without sending an NDR.
- A user reports that mail flow is slow. You examine the queues and see that many messages that have random subjects appear to be coming from a single domain. You can create a filter to view all the queued messages from that domain. If you determine that the messages are spam, you can select them all and delete them from the queues without sending an NDR.

Message properties as filters

You can use message properties to create a filter and locate messages that meet specified criteria. You can create filters in Queue Viewer, or by using the *Filter* parameter on the message management cmdlets. Note that the message management cmdlets support more filterable properties than the Queue Viewer. The following table lists the message properties by which you can filter and the values that are associated with those properties.

Message properties to use when filtering messages

Queue Viewer message property	Shell message property	Description
Date Received	`DateReceived`	The date/time when the message was placed in the queue.
n/a	`DeferReason`	This property identifies why the message was deferred. If the message wasn't deferred, this property has the value `None`. A deferred message is returned

		to the Submission queue because of transient errors that were encountered during recipient resolution. For more information about deferred messages, see Recipient resolution. The possible values are:
		`AD Transient Failure During Content Conversion`
		`AD Transient Failure During Resolve`
		`Agent`
		`Ambiguous Recipient`
		`Loop Detected`
		`Marked As Retry Delivery If Rejected`
		`Rerouted By Store Driver`
		`Storage Transient Failure During Content Conversion`
		`Transient Failure`
		`Target Site Inbound Mail Disabled`
		`Recipient Thread Limit Exceeded`
		`Transient Attribution Failure`
		`Transient Accepted Domains Load Failure`

Expiration Time	`ExpirationTime`	This property contains the date/time when the message will expire and be deleted from the queue if the message can't be delivered.
From Address	`FromAddress`	This property contains the SMTP address of the sender.
n/a	`Identity`	This property is the identity of the message in the form of *<Server>\<Queue>*

		\<*MessageInteger*>. For more information see the "Message identity" section in the Queues topic.
Internet Message ID	`InternetMessageId`	This property contains the value of the `Message-ID:` header field that's located in the message header. The value is expressed as an email address that contains a GUID and the FQDN the sending SMTP server. For example: `<67D754D6103DC4FB3BA6BC7205DACABA61231@mailbox01.contoso.com>`
Last Error	`LastError`	This property contains the text of the last error that was recorded for a message. For example, `A matching connector cannot be found to route the external recipient`.
n/a	`MessageLatency`	This property contains the amount of time elapsed between when the message first entered the Submission queue on the server, and when the message was placed in the queue. The value uses the syntax *hh:mm:ss.ff*, where *hh* = hour, *mm* = minute, *ss* = second, and *ff* = fractions of a second.

Message Source Name	`MessageSourceName`	This property contains a text string that indicates the name of the transport component that submitted the message to the queue. For example, if the message came in through a Receive connector, the value is: `SMTP:`<*ConnectorName*>. If the message is a delivery status notification (DSN), the value is `DSN`.
n/a	`Priority`	This property contains the priority of the message that's assigned by the user in Outlook or Outlook Web App. The possible values are `Low`, `Normal`, and `High`. For more information, see Priority queuing.
Queue ID	`Queue`	This property is the identity of the queue that holds the message. The queue identity uses the syntax <*Server*>\<*Queue*>. For more information, see the "Queue identity" section in the Queues topic.
n/a	`RetryCount`	This property identifies the number of times that delivery of the message to the destination was tried, either

		automatically or manually.
SCL	`SCL`	The value of the spam confidence level (SCL) property specifies the SCL of the message. Valid SCL entries are integers from 0 through 9. An empty SCL property value indicates that the message hasn't been processed by the Content Filter agent. This property contains the spam confidence level (SCL) value of the message. Valid SCL entries are integers from 0 through 9 and -1. For more information, see Spam Confidence Level Threshold.
Size (KB)	`Size`	This property indicates the size of the message.
Source IP	`SourceIP`	This property contains the IP address of the server that submitted the message to the Exchange server that holds the message in the queue. The address could be the IP address of a remote SMTP server, or the IP address of the local Exchange server.
Status	`Status`	This property indicates the current message status. A message can have one of the

		following status values:
		`Active`
		`Locked`
		`None`
		`Pending Remove`
		`Pending Suspend`
		`Ready`
		`Retry`
		`Suspended`
		For more information, see the "Message properties" section in the Queues topic.

Subject	`Subject`	This property indicates the subject of a message that's found in the `Subject:` header field in the message header.

Export messages from queues

Exchange Server 2013 > Mail flow > Queues >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

When you export a message from a queue to a file, the message isn't removed from the queue. A copy of the message is made in the specified location as a plain text file. The resulting file can be viewed in an application, such as a text editor or an email client application, or the message file can be resubmitted by using the Replay directory on any other Mailbox server or Edge Transport server inside or outside the Exchange organization.

What do you need to know before you begin?

- Estimated time to complete each procedure: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Queues" entry in Mail flow permissions topic.
- The messages must be in a Suspended state for the export process to be successful. You can export messages from delivery queues, the Unreachable queue, or the poison message queue. Messages in the poison message queue are already in the Suspended state. You can't suspend or

export messages that are in the Submission queue.

- You can't use Queue Viewer in the Exchange Toolbox to export messages. However, you can use Queue Viewer to locate, identify, and suspend the messages before you export them using the Shell.
- Verify the following information about the target directory location for the message files:
  - The target directory must exist before you export any messages. The directory won't be created for you. If an absolute path isn't specified, the current Exchange Management Shell working directory is used.
  - The path may be local to the Exchange server, or it may be a Universal Naming Convention (UNC) path to a share on a remote server.
  - Your account must have the **Write** permission to the target directory.
  - When you specify a file name for the exported messages, be sure to include the .eml file name extension so the files can be opened easily by email client applications or processed correctly by the Replay directory.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to export a specific message from a specific queue

To export a specific message from a specific queue, run the following command:

```
Export-Message -Identity <MessageIdentity> |
AssembleMessage -Path <FilePath>\<FileName>.eml
```

This example exports a copy of a message that has the **InternalMessageID** value 1234 that's located in the contoso.com delivery queue on the server named Mailbox01 to the file named export.eml in the path D:\Contoso Export.

```
Export-Message -Identity Exchange01\Contoso.com\1234 |
AssembleMessage -Path "D:\Contoso Export\export.eml"
```

Use the Shell to export all messages from a specific queue

To export all messages from a specific queue and use the **InternetMessageID** value of each message as the file name, use the following syntax.

```
Get-Message -Queue <QueueIdentity> | ForEach-Object
{$Temp=<Path>+$_.InternetMessageID
+".eml";$Temp=$Temp.Replace("<","_");$Temp=$Temp.Replace(">
","_");Export-Message $_.Identity | AssembleMessage -Path
$Temp}
```

Note that the **InternetMessageID** value contains angled brackets (> and <), which need to be removed because they aren't allowed in file names.

This example exports a copy of all the messages from the contoso.com delivery queue on the server named Mailbox01 to the local directory named D:\Contoso Export.

```
Get-Message -Queue Mailbox01\Contoso.com | ForEach-Object
{$Temp="D:\Contoso Export\"+$_.InternetMessageID
+".eml";$Temp=$Temp.Replace("<","_");$Temp=$Temp.Replace(">
","_");Export-Message $_.Identity | AssembleMessage -Path
$Temp}
```

Use the Shell to export specific messages from all the queues on a server

To export specific messages from all queues on a server and use the **InternetMessageID** value of each message as the file name, use the following syntax.

```
Get-Message -Filter {<MessageFilter>} [-Server
<ServerIdentity>] | ForEach-Object {$Temp=<Path>
+$_.InternetMessageID
+".eml";$Temp=$Temp.Replace("<","_");$Temp=$Temp.Replace(">
","_");Export-Message $_.Identity | AssembleMessage -Path
$Temp}
```

Note that the **InternetMessageID** value contains angled brackets (> and <), which need to be removed because they aren't allowed in file names.

This example exports a copy of all the messages from senders in the contoso.com domain from all queues on the server named Mailbox01 to the local directory named D:\Contoso Export.

```
Get-Message -Filter {FromAddress -like "*@contoso.com"} -
Server Mailbox01 | ForEach-Object {$Temp="D:\Contoso Export
\"+$_.InternetMessageID
+".eml";$Temp=$Temp.Replace("<","_");$Temp=$Temp.Replace(">
```

```
","_");Export-Message $_.Identity | AssembleMessage -Path
$Temp}
```

Note:
If you omit the *Server* parameter, the command operates on the local server.

Message retry, resubmit, and expiration intervals

Exchange Server 2013 > Mail flow > Queues >

Topic Last Modified: *2013-02-21*

In Microsoft Exchange Server 2013, messages that can't be successfully delivered are subject to various retry, resubmit, and expiration deadlines based on the message's source and destination. *Retry* is a renewed connection attempt with the destination. *Resubmit* is the act of sending messages back to the Submission queue for the categorizer to reprocess. The message timeout*expires* after all delivery efforts have failed over a specified period of time. After a message expires, the sender is notified of the delivery failure. Then the message is deleted from the queue.

In all three cases of retry, resubmit, or expire, you can manually intervene before the automatic actions are performed on the messages.

For instructions on how to configure these intervals, see Configure message retry, resubmit, and expiration intervals.

Configuration options for message retry

When a transport server can't connect to the next hop, the queue is put in a status of Retry. Connection attempts continue until the queue expires or a connection is made.

Configuration Options for Automatic Message Retry

The configuration options that are available for message retry intervals are described in the following table.

Configuration options that are available for message retry intervals

Parameter or key name	Default value	Where to configure	Description
QueueGlitchRetryCount	4	EdgeTransport.exe.config	This key specifies the number of connection

			attempts that are immediately tried when a transport server has trouble connecting with the destination server. Such connection problems are typically caused by very brief network outages. Valid input for this key is an integer from 0 through 15. Typically, you don't have to modify this key unless the network is unreliable and continues to experience many accidentally dropped connections.
QueueGlitchRetryInterval	`00:01:00` or 1 minute	EdgeTransport.exe.config	This key controls the connection interval between each connection attempt that's specified by the *QueueGlitchRetryCount* key. Typically, you don't have to modify this parameter unless the network is unreliable

			and continues to experience many accidentally dropped connections.
TransientFailureRetryCount	6	**Set-TransportService**cmdlet or server properties in the Exchange Administration Center (EAC)	This parameter specifies the number of connection attempts that are tried after the connection attempts that are controlled by the *QueueGlitchRetryCount* and *QueueGlitchRetryInterval* keys have failed. Connection problems that exhaust the *QueueGlitchRetryCount* and *QueueGlitchRetryInterval* keys can be caused by server restarts or cached DNS lookup failures. Valid input for this parameter is an integer from 0 through 15. If you set this parameter to 0, the next connection attempt is controlled by the

			OutboundConnectionFailureRetryInterval parameter.
TransientFailureRetryInterval	• Transport service on Mailbox servers: `00:05:00` or 5 minutes • Edge Transport servers: `00:01:00` or 10 minutes	**Set-TransportService**cmdlet or server properties in the EAC	This parameter controls the connection interval between each connection attempt that's specified by the *TransientFailureRetryCount* parameter. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
OutboundConnectionFailureRetryInterval	• Transport service on Mailbox servers: `00:10:00` or 10 minutes • Edge Transport Servers: `00:30:00` or 30 minutes	**Set-TransportService** cmdlet or server properties in the EAC	This parameter specifies the retry interval for outbound connection attempts that have previously failed. The previously failed connection attempts are controlled by the *TransientFailureRetryCount* and *TransientFailureRetryInterval* parameters. To specify a value,

			enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
MessageRetryInterval	`00:01:00` or 1 minute	**Set-TransportService** cmdlet	This parameter specifies the retry interval for individual messages that have a status of Retry. We recommend that you don't modify the default value unless Microsoft Customer Service and Support advises you to do this.
MailboxDeliveryQueueRetryInterval	`00:05:00` or 5 minutes	EdgeTransport.exe.config	This key specifies how frequently the queues try to connect to the Mailbox Transport Delivery service for a destination mailbox database that can't be successfully reached. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Valid input for this key

			is from 00:00:01 through 1.00:00:00.

Return to top

Configuration options for manual message retry

When a delivery queue is in the status of Retry, you can manually force an immediate connection attempt by using Queue Viewer in the Exchange toolbox or the **Retry-Queue** cmdlet in the Shell. The manual retry attempt overrides the next scheduled retry time. If the connection isn't successful, the retry interval timer is reset. The delivery queue must be in a status of Retry for this action to have any effect.

For more information, see the "Retry queues" section in Manage queues.

Return to top

Configuration options for delay DSN messages

After each message delivery failure, the Edge Transport server or the Transport service on the Mailbox server generates a delay delivery status notification (DSN) message and queues it for delivery to the sender of the undeliverable message. This delay DSN message is sent only after a specified delay notification timeout interval, and only if the failed message wasn't successfully delivered during that time. By default, the delay notification timeout interval is 4 hours. This delay prevents the sending of unnecessary delay DSN messages that may be caused by temporary message transmission failures. The sending of delay DSN notification messages can be selectively enabled or disabled for messages that originate inside or outside the Exchange organization.

The configuration options that are available for delay DSN notification messages are described in the following table.

Configuration options that are available for delay DSN notification messages

Parameter name	Default value	Location	Description
DelayNotificationTimeOut	`4:00:00`4 hours	**Set-TransportService** or server properties in the EAC	This parameter specifies how long the server waits before it sends a delay DSN message to the sender. The value of this parameter should always be greater than

			the value of the *TransientFailureRetryCount* parameter multiplied by the value of the *TransientFailureRetryInterval* parameter. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
ExternalDelayDSNEnabled	`$true`	**Set-TransportConfig**	This parameter specifies whether delay DSN messages can be sent to message senders who are outside the Exchange organization. Valid input for this parameter is `$true` or `$false`.
InternalDelayDSNEnabled	`$true`	**Set-TransportConfig**	This parameter specifies whether delay DSN messages can be sent to message senders who are inside the Exchange organization. Valid input for this

			parameter is `$true` or `$false`.

Note:

On Exchange 2007 Hub Transport servers, all *ExternalDSN** and *InternalDSN** parameters are available on the **Set-TransportServer** cmdlet, not the **Set-TransportConfig** cmdlet. If you have any Exchange 2007 Hub Transport servers in your organization, you need to make changes to these values using the **Set-TransportServer** cmdlet on each Exchange 2007 Hub Transport server.

Return to top

Configuration options for message resubmission

Message resubmission sends undelivered messages back to the Submission queue to be reprocessed by the categorizer.

Automatic message resubmission

Undelivered messages are automatically resubmitted if the delivery queue is in the status of Retry and has been unable to successfully deliver any messages for a specified period of time. That period of time is controlled by the *MaxIdleTimeBeforeResubmit* key in the EdgeTransport.exe.config application configuration file. Only messages in delivery queues are candidates for automatic resubmission.

To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.

The default value is `12:00:00` or 12 hours.

Manual Message Resubmission

You can manually resubmit messages that have the following status on a Hub Transport server or an Edge Transport server:

- Delivery queues that have the status of Retry. The messages in the queues must not be in the Suspended state.
- Messages that are in the Unreachable queue and aren't in the Suspended state.
- Messages that are in the poison message queue.

For more information about the poison message queue and the Unreachable queue, see "About the Poison Message Queue and the Unreachable Queue" in the topic Queues.

If you want to manually resubmit messages that are located in delivery queues or the Unreachable queue without waiting for the time that's specified by the *MaxIdleTimeBeforeResubmit* parameter to pass, you need to use the **Retry-Queue** cmdlet with the *Resubmit* parameter. To manually resubmit

messages that are located in the poison message queue, you can use Queue Viewer or the **Resume-Message** cmdlet to resume the message. For more information, see the "Resubmit messages in queues" section in Manage queues.

Another way that you can manually resubmit messages is to suspend the messages, export the messages to text files that have the .eml file name extension, and then copy the .eml files to the Replay directory on any Mailbox server or Edge Transport server. This resubmission method works for messages that are located in delivery queues or the Unreachable queue. Messages that are located in the poison message queue are already in the Suspended state. Messages that are located in the Submission queue can't be suspended or exported.

Note:
When you export messages from a queue, you don't remove the messages from the queue. After you export the messages and successfully resubmit them by using the Replay directory, you should remove the suspended messages to avoid duplicate message delivery.

For more information, see Export messages from queues.

Return to top

Configuration options for message expiration

The *message expiration timeout interval* specifies the maximum length of time that an Edge Transport server or a Hub Transport server tries to deliver a failed message. If the message can't be successfully delivered before the expiration timeout interval has passed, an NDR that contains the original message or the message headers is delivered to the sender.

Automatic message expiration

The message expiration timeout interval is controlled by the *MessageExpirationTimeOut* parameter in the **Set-TransportService** cmdlet or in the server properties in the EAC.

To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.

The default value is `2.00:00:00` or 2 days. The valid input range for this parameter is from `00:00:05` through `90.00:00:00`.

Manual Message Expiration

Although you can't manually force messages to expire, you can manually remove messages from any queue, except the Submission queue, with or without an NDR.

For more information, see the "Remove messages from queues" section in Manage messages in queues.

Return to top

Configure message retry, resubmit, and expiration intervals

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-20*

In Microsoft Exchange Server 2013, you can configure message retry, resubmit, and expiration intervals in the Transport service on Mailbox servers and on Edge Transport servers. For descriptions of these settings, see Message retry, resubmit, and expiration intervals.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport service" and "Edge Transport server" entries in the Mail flow permissions topic.
- Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use EdgeTransport.exe.config to configure the queue glitch retry count, the queue glitch retry interval, the mailbox delivery queue retry interval, and the maximum idle time before resubmit interval.

To configure the queue glitch retry count, the queue glitch retry interval, the mailbox delivery queue retry interval, and the maximum idle time before resubmit interval you modify keys in the %ExchangeInstallPath%Bin\EdgeTransport.exe.config XML application configuration file on the Mailbox server or Edge Transport server. Changes you save to this file are applied after you restart the Microsoft Exchange Transport service. When you restart this service, mail flow on the server is temporarily interrupted.

1. In a Command prompt window on the Mailbox server or Edge Transport server, open the EdgeTransport.exe.config file in Notepad by running the following command:

```
Notepad %ExchangeInstallPath%Bin\EdgeTransport.exe.config
```

2. Locate the following keys in the `<appSettings>` section.

```
<add key="QueueGlitchRetryCount" value="<Integer>" />
<add key="QueueGlitchRetryInterval" value="<hh:mm:ss>" />
<add key="MailboxDeliveryQueueRetryInterval"
value="<hh:mm:ss>" />
<add key="MaxIdleTimeBeforeResubmit" value="<hh:mm:ss>" />
```

This example changes the queue glitch retry count to 6, the queue glitch retry interval to 30 seconds, the mailbox delivery queue retry interval to 3 minutes, and the maximum idle time before resubmit interval to 6 hours.

```
<add key="QueueGlitchRetryCount" value="6" />
<add key="QueueGlitchRetryInterval" value="00:00:30" />
<add key="MailboxDeliveryQueueRetryInterval"
value="00:03:00" />
<add key="MaxIdleTimeBeforeResubmit" value="6:00:00" />
```

3. When you are finished save and close the EdgeTransport.exe.config file.
4. Restart the Microsoft Exchange Transport service by running the following command:

```
net stop MSExchangeTransport && net start
MSExchangeTransport
```

Configure the transient failure retry attempts, the transient failure retry interval, and the outbound connection failure retry interval

The transient failure retry attempts specifies the number of connection attempts that are tried after the connection attempts controlled by the `QueueGlitchRetryCount` and `QueueGlitchRetryInterval`

keys have failed. The default number of transient failure retry attempts is 6. The valid input range for this parameter is from 0 through 15. If you set the number of transient failure retry attempts to 0, the next connection attempt is controlled by the *outbound connection failure retry interval*.

The transient failure retry interval specifies the interval between each connection attempt that's specified by the number of transient failure retry attempts. In the Transport service on a Mailbox server, the default transient failure retry interval is 5 minutes. On an Edge Transport server, the default transient failure retry interval is 10 minutes.

The outbound connection failure retry interval specifies the retry interval for outgoing connection attempts that have previously failed. The previously failed connection attempts are controlled by the transient failure retry attempts and the transient failure retry interval. The default value for the outbound connection failure retry interval in the Transport service on a Mailbox server is 10 minutes. The default value on an Edge Transport server is 30 minutes.

Use the EAC to configure the transient failure retry attempts, the transient failure retry interval, or the outbound connection failure retry interval

1. In the Exchange admin center (EAC), click **Servers** > **Servers**, select the server, click **Edit** , and then click **Transport limits**.
2. In the **Retries** section, enter a value for **Outbound connection failure retry interval (seconds)**, the **Transient failure retry interval (minutes)**, or the **Transient failure retry attempts**.
3. When you are finished, click **Save**.

Use the Shell to configure the transient failure retry attempts, the transient failure retry interval, and the outbound connection failure retry interval

Use the following syntax to configure the transient failure retry attempts, the transient failure retry interval, and the outbound connection failure retry interval in the Transport service on a Mailbox server or on an Edge Transport server.

```
Set-TransportService <ServerIdentity> -
TransientFailureRetryCount <Integer> -
TransientFailureRetryInterval <hh:mm:ss> -
OutboundConnectionFailureRetryInterval <dd.hh:mm:ss>
```

This example changes the following values on the Mailbox server named Mailbox01: on the Edge Transport server Exchange01.

- The number of transient failure retry attempts is set to 8.
- The transient failure retry interval is set to 1 minute.
- The outbound connection failure retry interval is set to 45 minutes.

```
Set-TransportService Mailbox01 -TransientFailureRetryCount
8 -TransientFailureRetryInterval 00:01:00 -
OutboundConnectionFailureRetryInterval 00:45:00
```

Note:
The *TransientFailureRetryCount* and *TransientFailureRetryInterval* parameters are also available on the **Set-FrontEndTransportService** cmdlet for the Front End Transport service on Client Access servers.

Configure the transient failure retry attempts, the transient failure retry interval, and the outbound connection failure retry interval

Use the EAC to configure the transient failure retry attempts, the transient failure retry interval, and the outbound connection failure retry interval

1. In the EAC, click **Servers** > **Servers**, select the server, click **Edit** , and then click **Transport limits**.
2. In the **Retries** section, enter a value for **Outbound connection failure retry interval (seconds)**, the **Transient failure retry interval (minutes)**, or the **Transient failure retry attempts**.
3. When you are finished, click **Save**.

Use the Shell to configure the transient failure retry attempts, the transient failure retry interval, and the outbound connection failure retry interval

Use the following syntax to configure the transient failure retry attempts, the transient failure retry interval, and the outbound connection failure retry interval in the Transport service on a Mailbox server or on an Edge Transport server.

```
Set-TransportService <ServerIdentity> -
TransientFailureRetryCount <Integer> -
TransientFailureRetryInterval <hh:mm:ss> -
OutboundConnectionFailureRetryInterval <dd.hh:mm:ss>
```

This example changes the following values on the Mailbox server named Mailbox01: on the Edge Transport server Exchange01.

- The number of transient failure retry attempts is set to 8.
- The transient failure retry interval is set to 1 minute.
- The outbound connection failure retry interval is set to 45 minutes.

```
Set-TransportService Mailbox01 -TransientFailureRetryCount
8 -TransientFailureRetryInterval 00:01:00 -
OutboundConnectionFailureRetryInterval 00:45:00
```

Note:
The *TransientFailureRetryCount* and *TransientFailureRetryInterval* parameters are also available on the **Set-FrontEndTransportService** cmdlet for the Front End Transport service

on Client Access servers.

Use the Shell to configure the message retry interval

By default, the message retry interval is `00:01:00` or 1 minute. We recommend that you don't modify the default value unless Microsoft Customer Service and Support advises you to do this.

Use the following syntax to set the message retry interval.

```
Set-TransportService <ServerIdentity> -MessageRetryInterval
<dd.hh:mm:ss>
```

This example changes the message retry interval to 2 minutes on the Mailbox server named Mailbox01.

```
Set-TransportService Mailbox01 -MessageRetryInterval
00:02:00
```

Configure the delay DSN timeout settings

You can use the EAC or the Shell to configure the delay DSN notification timeout interval. This setting is applied to the local transport server only. You can only use the Shell to enable or disable the sending of delay DSN messages to internal and external senders. These setting are applied to all transport servers in your organization.

Note:

On Exchange 2007 Hub Transport servers, all *ExternalDSN** and *InternalDSN** parameters are available on the **Set-TransportServer** cmdlet, not the **Set-TransportConfig** cmdlet. If you have any Exchange 2007 Hub Transport servers in your organization, you need to make changes to these values using the **Set-TransportServer** cmdlet on each Exchange 2007 Hub Transport server.

Use the EAC to configure the delay DSN message notification timeout interval

1. In the EAC, click **Servers** > **Servers**, select the server, click **Edit** , and then click **Transport limits**.
2. In the **Notifications** section, enter a value for **Notify sender when message is delayed after (hours)**.
3. When you are finished, click **Save**.

Use the Shell to configure the delay DSN message notification timeout interval

Use the following syntax to set the message retry interval.

```
Set-TransportService <ServerIdentity> -
DelayNotificationTimeout <dd.hh:mm:ss>
```

This example changes the delay DSN message notification timeout interval to 6 hours on the

Mailbox server named Mailbox01.

```
Set-TransportService Mailbox01 -DelayNotificationTimeout
06:00:00
```

Use the Shell to enable or disable the sending of delay DSN notifications to external or internal message senders

Use the following syntax to configure the delay DSN notification settings.

```
Set-TransportConfig -ExternalDelayDSNEnabled <$true |
$false> -InternalDelayDSNEnabled <$true |$false>
```

This example prevents the sending of delay DSN notification messages to external senders.

```
Set-TransportConfig -ExternalDelayDSNEnabled $false
```

This example prevents the sending of delay DSN notification messages to internal senders.

```
Set-TransportConfig -InternalDelayDSNEnabled $false
```

Configure the message expiration timeout interval

Use the EAC to configure the message expiration timeout interval

1. In the EAC, click **Servers** > **Servers**, select the server, click **Edit** , and then click **Transport limits**.
2. In the **Message expiration** section, enter a value for **Maximum time since submission (days)**.
3. When you are finished, click **Save**.

Use the Shell to configure the message expiration timeout interval

To configure the message expiration timeout interval, use the following syntax.

```
Set-TransportService <ServerIdentity> -
MessageExpirationTimeout <dd.hh:mm:ss>
```

This example changes the message expiration timeout interval to 4 days on the Exchange server named Mailbox01.

```
Set-TransportService Mailbox01 -MessageExpirationTimeout
4.00:00:00
```

Priority queuing

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-17*

Priority queuing is a feature of Microsoft Exchange Server 2013 that enables the sender-defined priority of a message to influence the processing of the message by the Transport service on the Mailbox server.

The message priority is assigned by the sender in Microsoft Outlook when the sender creates and sends the message. The sender can set any of the following message priority values in Outlook:

- Low importance
- Normal importance
- High importance

The default priority for a message created in Outlook or Outlook Web App is Normal priority. The message priority is stored in the `X-Priority` header field in the message header.

Every message sent or received in an Exchange 2013 organization must be categorized by the Transport service on a Mailbox server before it can be routed and delivered. The categorizer in the Transport service on a Mailbox server picks up one message at a time from the Submission queue and performs recipient resolution, routing resolution, and content conversion on the message before putting the message in a delivery queue. For more information, see Mail flow.

Delivery queues are dynamically created based on the destination of a message. For more information, see Queues.

All messages that have the same destination are put in the same delivery queue. Priority queuing affects the transmission of messages from a delivery queue to the destination messaging server. When priority queuing is enabled, High priority messages are transmitted to their destinations before Normal priority messages, and Normal priority messages are transmitted to their destinations before Low priority messages. The prioritized delivery of messages based on the message priority can help you define specific service level agreement (SLA) requirements for message delivery times.

Options for configuring priority queuing

Support for priority queuing is controlled by keys in the `%ExchangeInstallPath%bin\EdgeTransport.exe.config` XML application configuration file. For instructions on how to configure priority queuing, see Enable and configure priority queuing.

The following table explains each key in more detail.

Priority queuing keys in the EdgeTransport.exe.config file

Key	Default value	Description
PriorityQueuingEnabled	`false`	This key enables or disables

		priority queuing in the Transport service on the Mailbox server. Valid input for this key is `true` or `false`. When this key is `false`, priority queuing is disabled, and all the priority queuing message limits that exist in the EdgeTransport.exe.config file are ignored.
MaxHighPriorityMessageSize	`250KB`	This key specifies the maximum allowed size of a High priority message. If a High priority message is larger than the value specified by this key, the message is automatically downgraded from High priority to Normal priority. The value of this key should be significantly less than the value of the *MaxSendMessageSize* parameter on the **Set-TransportConfig** cmdlet. The default value of this parameter is `10 MB`. The difference in these two values helps ensure consistent and predictable delivery times for High priority messages. When you enter a value, qualify the value with one of the following units: • KB (kilobytes)

		• MB (megabytes)
LowPriorityDelayNotificationTimeout *NormalPriorityDelayNotificationTimeout* *HighPriorityDelayNotificationTimeout*	**Low** `8:00:00` (8 hours) **Normal** `4:00:00` (4 hours) **High** `00:30:00` (30 minutes)	These keys specify the timeout interval for delay delivery status notification (DSN) messages based on the message priority. After each message delivery failure, the Transport service generates a delay DSN message and queues it for delivery to the sender of the undeliverable message. This delay DSN message is sent only after a specified delay notification time-out interval, and only if the failed message wasn't successfully delivered during that time. This delay prevents the sending of unnecessary delay DSN messages that may be caused by temporary message transmission failures. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
LowPriorityMessageExpirationTimeout *NormalPriorityMessageExpirationTimeout* *HighPriorityMessageExpirationTimeout*	**Low** `2.00:00:00` (2 days) **Normal** `2.00:00:00` (2 days) **High** `8:00:00` (8 hours)	These keys specify the maximum length of time that the Transport service tries to deliver a failed message. If the message can't be successfully delivered before the expiration time-out interval has passed, a non-delivery report (NDR) that contains the original

		message or the message headers is delivered to the sender. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
MaxPerDomainLowPriorityConnections *MaxPerDomainNormalPriorityConnections* *MaxPerDomainHighPriorityConnections*	**Low** 2 **Normal** 15 **High** 3	These keys specify the maximum number of connections that the Transport service can have open to any single remote domain. The outgoing connections to remote domains occur by using the delivery queues and Send connectors that exist on the Mailbox server. The sum these three keys should be less than or equal to the value of the *MaxPerDomainOutboundConnections* parameter on the **Set-TransportService** cmdlet. The default value of this parameter is 20.

How priority queuing affects other message limits on Mailbox servers

All messages that pass through the Transport service are subject to a variety of message retry, resubmit, and expiration limits. For more information, see Message size limits.

Some message limits available in the **Set-TransportService** cmdlet have corresponding priority queuing message limits available in the EdgeTransport.exe.config application configuration file. The following table shows these corresponding message limits.

Message limits in the Set-TransportService cmdlet that correspond to priority queuing message limits in the EdgeTransport.exe.config file

Source	Parameter or key	Default value
Set-TransportService	*DelayNotificationTimeOut*	`4:00:00` (4 hours)
EdgeTransport.exe.config	*NormalPriorityDelayNotificationTimeout*	`4:00:00` (4 hours)
Set-TransportService	*MessageExpirationTimeOut*	`2.00:00:00` (2 days)
EdgeTransport.exe.config	*NormalPriorityMessageExpirationTimeout*	`2.00:00:00` (2 days)

When priority queuing is disabled, all the priority queuing message limits that exist in the EdgeTransport.exe.config configuration file are ignored. All the message limits on the **Set-TransportService** cmdlet apply to all messages that travel through the Transport service on the Mailbox server.

When priority queuing is enabled, the priority queuing message limits in the EdgeTransport.exe.config configuration file override the corresponding message limits in the **Set-TransportService** cmdlet. All other message limits in the **Set-TransportService** cmdlet still apply to Low priority, Normal priority, and High priority messages that travel through the Transport service on the Mailbox server.

User Settings for Priority Queuing

The **Set-Mailbox** cmdlet has the *DowngradeHighPriorityMessagesEnabled* parameter. The default value is `$false`. When this parameter is set to `$true`, any High priority messages sent from the mailbox are automatically downgraded to Normal priority.

Enable and configure priority queuing

Mail flow > Queues > Priority queuing >

Topic Last Modified: *2013-02-22*

Priority queuing is a feature of Microsoft Exchange Server 2013 that enables the message priority that's configured by the sender in Microsoft Outlook or Outlook Web Access to influence the processing of the message by the Transport service on the Mailbox server. When priority queuing is enabled, High priority messages are transmitted to their destinations before Normal priority

messages, and Normal priority messages are transmitted to their destinations before Low priority messages. For more information, see Priority queuing.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- Exchange permissions don't apply to the procedures in this topic. These procedures are performed in the operating system of the Exchange Server.
- Changes you save to the EdgeTransport.exe.config application configuration file are applied after you restart the Microsoft Exchange Transport service.
- When you restart the Microsoft Exchange Transport service, mail flow on the server is temporarily interrupted.
- Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Command Prompt to enable and configure priority queuing in the EdgeTransport.exe.config file

1. In a Command prompt window, open the EdgeTransport.exe.config application configuration file in Notepad by running the following command:

```
Notepad %ExchangeInstallPath%Bin\EdgeTransport.exe.config
```

2. Find the following keys in the `<appSettings>` section.

```
<add key="PriorityQueuingEnabled" value="false" />
<add key="MaxPerDomainHighPriorityConnections" value="3" />
<add key="MaxPerDomainNormalPriorityConnections" value="15"
/>
<add key="MaxPerDomainLowPriorityConnections" value="2" />
<add key="HighPriorityMessageExpirationTimeout"
value="8:00:00" />
<add key="NormalPriorityMessageExpirationTimeout"
```

```
value="2.00:00:00" />
<add key="LowPriorityMessageExpirationTimeout"
value="2.00:00:00" />
<add key="HighPriorityDelayNotificationTimeout"
value="00:30:00" />
<add key="NormalPriorityDelayNotificationTimeout"
value="4:00:00" />
<add key="LowPriorityDelayNotificationTimeout"
value="8:00:00" />
<add key="MaxHighPriorityMessageSize" value="250KB" />
```

To enable priority queuing in the Transport service on the Mailbox server, use the following value:

```
<add key="PriorityQueuingEnabled" value="true" />
```

Configure the remaining priority queuing values, or leave them at their default values.

3. When you are finished, save and close the EdgeTransport.exe.config file.
4. Restart the Microsoft Exchange Transport service by running the following command:

```
net stop MSExchangeTransport && net start
MSExchangeTransport
```

How do you know this worked?

To verify that you have successfully enabled and configured priority queuing, do the following:

1. Verify the **PriorityQueueinEnabled** key in the EdgeTransport.exe.config file has the value `"true"`.
2. Use Outlook to create a high priority test message that's larger than the value specified by the **MaxHighPriorityMessageSize** key, and verify the message arrives as a normal priority message.
3. Try to verify that higher priority messages arrive before lower priority messages sent to the same recipient or destination. You can try to use multiple mailboxes to send multiple, similar test messages with different priority values to the same recipient simultaneously.

Change the location of the queue database

Exchange Server 2013 > Mail flow > Queues >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-22*

A *queue* is a temporary holding location for messages that are waiting to enter the next stage of processing. Each queue represents a logical set of messages that a transport server processes in a specific order.

Like the previous versions of Exchange, Microsoft Exchange Server 2013 uses an Extensible Storage Engine (ESE) database for queue message storage. All the different queues are stored in a single ESE database. Queues exist only on Mailbox servers or on Edge Transport servers.

The location of the queue database and the queue database transaction logs is controlled by keys in the `%ExchangeInstallPath%Bin\EdgeTransport.exe.config` XML application configuration file. This file is associated with the Microsoft Exchange Transport service. The following table explains each key in more detail.

Key	Description
QueueDatabasePath	This key specifies the location of the queue database files. The files are: • Mail.que • Trn.chk The default location is `%ExchangeInstallPath%TransportRoles\data\Queue`.
QueueDatabaseLoggingPath	This key specifies the location of the queue database transaction log files. The files are: • Trn.log • Trntmp.log • Trn*nnn*.log • Trnres00001.jrs • Trnres00002.jrs • Temp.edb Note that Temp.edb is used to verify the queue database schema when the Microsoft Exchange Transport service starts. Although Temp.edb isn't a transaction log file, it's kept in the same location as the transaction log files. The default location is `%ExchangeInstallPath%TransportRoles\data\Queue`.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes.
- Exchange permissions don't apply to the procedures in this topic. These procedures are performed in the operating system of the Exchange Server.
- When you stop or restart the Microsoft Exchange Transport service, mail flow on the server is interrupted.
- When you change the location of the queue database or the transaction logs, the existing queue database and transaction log files aren't moved. A new queue database and new transaction logs are created at the new location. The existing files are left at the old location. However, they're no longer used. If you want to reuse the existing queue database or transaction log files at the new location, you must move the existing files to the new location after the Microsoft Exchange Transport service is stopped, but before the service is started.
- If the target folder for the queue database or transaction logs doesn't exist, it will be created for you if the parent folder has the following permissions applied to it:
  - Network Service: Full Control
  - System: Full Control
  - Administrators: Full Control
- Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the Command Prompt to create a new queue database and transaction logs in a new location

1. Create the folders where you want to keep the queue database and transaction logs. Make sure that the correct permissions are applied to the folders.
2. In a Command prompt window, open the EdgeTransport.exe.config file in Notepad by running the following command:

```
Notepad %ExchangeInstallPath%Bin\EdgeTransport.exe.config
```

3. Modify the following keys in the `<appSettings>` section.

```
<add key="QueueDatabasePath" value="<LocalPath>" />
```

```
<add key="QueueDatabaseLoggingPath" value="<LocalPath>" />
```

For example, to create a new queue database in D:\Queue\QueueDB and new transaction logs in D:\Queue\QueueLogs, use the following values:

```
<add key="QueueDatabasePath" value="D:\Queue\QueueDB" />
<add key="QueueDatabaseLoggingPath" value="D:\Queue
\QueueLogs" />
```

4. When you are finished, save and close the EdgeTransport.exe.config file.
5. Restart the Microsoft Exchange Transport service by running the following command:

```
net stop MSExchangeTransport && net start
MSExchangeTransport
```

How do you know this worked?

To verify that you successfully created a new queue database and new transaction logs at a new location, do the following:

1. Verify the new database files Mail.que and Trn.chk exist at the new location.
2. Verify the new transaction log files Trn.log, Trntmp.log, Trnres00001.jrs, Trnres00002.jrs, and Temp.edb files exist at the new location.
3. If you can delete the old queue database and transaction log files from the old location after the Microsoft Exchange Transport service has started, those files are no longer being used.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Return to top

Use the Command Prompt to move the existing queue database and transaction logs to a new location

Only disaster recovery scenarios where the Microsoft Exchange Transport service wasn't shut down correctly or a hard disk drive failure would require that you restore and relocate an existing queue database and its existing transaction logs.

Under ordinary circumstances, you shouldn't have to reuse existing transaction logs. An ordinary shutdown of the Microsoft Exchange Transport service writes all uncommitted transaction log entries to the queue database. And, circular logging is used, so transaction logs that contain previously committed database changes aren't preserved.

Use the following procedure to move the existing queue database and transaction logs at a new location:

1. Create the folders where you want to keep the queue database and transaction logs. Make sure that the correct permissions are applied to the folders.

2. In a Command prompt window, open the EdgeTransport.exe.config file in Notepad by running the following command:

```
Notepad %ExchangeInstallPath%Bin\EdgeTransport.exe.config
```

3. Modify the following keys in the `<appSettings>` section:

```
<add key="QueueDatabasePath" value="<LocalPath>" />
<add key="QueueDatabaseLoggingPath" value="<LocalPath>" />
```

For example, to change the location of the queue database to D:\Queue\QueueDB and the transaction logs to D:\Queue\QueueLogs, use the following values:

```
<add key="QueueDatabasePath" value="D:\Queue\QueueDB" />
<add key="QueueDatabaseLoggingPath" value="D:\Queue
\QueueLogs" />
```

4. When you are finished, save and close the EdgeTransport.exe.config file.
5. Stop the Microsoft Exchange Transport service by running the following command:

```
net stop MSExchangeTransport
```

6. Move the existing database files Mail.que and Trn.chk from the original location to the new location.
7. Move the existing transaction log files Trn.log, Trntmp.log, Trn*nnnnn*.log, Trnres00001.jrs, Trnres00002.jrs, and Temp.edb from the old location to the new location.
8. Start the Microsoft Exchange Transport service by running the following command:

```
net start MSExchangeTransport
```

How do you know this worked?

To verify that you successfully moved the existing queue database and transaction logs to the new location, do the following:

1. Verify the queue database files Mail.que and Trn.chk exist at the new location.
2. Verify the transaction log files Trn.log, Trntmp.log, Trnres00001.jrs, Trnres00002.jrs, and Temp.edb files exist at the new location.
3. Verify there are no queue database or transaction log files at the original location.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Return to top

Pickup directory and Replay directory

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-10*

By default, the Pickup and Replay directories exist on every Microsoft Exchange Server 2013 Mailbox server or Edge Transport server. Correctly formatted email message files that you copy to the Pickup or Replay directories are submitted for delivery. The Pickup directory is used by administrators for mail flow testing, or by applications that must create and submit their own messages. The Replay directory receives messages from foreign gateway servers and can also be used to resubmit messages that administrators export from the queues of Exchange servers.

Contents

Anatomy of an email message file

A standard SMTP email message consists of a *message envelope* and message content. The message envelope contains information required for transmitting and delivering the message. The message content contains message header fields (collectively called the *message header*) and the message body. The message envelope is described in RFC 2821, and the message header is described in RFC 2822.

When a sender composes an email message and submits it for delivery, the message contains the basic information required to comply with SMTP standards, such as a sender, a recipient, the date and time that the message was composed, an optional subject line, and an optional message body. This information is contained in the message itself and, by definition, is contained in the message header.

The sender's messaging server generates a message envelope for the message by using the sender and recipient information found in the message header and transmits the message to the Internet for delivery to the recipient's messaging server. Recipients never see the message envelope, because it's generated by the message transmission process and isn't actually part of the message.

Each server involved in the transmission of the message may insert message header fields related to the server's role in delivering the message or other application-specific message header fields into the message header. When the recipient opens the message by using an email client, the email client displays some of the more relevant information from the message header, such as the sender, the recipients, and the subject together with the message body.

Return to top

How the Pickup and Replay directories process messages

In Exchange 2013, the default location of the Pickup directory is `%ExchangeInstallPath%TransportRoles\Pickup`. The default location of the Replay directory is `%ExchangeInstallPath%TransportRoles\Replay`. A correctly formatted .eml message file copied to the Pickup or Replay directory is processed for submission in the following steps:

1. The Pickup and Replay directories are checked for new message files every five seconds. You can't modify this polling interval. You can adjust the rate of message file processing by using the *PickupDirectoryMaxMessagesPerMinute* parameter on the **Set-TransportService** cmdlet. This parameter affects the Pickup directory and the Replay directory. The default value is 100 messages per minute. Files that can't be opened are left in the Pickup directory and are reevaluated at the next poll.
2. Limits put on message files in the Pickup directory, such as the maximum header size and the maximum number of recipients, are checked. By default, the maximum header size is 64 kilobytes (KB), and the maximum number of recipients is 100. You change these limits by using the **Set-TransportService** cmdlet. These settings affect the Pickup directory only.
3. The file is renamed from <*filename*>.eml to <*filename*>.tmp. If the <*filename*>.tmp file already exists, the file is renamed as <*filename*><*datetime*>.tmp. If the file renaming fails, an event log error is generated, and the pickup process proceeds to the next file.
4. After the .tmp file is successfully converted into an email message, a **delete on close** command is issued to the .tmp file. The .tmp file appears to remain in the Pickup directory, but the file can't be opened.
5. After the message is successfully queued for delivery, a **close** command is issued, and the .tmp file is deleted from the Pickup directory. If the deletion fails, an event log error is generated. If the Microsoft Exchange Transport service is restarted when there are .tmp files in the Pickup directory, all .tmp files are renamed as .eml files and are reprocessed. This could lead to duplicate message transmission.

Return to top

Pickup directory message file requirements

A message file copied to the Pickup directory must meet the following requirements for successful delivery:

- The message file must be a text file that complies with the basic SMTP message format. MIME

message header fields and content are supported.

- The message file must have an .eml file name extension.
- At least one email address must exist in the `Sender` or `From` message header fields in the message header. If a single email address exists in both the `Sender` and `From` fields, the email address in the `From` field is used as the originator of the message in the message envelope.
- Only one email address can exist in the `Sender` field. Multiple email addresses aren't allowed. The `Sender` field is optional if only one email address exists in the `From` field.
- Multiple email addresses are allowed in the `From` field, but a single email address must also exist in the `Sender` field. The address in the `Sender` field is then used as the originator of the message in the message envelope.
- At least one email address must exist in the `To`, `Cc`, or `Bcc` fields.
- A blank line must exist between the message header and the message body.

This example shows a plain text message that uses acceptable formatting for the Pickup directory.

```
To: mary@contoso.com
From: bob@fabrikam.com
Subject: Message subject
This is the body of the message.
```

MIME content is also supported in Pickup directory message files. MIME defines a broad range of message content that includes languages that can't be represented in 7-bit ASCII text, HTML, and other multimedia content. A complete description of MIME and its requirements is beyond the scope of this topic. This example shows a simple MIME message that uses acceptable formatting for the Pickup directory.

```
To: mary@contoso.com
From: bob@fabrikam.com
Subject: Message subject
MIME-Version: 1.0
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
<HTML><BODY>
<TABLE>
<TR><TD>cell 1</TD><TD>cell 2</TD></TR>
<TR><TD>cell 3</TD><TD>cell 4</TD></TR>
</TABLE>
</BODY></HTML>
```

Return to top

Pickup directory message header modifications

The Pickup directory removes any of the following message header fields from the message header:

- `Received`
- `Resent-*`
- `Bcc`

Note:

Any email addresses found in the optional `Bcc` message header fields in the message header are correctly processed. After the `Bcc` recipients are promoted to invisible message envelope recipients, they are removed from the message header to protect their identity. If a message contains only `Bcc` recipients, the value of **Undisclosed Recipients** is added to the `To` field in the message header.

The Pickup directory adds its own `Received` header field to a message as part of the message submission process. The `Received` header field is applied in the following format.

```
Received: from localhost by Pickup with Microsoft SMTP
Server id <ExchangeServerVersion><datetime>
```

The Pickup directory modifies the following message header fields if they're missing or malformed:

- **Message-Id** If the `Message-Id` field is missing or empty, the Pickup directory adds a Message-Id field by using the format <*GUID*>@<*defaultdomain*>.
- **Date** If the `Date` field is missing or malformed, the Pickup directory adds the date and time of message processing by the Pickup directory.

Return to top

Replay directory message file requirements

The Replay directory is used to resubmit exported Exchange messages and to receive messages from foreign gateway servers. These messages are already formatted for the Replay directory. There is little or no need for administrators or applications to compose and submit new message files by using the Replay directory. The Pickup directory should be used to create and submit new message files.

The Replay directory messages make extensive use of *X-Headers*. X-Headers are user-defined, unofficial message header fields that exist in the message header. X-Headers aren't specifically mentioned in RFC 2822, but the use of an undefined message header field starting with "X-" has become an accepted way to add unofficial message header fields to a message. The Exchange-specific X-Headers used in the message files in the Replay directory can actually set delivery information that normally exists in the message envelope. This feature is required to preserve original message information when you use the Replay directory to process exported messages from another Exchange server.

A message file copied to the Replay directory must meet the following requirements for successful delivery:

- The message file must be a text file that complies with the basic SMTP message format. MIME message header fields and content are supported.
- The message file must have an .eml file name extension.
- X-Headers must occur before all regular header fields.
- A blank line must exist between the header fields and the message body.

The X-Headers described in the following list are required by messages in the Replay directory:

- **X-Sender** This X-Header replaces the `From` message header field requirement in a typical SMTP message. One `X-Sender` field that contains one email address must exist. The Replay directory ignores the `From` message header field if it's present, although the recipient's email client displays the value of the `From` message header field as the sender of the message. Other parameters usually exist in the `X-Sender` field, as shown in the following example.

```
X-Sender: <bob@fabrikam.com> BODY=7bit RET=HDRS
ENVID=12345ABCD auth=<someAuth>
```

Note:

These parameters are message envelope values that are ordinarily generated by the sending server. You may see parameters similar to this in exported message files.
`RET` specifies whether the whole message or only the headers should be returned to the sender if the message can't be delivered. `RET` can have a value of `HDRS` or `FULL`. `ENVID` is a message envelope identifier. `BODY` specifies the text encoding of the message. `auth` specifies an authentication mechanism to the messaging server as described in RFC 2554.

- **X-Receiver** This X-Header replaces the `To` message header field requirement in a typical SMTP message. At least one `X-Receiver` field that contains one email address must exist. Multiple `X-Receiver` fields are allowed for multiple recipients. The Replay directory ignores the `To` message header fields if they're present, although the recipient's email client displays the values of the `To` message header fields as the recipients of the message. Other optional parameters may exist in the `X-Receiver` fields, as shown in the following example.

```
X-Receiver: <mary@contoso.com> NOTIFY=NEVER
ORcpt=mary@contoso.com
```

Note:

These parameters are message envelope values that are ordinarily generated by the sending server. You may see parameters similar to this in exported message files. These parameters are related to delivery status notification (DSN) messages as described in RFC 1891.
`NOTIFY` can have a value of `NEVER`, `DELAY`, or `FAILURE`. `ORcpt` preserves the original recipient of the message.

The X-Headers described in the following list are optional for message files in the Replay directory:

- **X-CreatedBy** Used for header firewall functionality. If this X-Header exists, it must not be blank. If the `X-CreatedBy` field doesn't exist, it's added with a value of **Unspecified**. Typically, the value of this field is **MSExchange15**, but it also may contain the non-SMTP address space type set on a Send connector, such as **Notes**.

- **X-EndOfInjectedXHeaders** Size in bytes of all the X-Headers present. This X-Header may be used as a marker to indicate the last X-Header before the regular message header fields start.
- **X-ExtendedMessageProps** Extended message properties for the message.
- **X-HeloDomain** HELO/EHLO domain string presented during the initial SMTP protocol conversation.
- **X-Source** Used by Queue Viewer under the **MessageSourceName** column. If the value of this X-Header isn't specified, the value of **Replay** is used. Other possible values for this X-Header are **Smtp Receive Connector** and **Smtp Send Connector**.
- **X-SourceIPAddress** IP address of the sending server. This field is `0.0.0.0` if no IP address is specified.

This example shows a plain text message that uses acceptable formatting for the Replay directory.

```
X-Receiver: <mary@contoso.com> NOTIFY=NEVER
ORcpt=mary@contoso.com
X-Sender: <bob@fabrikam.com> BODY=7bit ENVID=12345AB
auth=<someAuth>
Subject: Optional message subject
This is the body of the message.
```

MIME content is also supported in Replay directory message files. MIME defines a broad range of message content that includes languages that can't be represented in 7-bit ASCII text, HTML, and other multimedia content. A complete description of MIME and its requirements is beyond the scope of this topic. This example shows a simple MIME message that uses acceptable formatting for the Replay directory.

```
X-Receiver: <mary@contoso.com> NOTIFY=NEVER
ORcpt=mary@contoso.com
X-Sender: <bob@fabrikam.com> BODY=7bit ENVID=12345ABCD
auth=<someAuth>
To: mary@contoso.com
From: bob@fabrikam.com
Subject: Optional message subject
MIME-Version: 1.0
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
<HTML><BODY>
<TABLE>
<TR><TD>cell 1</TD><TD>cell 2</TD></TR>
<TR><TD>cell 3</TD><TD>cell 4</TD></TR>
</TABLE>
</BODY></HTML>
```

Return to top

Replay directory message header modifications

The Replay directory deletes the `Bcc` message header field from the message file.

The Replay directory adds its own `Received` message header field to a message as part of the message submission process. The Received message header field is applied in the following format.

```
Received: from <ReceivingServerName> by Replay with
<ExchangeServerVersion><DateTime>
```

The Replay directory modifies the following message header fields in the message header:

- **Message-ID** If this message header field is missing or empty, the Replay directory adds a Message-ID message header field by using the format <*GUID*>@<*defaultdomain*>.
- **Date** If this message header field is missing or malformed, the Replay directory adds the Date message header field using the date and time of message processing by the Replay directory.

Return to top

Failures in Pickup and Replay directory message processing

A message file copied to the Pickup or Replay directories may not be successfully queued for delivery. The following categories of message submission failure can occur:

- **Delivery failures** A correctly formatted message file together with a valid sender that can't be successfully submitted for delivery generates a non-delivery report (NDR). Malformed content or Pickup directory message restriction violations could also cause an NDR. When an NDR is generated during message processing, the original message file is attached to the NDR message, and the message file is deleted from the Pickup directory or the Replay directory.

Note:

A correctly formatted message submitted into the transport pipeline may later experience a delivery failure and be returned to the sender with an NDR. This kind of failure may be caused by transmission issues unrelated to the Pickup or Replay directories, such as messaging server failures or routing failures along the delivery path of the message.

- **Badmail** A message classified as *badmail* has serious problems that prevent the Pickup or Replay directories from submitting the message for delivery. The other condition that causes badmail is when the message is formatted correctly, but the recipients aren't valid, and an NDR message can't be sent to the sender because the sender isn't valid.

Message files determined to be badmail are left in the Pickup or Replay directories and are renamed from <*filename*>.eml to <*filename*>.bad. If the <*filename*>.bad file already exists, the file

is renamed to <*filename*><*datetime*>.bad. If badmail exists in the Pickup or Replay directories, an event log error is generated, but the same badmail messages don't generate repeated event log errors.

Note:

Always compose and save message files in a different location before you copy them into the Pickup directory for delivery. The Pickup directory polls for new messages every five seconds. Therefore, if you try to compose and save the message files in the Pickup directory itself, the Pickup directory may try to process the message files before you finish composing them.

Return to top

Security considerations for the Pickup and Replay directories

The following list describes security concerns that are common to the Pickup directory and the Replay directory:

- Any security checks configured on a Receive connector, such as anti-spam, anti-malware, sender filtering, or recipient filtering actions, aren't performed on messages submitted through the Pickup directory or the Replay directory.
- A compromised Pickup directory or Replay directory can act as an open relay. This enables messages to be resubmitted or *relayed* by using a different server to mask the true source of the messages.

The following list describes additional security concerns that apply to the Replay directory:

- The X-Headers used by the Replay directory allow for the manual creation of the message envelope. The information in the `X-Sender` and `X-Receiver` fields can be completely different from the `To` or `From` message header fields displayed by email clients. Such an impersonation of a sender and a domain is frequently called *spoofing*. A *spoofed mail* is an email message that has a sending address that was modified to appear as if it originates from a sender other than the actual sender of the message.
- If the `X-CreatedBy` field has the value of **MSExchange15**, the destination is considered trustworthy, and header firewall isn't applied. *Header firewall* is a way for Exchange to preserve X-Headers in messages transmitted between trusted Exchange servers or to remove potentially revealing X-Headers from messages transmitted to untrusted destinations outside the Exchange organization. These X-Headers can be used to share Exchange information such as spam confidence level (SCL), message signing, or encryption between authorized Exchange servers. Revealing this information to unauthorized sources could pose a potential security risk. For more information about header firewall, see Understanding Header Firewall.

Tighter security should be applied to the Replay directory because of the additional security risks associated with the Replay directory. Users or applications that must generate and submit messages can be granted access to the Pickup directory, but they shouldn't require access to the Replay directory.

Both the Pickup directory and the Replay directory are enabled by default on all Mailbox servers and Edge Transport servers. If the Pickup directory or the Replay directory isn't required on a specific Mailbox server or Edge Transport server in your organization, you can disable the Pickup directory or the Replay directory on that server by setting the Pickup directory path or Replay directory path to the value `$null`. For more information, see Configure the Pickup directory and the Replay directory.

Return to top

Permissions for the Pickup and Replay directories

The following permissions are required on the Pickup and Replay directories:

- Administrator: Full Control
- System: Full Control
- Network Service: Read, Write, and Delete Subfolders and Files

By default, the Microsoft Exchange Transport service uses the security credentials of the Network Service user account to manage the location and permissions of the Pickup and Replay directories. The Network Service account requires these permissions on the Pickup directory so that .eml files can be opened, renamed to .tmp and deleted, or renamed to .bad if the message is classified as badmail.

You can move the location of these directories by using the *PickupDirectoryPath* and *ReplayDirectoryPath* parameters on the **Set-TransportService** cmdlet. Successfully changing the location of the Pickup directory depends on the rights granted to the Network Service account at the new directory locations, and whether the new directories already exist. If the directory doesn't exist, and the Network Service account has the rights required to create folders and apply permissions at the new location, the directory is created, and the correct permissions are applied to it. If the new directory already exists, the existing folder permissions aren't checked. Whenever you move the directory locations by using the *PickupDirectoryPath* or *ReplayDirectoryPath* parameter with the **Set-TransportService** cmdlet, always verify that the new directory exists and that the new directory has the correct permissions applied to it.

Return to top

Configure the Pickup directory and the Replay directory

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-04*

The Pickup and replay directories are used by the Transport service on Mailbox servers and Edge Transport servers to insert message files directly into the transport pipeline. Correctly formatted email message files that you copy to the Pickup or Replay directories are submitted for delivery. The Pickup directory is used by administrators for mail flow testing, or by applications that must create and submit their own messages. The Replay directory receives messages from non-SMTP foreign gateway servers and resubmits messages that you exported from the queues of Microsoft Exchange servers.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport service" entry in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.
- The value of the *PickupDirectoryMaxMessagesPerMinute* parameter on the **Set-TransportService** cmdlet is used by the Pickup and Replay directories.
- Changing the location of the Pickup or Replay directories doesn't copy any existing message files from the old location to the new location. The new Pickup or Replay directory location is active almost immediately after the configuration change, but any existing message files are left in the old location.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What Do You Want to Do?

Use the Shell to configure the Pickup directory

To configure the Pickup directory, use the following syntax.

```
Set-TransportService <ServerIdentity> -PickupDirectoryPath
<LocalFilePath> -PickupDirectoryMaxHeaderSize <Size> -
PickupDirectoryMaxRecipientsPerMessage <Integer> -
PickupDirectoryMaxMessagesPerMinute <Integer>
```

This example makes the following changes to the Pickup directory on the Mailbox server named Exchange01:

- The Pickup directory location is set to D:\Pickup Directory.
- The maximum size allowed for message headers in a message file is increased to 96 KB.
- The maximum number of recipients allowed in a message file is increased to 250.
- The maximum rate of message processing for the Pickup and Replay directories is increased to 200 messages per minute.

```
Set-TransportService Exchange01 -PickupDirectoryPath "D:
\Pickup Directory" -PickupDirectoryMaxHeaderSize 96KB -
PickupDirectoryMaxRecipientsPerMessage 250 -
PickupDirectoryMaxMessagesPerMinute 200
```

Note:

- Setting the *PickupDirectoryPath* parameter to the value `$null` disables the Pickup directory.
- The directory specified by the *PickupDirectoryPath* parameter and the *ReplayDirectoryPath* parameter can't be the same.

Use the Shell to configure the Replay directory

To configure the Replay directory, use the following syntax.

```
Set-TransportService <ServerIdentity> -ReplayDirectoryPath
"C:\Replay Directory" <LocalFilePath> -
PickupDirectoryMaxMessagesPerMinute <Integer>
```

This example makes the following changes to the Replay directory on the Mailbox server named Exchange01:

- The Replay directory location is set to D:\Replay Directory.
- The maximum rate of message processing for the Pickup and Replay directories is increased to 200 messages per minute.

```
Set-TransportService Exchange01 -ReplayDirectoryPath "D:
\Replay Directory" -PickupDirectoryMaxMessagesPerMinute 200
```

Note:

- Setting the *ReplayDirectoryPath* parameter to the value `$null` disables the Replay directory.
- The directory specified by the *PickupDirectoryPath* parameter and the *ReplayDirectoryPath* parameter can't be the same.

How do you know this worked?

To verify that you have successfully configured the Pickup and Replay directories, do the following:

1. Run the following command:

```
Get-TransportService <ServerIdentity> | Format-List
Pickup*,Replay*
```

2. Verify the values displayed are the values you configured.

TLS functionality and related terminology

Exchange Server 2013 > Mail flow >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-06-18*

Microsoft Exchange Server 2013 provides administrative functionality and other enhancements that improve the overall management of Transport Layer Security (TLS). As you work with this functionality, you need to learn about some TLS-related features and functionality. Some terms and concepts apply to more than one TLS-related feature. In this topic, a brief explanation of each feature is provided, which is intended to help you understand some differences and general terminology related to TLS and the Domain Security feature set:

- **Transport Layer Security** TLS is a standard protocol that's used to provide secure Web communications on the Internet or intranets. It enables clients to authenticate servers or, optionally, servers to authenticate clients. It also provides a secure channel by encrypting communications. TLS is the latest version of the Secure Sockets Layer (SSL) protocol.
- **Mutual TLS** Mutual TLS authentication differs from TLS as TLS is usually deployed. Typically, when TLS is deployed, it's used only to provide confidentiality in the form of encryption. No authentication occurs between the sender and receiver. Additionally, sometimes when TLS is deployed, only the receiving server is authenticated. This deployment of TLS is typical of the HTTP implementation of TLS. This implementation, where only the receiving server is authenticated, is SSL.

With mutual TLS authentication, each server verifies the identity of the other server by validating a certificate that's provided by that other server. In this scenario, where messages are received from external domains over verified connections in an Exchange 2013 environment, Microsoft Outlook displays a **Domain Secured** icon.

- **Domain Security** Domain Security is the set of features, such as certificate management, connector functionality, and Outlook client behavior that enables mutual TLS as a manageable and useful technology. Domain Security isn't supported when outbound email is routed through an Exchange 2013 Client Access server.
- **Opportunistic TLS** In Exchange 2013, Setup creates a self-signed certificate. By default, TLS is

enabled. This enables any sending system to encrypt the inbound SMTP session to Exchange. By default, Exchange 2013 also attempts TLS for all remote connections.

- **Direct trust** By default, all traffic between Edge Transport servers and Mailbox servers is authenticated and encrypted. Again, the underlying mechanism for authentication and encryption is mutual TLS. Instead of using X.509 validation, Exchange 2013 uses direct trust to authenticate the certificates. Direct trust means that the presence of the certificate in Active Directory or Active Directory Lightweight Directory Services (AD LDS) validates the certificate. Active Directory is considered a trusted storage mechanism. When direct trust is used, it doesn't matter if the certificate is self-signed or signed by a certification authority. When you subscribe an Edge Transport server to the Exchange organization, the Edge Subscription publishes the Edge Transport server certificate in Active Directory for the Mailbox servers to validate. The Microsoft Exchange EdgeSync service updates AD LDS with the set of Mailbox server certificates for the Edge Transport server to validate.

Scenario: Configure Exchange to support WAN Optimization Controllers

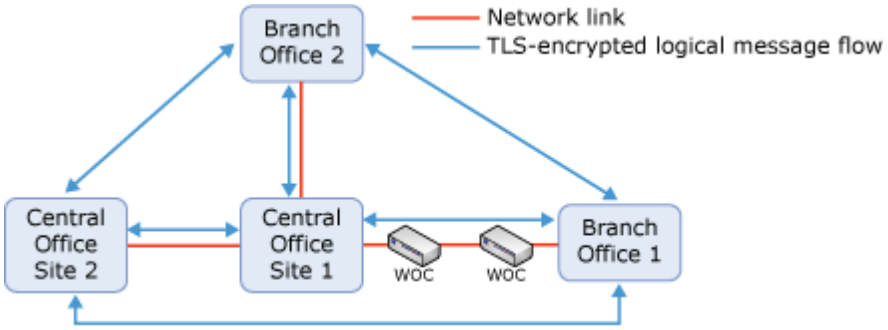
Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-28*

In Microsoft Exchange Server 2013, Transport Layer Security (TLS) encryption is mandatory for all SMTP communication in the Transport service between Mailbox servers. This increases overall security of Transport service communication between Mailbox servers. However, in certain topologies where WAN Optimization Controller (WOC) devices are used, the TLS encryption of SMTP traffic may be undesirable. You can disable TLS for Transport service communication between Mailbox servers for these specific scenarios.

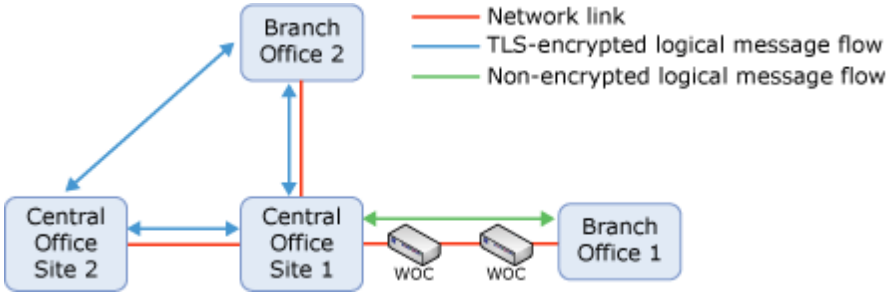
Consider the topology shown in the following figure. The assumption for this four-site topology is that the two central office sites and Branch Office 2 are well-connected, whereas the connection between Central Office Site 1 and Branch Office 1 is over a WAN link. The company has installed WOC devices on this link to compress and optimize traffic over the WAN.

Sample network topology with WOC devices



In this topology, because Exchange 2013 uses TLS encryption for communication between Mailbox servers, the SMTP traffic that goes over the WAN link can't be compressed. Ideally, all SMTP traffic that goes over the WAN link should be unencrypted SMTP, while retaining TLS security within well-connected sites. Exchange 2013 gives you the option to disable TLS encryption for traffic between sites by configuring Receive connectors. Using this ability in Exchange 2013, you can configure an exception to the SMTP traffic between Central Office Site 1 and Branch Office 1, as shown in the following figure.

Preferred logical message flow



The recommended configuration is to limit the non-encrypted SMTP traffic to only those messages that pass over the WAN link. Therefore, the intrasite Transport service communication between Mailbox servers in all sites, and the cross-site Transport service communication between Mailbox servers that don't involve Branch Office 1 should all be TLS encrypted.

To achieve this end result, you need to do the following actions, in the specified order, on every Mailbox server in the sites that contain the WOC devices (Central Office Site 1 and Branch Office 1 in the sample topology):

1. Enable downgraded Exchange Server authentication.
2. Create a dedicated Receive connector to handle the traffic over the connection that has WOC devices.
   a. Configure the remote IP address range property of the dedicated Receive connector to the IP address ranges of the Mailbox servers in the remote Active Directory site.
   b. Disable TLS on the dedicated Receive connector.

In addition, you need to do the following actions to ensure all SMTP traffic over the WAN is handled by the dedicated Receive connectors you created:

- Configure the Active Directory sites that will participate in the non-TLS communication as hub sites to force all message flow through the dedicated Receive connectors (Central Office Site 1 and Branch Office Site 1 in the sample topology).
- Verify that the Active Directory IP site link costs are configured in a way that ensures the least cost

routing path to your remote site (Branch Office 1 in the sample topology) goes through the network link that has the WOC devices. Assign an Exchange-specific cost to the Active Directory site links as necessary.

The following sections provide an overview of these steps. For step-by-step instructions on how to configure your organization for this scenario, see Disable TLS between Active Directory sites.

Contents

Downgrade authentication over TLS-disabled connections

Kerberos authentication is used with TLS encryption in Exchange. When you disable TLS on the Transport service communication between Mailbox servers, you need to perform another form of authentication. When Exchange 2013 communicates with other servers running Exchange that don't support **X-ANONYMOUSTLS**, it falls back to using Generic Security Services Application Programming Interface (GSSAPI) authentication. All Transport service communications between Exchange 2013 Mailbox servers use **X-ANONYMOUSTLS**. When you configure the Transport service on your Mailbox server to use downgraded Exchange Server authentication, you are in effect enabling GSSAPI authentication for Transport service communication with other Exchange 2013 Mailbox servers.

Return to top

Create and configure dedicated Receive connectors

You need to create Receive connectors that will solely be responsible for handling the non-TLS encrypted traffic. Using separate Receive connectors for this purpose ensures that all traffic that doesn't pass through the WAN link remains protected by TLS encryption.

To restrict the dedicated Receive connectors to only the traffic over the WAN, you need to configure the remote IP address range property. Exchange always uses the connector with the most specific remote IP address range. Therefore, these new connectors will be preferred over the default Receive connector configured to receive messages from anywhere.

Going back to the sample topology, assume that the class C subnet 10.0.1.0/24 is used for the Central Office Site 1 and 10.0.2.0/24 is used for the Branch Office 1. To prepare for disabling TLS between these two sites, you need to:

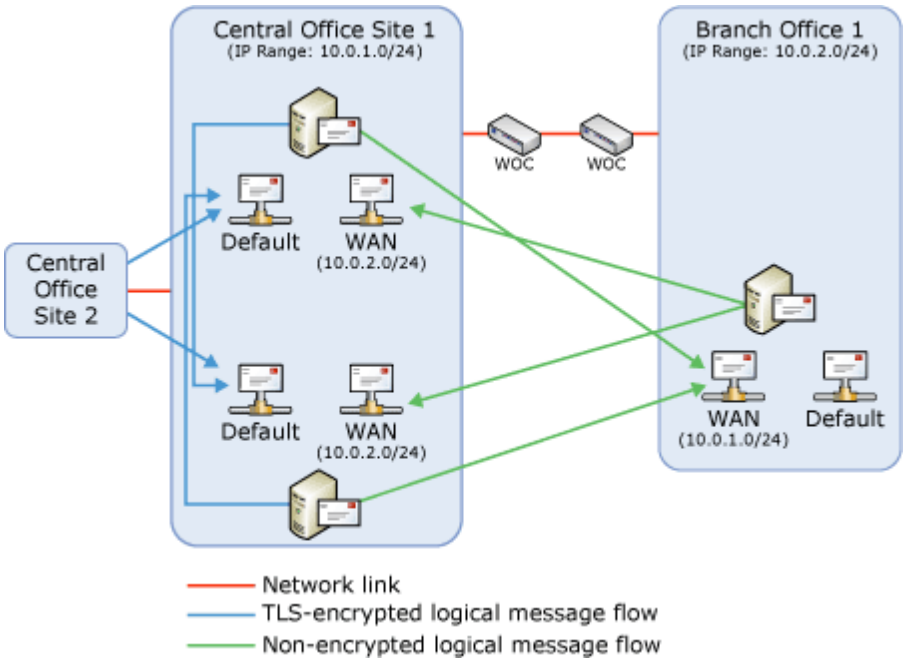
1. Create a Receive connector named WAN on each Mailbox server in Central Office Site 1 and Branch Office 1.
2. Configure the remote IP address range of 10.0.2.0/24 on each dedicated Receive connector in

Central Office Site 1.

3. Configure the remote IP address range of 10.0.1.0/24 on each dedicated Receive connector in Branch Office 1.
4. Disable TLS on all of the dedicated Receive connectors.

The end result is shown in the following figure (with the remote IP address range property of the Receive connectors named WAN shown in parentheses). Only a single Mailbox server is shown in Branch Office 1, and Branch Office 2 is omitted for clarity purposes.

Receive connector configuration



Return to top

Configure Hub sites

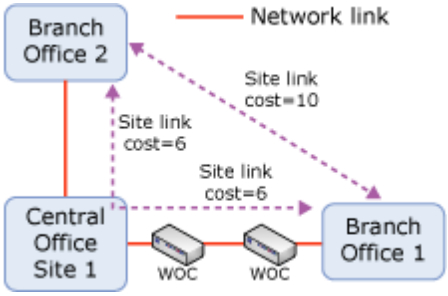
By default, an Exchange 2013 Mailbox server will attempt a direct connection to the Mailbox server closest to the final destination of a specific message. In the sample topology, if a user in Branch Office 2 sends a message to a user in Branch Office 1, the Mailbox server in Branch Office 2 will connect to the Mailbox server in Branch Office 1 directly to deliver that message. That connection will be encrypted and therefore not desirable in the specific topology. To have such messages pass through the Mailbox servers on Central Office Site 1, thereby ensuring they aren't encrypted while in transit over the WAN link, Central Office Site 1 and Branch Office 1 need to be configured as hub sites. In short, any site where you have a Mailbox server with a Receive connector with TLS disabled needs to be configured as a hub site, so you can force servers in other sites to route traffic through that site. For more information, see Configure Exchange mail routing settings in Active Directory.

Return to top

Configure Exchange-specific Active Directory site link costs

Configuring hub sites alone isn't sufficient to ensure that all traffic is unencrypted over the WAN link. This is because Exchange will route messages through hub sites only if the hub site lies within the least cost routing path. For example, assume that the IP site link costs for our sample topology are configured in Active Directory as shown in the following figure (Central Office Site 2 is omitted for clarity).

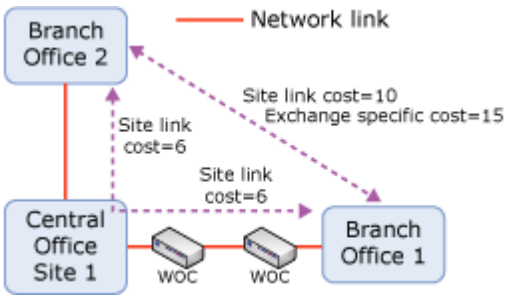
IP site link costs for the sample topology



In this case, the path from Branch Office 2 to Branch Office 1 that goes through the hub site has a total cost of 12 (6+6), whereas the cost of the direct path is 10. Therefore, none of the messages from Branch Office 2 to Branch Office 1 will go through Central Office Site 1 and therefore all of that traffic will still be TLS encrypted.

To avoid this issue, you need to designate an Exchange-specific cost that is higher than 12 for the IP site link between Branch Office 2 and Branch Office 1, as shown in the following figure. This will ensure that all messages go through the unencrypted channel between Central Office Site 1 and Branch Office 1.

Sample topology configured with Exchange-specific IP site link costs



For more information, see Configure Exchange mail routing settings in Active Directory.

Return to top

Disable TLS between Active Directory sites

Exchange Server 2013 > Mail flow > TLS functionality and related terminology >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-19*

Microsoft Exchange Server 2013 supports disabling TLS for SMTP communication between Mailbox servers in certain topologies where WAN Optimization Controller (WOC) devices that compress SMTP traffic are used.

This topic provides step-by-step instructions on how to configure the Transport service in your affected Mailbox servers to disable TLS, and to ensure your Active Directory routing topology is configured to correctly route messages. To learn more about this scenario, see Scenario: Configure Exchange to support WAN Optimization Controllers.

What do you need to know before you begin?

- Estimated time to complete this task: 60 minutes.
- Even though individual configuration steps within this scenario can be accomplished with lesser rights, to complete the entire end-to-end scenario tasks, your account needs to be a member of the Organization Management role group.
- Make sure you disable TLS only on connections that pass through WOC devices.
- This procedure requires that Exchange 2013 is deployed in multiple Active Directory sites, with at least one site connected to the other sites over a WAN link.
- This procedure requires that WOC devices are deployed to compress SMTP traffic over the WAN link.
- This procedure requires that a logical message flow path exists for Exchange going over the WAN link that has the WOC devices deployed.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Step 1: Use the Shell to configure the Transport service on the Mailbox server to use downgraded Exchange Server authentication

To configure the Transport service on a Mailbox server to use downgraded Exchange server authentication, run the following command:

```
Set-TransportService <ServerIdentity> -
```

```
UseDowngradedExchangeServerAuth $true
```

This example makes this configuration change on the server named Mailbox01.

```
Set-TransportService Mailbox01 -
UseDowngradedExchangeServerAuth $true
```

Step 2: Create a dedicated Receive connector on the Mailbox server for the target Active Directory site

Use the EAC to create the Receive connector

1. In the Exchange admin center (EAC), click **Mail flow** > **Receive connectors**, and then click **Add +** .
2. On the first page of the **New Receive connector** wizard, enter the following values
   - **Name** Enter a descriptive value.
   - **Type** Internal

When you are finished, click **Next**.

3. On the second page of the **New Receive connector** wizard, in the **Remote settings** section, enter the IP addresses or IP address ranges for the target Active Directory site. When you are finished, click **Finish**.

Use the Shell to create the Receive connector

To create a Receive connector on the Mailbox server, run the following command:

```
New-ReceiveConnector -Name <Name> -Server <ServerIdentity>
-RemoteIPRanges <IPAddressRange> -Internal
```

This example creates the Receive connector named WAN on server named Mailbox01 with the following settings:

- The *RemoteIPRanges* parameter is set to 10.0.2.0/24. This IP address range should correspond to the remote Active Directory site from where this Receive connector will receive unencrypted connections. If there's more than one IP subnet in the remote site, you can enter them all separated by commas.
- The usage type is set to Internal.

```
New-ReceiveConnector -Name WAN -Server Hub01 -
RemoteIPRanges 10.0.2.0/24 -Internal
```

Step 3: Use the Shell to disable TLS on the dedicated

Receive connector

To disable TLS on the Receive connector, run the following command:

```
Set-ReceiveConnector <ReceiveConnectorIdentity> -
SuppressXAnonymousTLS $true
```

This example disables TLS on the Receive connector named WAN on Mailbox server named Mailbox01.

```
Set-ReceiveConnector Mailbox01\WAN -SuppressXAnonymousTLS
$true
```

Step 4: Use the Shell to designate the Active Directory sites as hub sites

To designate an Active Directory site as a hub site, run the following command:

```
Set-AdSite <ADSiteIdentity> -HubSiteEnabled $true
```

You need to perform this procedure once in each Active Directory site that has Mailbox servers that participate in non-encrypted traffic.

This example configures the Active Directory site named Central Office Site 1 as a hub site.

```
Set-AdSite "Central Office Site 1" -HubSiteEnabled $true
```

Step 5: Use the Shell to configure the least cost routing path through the WAN connection

Depending on how the IP site link costs are configured in Active Directory, this step may not be necessary. You need to verify that the network link with the WOC devices deployed is in the leastcost routing path. To view the Active Directory site link costs, and the Exchange-specific site link costs, run the following command:

```
Get-AdSiteLink
```

If the network link with the WOC devices deployed isn't on the least cost routing path, you'll need to assign an Exchange-specific cost to the particular IP site link to ensure messages are routed correctly. To learn more about this particular issue, see the "Configure Exchange-specific Active Directory site link costs" section in Scenario: Configure Exchange to support WAN Optimization

Controllers.

This example configures an Exchange-specific cost of 15 on the IP site link named Branch Office 2-Branch Office 1.

```
Set-AdSiteLink "Branch Office 2-Branch Office 1" -
ExchangeCost 15
```

Recipients

Exchange Server 2013 >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-08-13*

The people and resources that send and receive messages are the core of any messaging and collaboration system. In an Exchange organization, these people and resources are referred to as *recipients*. A recipient is any mail-enabled object in Active Directory to which Microsoft Exchange can deliver or route messages.

Exchange recipient types

Exchange includes several explicit recipient types. Each recipient type is identified in the Exchange Administration Center (EAC) and has a unique value in the *RecipientTypeDetails* property in the Exchange Management Shell. The use of explicit recipient types has the following benefits:

- At a glance, you can differentiate between various recipient types.
- You can search and sort by each recipient type.
- You can more easily perform bulk management operations for selected recipient types.
- You can more easily view recipient properties because the EAC uses the recipient types to render different property pages. For example, the resource capacity is displayed for a room mailbox, but isn't present for a user mailbox.

The following table lists the available recipient types. All these recipient types are discussed in more detail later in this topic.

Recipient type	Description
Dynamic distribution group	A distribution group that uses recipient filters and conditions to derive its membership at the time messages are sent.
Equipment mailbox	A resource mailbox that's assigned to a

	resource that's not location-specific, such as a portable computer, projector, microphone, or a company car. Equipment mailboxes can be included as resources in meeting requests, providing a simple and efficient way of using resources for your users.
Linked mailbox	A mailbox that's assigned to an individual user in a separate, trusted forest.
Mail contact	A mail-enabled Active Directory contact that contains information about people or organizations that exist outside the Exchange organization. Each mail contact has an external email address. All messages sent to the mail contact are routed to this external email address.
Mail forest contact	A mail contact that represents a recipient object from another forest. Mail forest contacts are typically created by Microsoft Identity Integration Server (MIIS) synchronization. **Important:** Mail forest contacts are read-only recipient objects that are updated only through MIIS or similar custom synchronization. You can't use the EAC or the Shell to remove or modify a mail forest contact.
Mail user	A mail-enabled Active Directory user that represents a user outside the Exchange organization. Each mail user has an external email address. All messages sent to the mail user are routed to this external email address. A mail user is similar to a mail contact, except that a mail user has Active Directory logon

	credentials and can access resources.
Mail-enabled non-universal group	A mail-enabled Active Directory global or local group object. Mail-enabled non-universal groups were discontinued in Exchange Server 2007 and can exist only if they were migrated from Exchange 2003 or earlier versions of Exchange. You can't use Exchange Server 2013 to create non-universal distribution groups.
Mail-enabled public folder	An Exchange public folder that's configured to receive messages.
Distribution groups	A distribution group is a mail-enabled Active Directory distribution group object that can be used only to distribute messages to a group of recipients.
Mail-enabled security group	A mail-enabled security group is an Active Directory universal security group object that can be used to assign access permissions to resources in Active Directory and can also be used to distribute messages.
Microsoft Exchange recipient	A special recipient object that provides a unified and well-known message sender that differentiates system-generated messages from other messages. It replaces the System Administrator sender used for system-generated messages in earlier versions of Exchange.
Room mailbox	A resource mailbox that's assigned to a meeting location, such as a conference room, auditorium, or training room. Room mailboxes can be included as resources in meeting

	requests, providing a simple and efficient way of organizing meetings for your users.
Shared mailbox	A mailbox that's not primarily associated with a single user and is generally configured to allow access for multiple users.
Site mailbox	A mailbox comprised of an Exchange mailbox to store email messages and a SharePoint site to store documents. Users can access both email messages and documents using the same client interface. For more information, see Site mailboxes.
User mailbox	A mailbox that's assigned to an individual user in your Exchange organization. It typically contains messages, calendar items, contacts, tasks, documents, and other important business data.
Office 365 mailbox	In hybrid deployments, an Office 365 mailbox consists of a mail user that exists in Active Directory on-premises and an associated cloud mailbox that exists in Exchange Online.
Linked user	A linked user is a user whose mailbox resides in a different forest than the forest in which the user resides.

Mailboxes

Mailboxes are the most common recipient type used by information workers in an Exchange organization. Each mailbox is associated with an Active Directory user account. The user can use the mailbox to send and receive messages, and to store messages, appointments, tasks, notes, and documents. Mailboxes are the primary messaging and collaboration tool for the users in your Exchange organization.

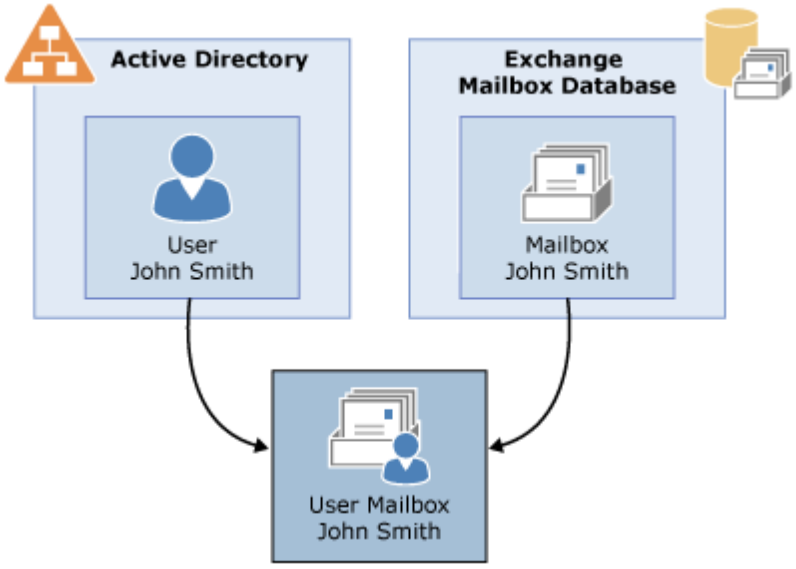
Mailbox components

Each mailbox consists of an Active Directory user and the mailbox data that's stored in the Exchange mailbox database (as shown in the following figure). All configuration data for the mailbox is stored in the Exchange attributes of the Active Directory user object. The mailbox database contains the actual data that's in the mailbox associated with the user account.

Important:

When you create a mailbox for a new or existing user, the Exchange attributes required for a mailbox are added to the user object in Active Directory. The associated mailbox data isn't created until the mailbox either receives a message or the user signs in to it.

Mailbox components



Warning:

If you remove a mailbox, the mailbox data stored in the Exchange mailbox database is marked for deletion and the associated user account is also deleted from Active Directory. To retain the user account and delete only the mailbox data, you must disable the mailbox.

Mailbox types

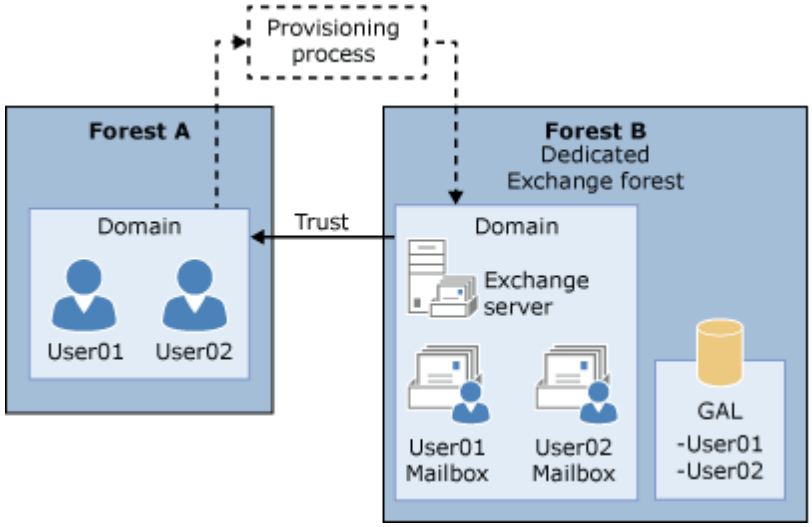
Exchange supports the following mailbox types:

- **User mailboxes** User mailboxes are assigned to individual users in your Exchange organization. User mailboxes provide your users with a rich collaboration platform. Users can send and receive messages, manage their contacts, schedule meetings, and maintain a task list. They can also have voice mail messages delivered to their mailboxes. User mailboxes are the most commonly used mailbox type and are typically the mailbox type assigned to users in your organization.
- **Linked mailboxes** Linked mailboxes are mailboxes that are accessed by users in a separate, trusted forest. Linked mailboxes may be necessary for organizations that deploy Exchange in a resource forest. The resource forest scenario allows an organization to centralize Exchange in a single forest, while allowing access to the Exchange organization with user accounts in one or more trusted forests.

As stated earlier, every mailbox must have a user account associated with it. However, the user account that accesses the linked mailbox doesn't exist in the forest where Exchange is deployed. Therefore, a disabled user account that exists in the same forest as Exchange is associated with

each linked mailbox. The following figure illustrates the relationship between the linked user account used to access the linked mailbox and the disabled user account in the Exchange resource forest associated with the linked mailbox.

Linked mailbox



- **Office 365 mailboxes** When you create an Office 365 mailbox in Exchange Online in a hybrid deployment, the mail user is created in Active Directory on-premises. Directory synchronization, if it's configured, automatically synchronizes this new user object to Office 365, where it's converted to a cloud mailbox in Exchange Online. You can create Office 365 mailboxes as regular user mailboxes, resource mailboxes for meeting rooms and equipment, and shared mailboxes.
- **Shared mailboxes** Shared mailboxes aren't primarily associated with individual users and are generally configured to allow access by multiple users.

Although it's possible to assign additional users the logon access permissions to any mailbox type, shared mailboxes are dedicated for this functionality. The Active Directory user associated with a shared mailbox must be a disabled account. After you create a shared mailbox, you must assign permissions to all users that require access to the shared mailbox.

- **Resource mailboxes** Resource mailboxes are special mailboxes designed to be used for scheduling resources. Like all mailbox types, a resource mailbox has an associated Active Directory user account, but it must be a disabled account. The following are the types of resource mailboxes:
  - **Room mailboxes** These mailboxes are assigned to meeting locations, such as conference rooms, auditoriums, and training rooms.
  - **Equipment mailboxes** These mailboxes are assigned to resources that aren't location-specific, such as portable computers, projectors, microphones, or company cars.

You can include both types of resource mailboxes in meeting requests, providing a simple and efficient way for your users to use resources. You can configure resource mailboxes to automatically process incoming meeting requests based on the resource booking policies that are defined by the resource owners. For example, you can configure a conference room to automatically accept incoming meeting requests except recurring meetings, which can be subject to approval by the resource owner.

System mailboxes

System mailboxes are created by Exchange in the root domain of the Active Directory forest during installation. Users or administrators can't sign in to these mailboxes. System mailboxes are created for Exchange features such as Unified Messaging (UM), migration, message approval, and In-Place eDiscovery. This table lists information about system mailboxes as they're displayed in Active Directory.

Mailbox	Name
Organization	SystemMailbox {bb558c35-97f1-4cb9-8ff7-d53741dc928c}
Message approval	SystemMailbox {1f05a927-*xxxx*- *xxxx* - *xxxx* - *xxxxxxxxxxxx*} where *x* is a randomly assigned and unique number for each Exchange forest
UM data storage	SystemMailbox {e0dc1c29-89c3-4034-b678-e6c29d823ed9}
Discovery	DiscoverySearchMailbox {D919BA05-46A6-415f-80AD-7E09334BB852}
Federated email	FederatedEmail.4c1f4d8b-8179-4148-93bf-00a95fa1e042
Migration	Migration.8f3e7716-2011-43e4-96b1-aba62d229136

If you want to decommission the last Mailbox server in your Exchange organization, you should first disable these system mailboxes by using the Disable-Mailbox cmdlet. When you decommission a Mailbox server that contains these system mailboxes, you should move the system mailboxes to another Mailbox server to make sure that you don't lose functionality.

Planning for mailboxes

Mailboxes are created in mailbox databases on Exchange servers that have the Mailbox server role installed. To help provide a reliable and effective platform for your mailbox users, detailed planning for the deployment of Mailbox servers and databases is essential. To learn more about planning for Mailbox servers and databases, see Planning and deployment.

Distribution groups

Distribution groups are mail-enabled Active Directory group objects that are primarily used for distributing messages to multiple recipients. Any recipient type can be a member of a distribution group.

Important:

Note the terminology differences between Active Directory and Exchange. In Active Directory, a distribution group refers to any group that doesn't have a security context, whether it's mail-enabled or not. In Exchange, all mail-enabled groups are referred to as distribution groups, whether they have a security context or not.

Exchange supports the following types of distribution groups:

- **Distribution groups** These are Active Directory universal distribution group objects that are mail-enabled. They can be used only to distribute messages to a group of recipients.
- **Mail-enabled security groups** These are Active Directory universal security group objects that are mail-enabled. They can be used to assign access permissions to resources in Active Directory and can also be used to distribute messages.
- **Mail-enabled non-universal groups** These are Active Directory global or local group objects that are mail-enabled. You can create or mail-enable only universal distribution groups. You may have mail-enabled groups that were migrated from previous versions of Exchange that aren't universal groups. These groups can still be managed by using the EAC or the Shell.

Note:

To convert a domain-local or a global group to a universal group, you can use the Set-Group cmdlet in the Shell.

Dynamic distribution groups

Dynamic distribution groups are distribution groups whose membership is based on specific recipient filters rather than a defined set of recipients.

Unlike regular distribution groups, the membership list for dynamic distribution groups is calculated each time a message is sent to them, based on the filters and conditions that you specify. When an email message is sent to a dynamic distribution group, it's delivered to all recipients in the organization that match the criteria defined for that dynamic distribution group.

Important:

A dynamic distribution group includes any recipient in Active Directory that has attributes that match the group's filter at the time a message is sent. If a recipient's properties are modified to match the group's filter, that recipient could inadvertently become a group member and start receiving messages that are sent to the dynamic distribution group. Well-defined, consistent account provisioning processes can reduce the chances of this issue occurring.

To help you create recipient filters for dynamic distribution groups, you can use precanned filters. A *precanned filter* is a commonly used filter that you can use to meet a variety of recipient-filtering criteria. You can use these filters to specify the recipient types that you want to include in a dynamic distribution group. In addition, you can also specify a list of conditions that the recipients must

meet. You can create precanned conditions based on the following properties:

- Custom attributes 1–15
- State or province
- Company
- Department
- Recipient container

You can also specify conditions based on recipient properties other than those previously listed. To do this, you must use the Shell to create a custom query for the dynamic distribution group. Remember that the filter and condition settings for dynamic distribution groups that have custom recipient filters can be managed only by using the Shell. For an example of how to create a dynamic distribution group by using a custom query, see Manage dynamic distribution groups.

Mail contacts

Mail contacts typically contain information about people or organizations that exist outside your Exchange organization. Mail contacts can appear in your organization's shared address book (also called the global address list or GAL) and other address lists, and can be added as members to distribution groups. Each contact has an external email address, and all email messages that are sent to a contact are automatically forwarded to that address. Contacts are ideal for representing people external to your Exchange organization (in the shared address book) who don't need access to any internal resources. The following are mail contact types:

- **Mail contacts** These are mail-enabled Active Directory contacts that contain information about people or organizations that exist outside your Exchange organization.
- **Mail forest contacts** These represent recipient objects from another forest. These contacts are typically created by directory synchronization. Mail forest contacts are read-only recipient objects that can be updated or removed only by means of synchronization. You can't use Exchange management interfaces to modify or remove a mail forest contact.

Mail users

Mail users are similar to mail contacts. Both have external email addresses, both contain information about people outside your Exchange organization, and both can be displayed in the shared address book and other address lists. However, unlike a mail contact, mail users have Active Directory logon credentials and can access resources to which they are assigned permissions.

If a person external to your organization requires access to resources on your network, you should create a mail user instead of a mail contact. For example, you may want to create mail users for short-term consultants who require access to your server infrastructure, but who will use their own external addresses.

Another scenario is to create mail users in your organization for users who you don't want to maintain an Exchange mailbox. For example, after an acquisition, the acquired company may maintain their separate messaging infrastructure, but may also need access to resources on your

network. For those users, you may want to create mail users instead of mailbox users.

Note:
In the EAC, you use the **Recipients** > **Contacts** page to create and manage mail users. There isn't a separate page for mail users.

Mail-enabled public folders

Public folders are intended to serve as a repository for information shared among many users. Mail-enabling a public folder provides an extra level of functionality to users. In addition to being able to post messages to the folder, users can send email messages to, and sometimes receive email messages from, the public folder. Each mail-enabled folder has an object in Active Directory that stores its email address, address book name, and other mail-related attributes.

You can manage public folders by using either the EAC or the Shell. For more information about managing public folders, see Public folders.

Microsoft Exchange recipient

The Microsoft Exchange recipient is a special recipient object that provides a unified and well-known message sender that differentiates system-generated messages from other messages. It replaces the System Administrator sender that was used for system-generated messages in earlier versions of Exchange.

The Microsoft Exchange recipient isn't a typical recipient object, such as a mailbox, mail user, or mail contact, and it isn't managed by using the typical recipient tools. However, you can use the Set-OrganizationConfig cmdlet in the Shell to configure the Microsoft Exchange recipient.

Note:
When system-generated messages are sent to an external sender, the Microsoft Exchange recipient isn't used as the sender of the message. Instead, the email address specified by the *ExternalPostmasterAddress* parameter in the Set-TransportConfig cmdlet is used.

Recipients documentation

The following table contains links to topics that will help you learn about and manage Exchange recipients.

Topic	Description
Create user mailboxes	Learn how to create user mailboxes using the Exchange admin center or the Exchange Management Shell.
Manage user mailboxes	Learn how to create user mailboxes, change

	mailbox properties, and bulk-edit selected properties for multiple mailboxes.
Manage linked mailboxes	Learn about the requirements for linked mailboxes, how to create and link them to a master account, and change linked mailbox properties.
Manage Distribution Groups	Learn how to create and manage distribution groups, and create a group naming policy for your organization.
Manage mail-enabled security groups	Learn how to create and manage mail-enabled security groups.
Manage dynamic distribution groups	Learn how to create dynamic distribution groups and manage dynamic distribution group properties, such as using custom attributes and other properties to determine group membership.
Manage mail contacts	Learn how to create and manage mail contacts.
Manage mail users	Learn how to create and manage mail users.
Create and Manage Room Mailboxes	Learn how to create room mailboxes and manage room mailbox properties, such as enabling recurring meetings and configuring booking and scheduling options.
Manage equipment mailboxes	Learn how to create equipment mailboxes, configure booking and scheduling options, and manage other mailbox properties.
Disconnected mailboxes	Learn about the two types of disconnected mailboxes and how to work with them.
Custom attributes	Learn how to add information about a recipient

	by using custom attributes.
Filters in recipient Shell commands	Learn how to use precanned or custom filters with commands to filter a set of recipients.
Manage Permissions for Recipients	Learn how to use the EAC or the Shell to assign permissions to users and groups.
Automatic mailbox distribution	Learn about how automatic mailbox distribution works and how to control which mailbox databases are selected for new and moved mailboxes.

Create user mailboxes

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-12*

Mailboxes are the most common recipient type used by information workers in an Exchange organization. Each mailbox is associated with an Active Directory user account. The user can use the mailbox to send and receive messages, and to store messages, appointments, tasks, notes, and documents. Use the EAC or the Shell to create user mailboxes.

You can also create user mailboxes for existing users that have an Active Directory user account but don't have a corresponding mailbox. This is known as *mailbox-enabling* existing users.

What do you need to know before you begin?

- Estimated time to complete each user mailbox task: 2 to 5 minutes.
- When you create a new user mailbox, you can't use an apostrophe (') or a quotation mark (") in the alias or the user logon name because these characters aren't supported. Although you might not receive an error if you create a new mailbox using unsupported characters, these characters can cause problems later. For example, users that have been assigned access permissions to a mailbox that was created using an unsupported character may experience problems or unexpected behavior.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the

Recipients Permissions topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Create a user mailbox

Use the EAC to create a user mailbox

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. Click **New** > **User mailbox**.
3. On the **New user mailbox** page, in the **Alias** box, type the user's alias, which specifies the email alias for the user. The user's alias is the portion of the email address on the left side of the at (@) symbol. It must be unique in the forest.

Note:
If you leave this box blank, the value from the user name portion of the **User logon name** is used for the email alias.

4. Select one of the following options:
   - **Existing user** Select to mail-enable and create a mailbox for an existing user.

Click **Browse** to open the **Select User – Entire Forest** dialog box. This dialog box displays a list of Active Directory user accounts in the forest that aren't mail-enabled or don't have Exchange mailboxes. Select the user account you want to mail-enable, and then click **OK**. If you select this option, you don't have to provide user account information because this information already exists in Active Directory.

   - **New user** Select to create a new user account in Active Directory and create a mailbox for this user. If you select this option, you'll have to provide the required user account information.

Note:
The Active Directory account that is associated with user mailboxes must reside in the same forest as the Exchange server. To create a mailbox for a user account that resides in a trusted forest, you have to create a linked mailbox. See Manage linked mailboxes.

5. If you selected **New user** in Step 4, complete the following boxes on the **New user mailbox** page. Otherwise skip to Step 7.
   - **First name** Use this box to type the first name of the user.
   - **Initials** Use this box to type the initials of the user.
   - **Last name** Use this box to type the last name of the user.
   - *** Display name** Use this box to type a display name for the user. This is the name that's listed in the mailbox list in the EAC and in the shared address book. By default, this box is populated

with the names you enter in the **First name**, **Initials**, and **Last name** boxes. If you didn't use those boxes, you must still type a name in this box because it's required. The name can't exceed 64 characters.

- *** Name** Use this box to type a name for the user. This is the name that's listed in Active Directory. This box is also populated with the names you enter in the **First name**, **Initials**, and **Last name** boxes. If you didn't use those boxes, you must still type a name because this box is required. This name also can't exceed 64 characters.
- **Organizational unit** You can select an organizational unit (OU) other than the default (which is the recipient scope). If the recipient scope is set to the forest, the default value is set to the Users container in the Active Directory domain that contains the computer on which the EAC is running. If the recipient scope is set to a specific domain, the Users container in that domain is selected by default. If the recipient scope is set to a specific OU, that OU is selected by default.

To select a different OU, click **Browse**. The dialog box displays all OUs in the forest that are within the specified scope. Select the desired OU, and then click **OK**.

- *** User logon name** Use this box to type the name that the user will use to sign in to the mailbox and to log on to the domain. The user logon name consists of a user name on the left side of the at (@) symbol and a suffix on the right side. Typically, the suffix is the domain name in which the user account resides. Note that you can't use an apostrophe (') or a quotation mark (") in the user logon name because these characters aren't supported.

Note:

If the value for the user name is different than the value used in the **Alias** box, then the user's email address and the user logon name will be different.

- *** New Password** Use this box to type the password that the user must use to sign in to his or her mailbox.

Note:

Make sure that the password you supply complies with the password length, complexity, and history requirements of the domain in which you are creating the user account.

- *** Confirm password** Use this box to confirm the password that you typed in the **Password** box.
- **Require password change on next logon** Select this check box if you want the user to reset the password when they first sign in to the mailbox.

If you select this check box, at first sign-in, the new user will be prompted with a dialog box in which to change the password. The user won't be allowed to perform any tasks until the password is successfully changed.

6. Click **More options** to configure the following boxes. Otherwise, skip to Step 7 to save the new user mailbox.
   - **Specify the mailbox database** Use this option to specify a mailbox database instead of allowing Exchange to select a database for you. Click **Browse** to open the **Select Mailbox Database** dialog box. This dialog box lists all the mailbox databases in your Exchange organization. By default, the mailbox databases are sorted by name. You can also click the title of the corresponding column to sort the databases by server name or version. Select the

mailbox database you want to use, and then click **OK**.

- **Create local archive storage for this user** Select this check box to create an archive mailbox for the mailbox. If you create an archive mailbox, mailbox items will be moved automatically from the primary mailbox to the archive, based on the default retention policy settings or those you define.

Click **Browse** to select a database that resides in the local forest to store the archive mailbox.

To learn more, see In-Place Archiving.

- **Address book policy** Use this option to specify an address book policy (ABP) for the mailbox. ABPs contain a global address list (GAL), an offline address book (OAB), a room list, and a set of address lists. When assigned to mailbox users, an ABP provides them with access to a customized GAL in Outlook and Outlook Web App. To learn more, see Address book policies.

In the drop-down list, select the policy that you want associated with this mailbox.

7. When you're finished, click **Save** to create the mailbox.

Use the Shell to create a user mailbox

This example creates a new user account and mailbox for Pilar Pinilla with the following details:

- The alias is pilarp
- The first name is Pilar and the last name is Pinilla
- The name and display name is Pilar Pinilla
- The user logon name is pilarp@contoso.com
- The password is Pa$$word1
- The mailbox will be created in the default OU. To specify a different OU, you can use the *OrganizationalUnit* parameter.

```
New-Mailbox -Alias pilarp -Name "Pilar Pinilla" -FirstName
Pilar -LastName Pinilla -DisplayName "Pilar Pinilla" -
UserPrincipalName pilarp@contoso.com -Password (ConvertTo-
SecureString -String 'Pa$$word1' -AsPlainText -Force)
```

For syntax and parameter information, see New-Mailbox.

How do you know this worked?

To verify that you've successfully created a user mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**. The new user mailbox is displayed in the mailbox list. Under **Mailbox Type**, the type is **User**.
- In the Shell, run the following command to display information about the new user mailbox.

```
Get-Mailbox <Name> | FL
Name,RecipientTypeDetails,PrimarySmtpAddress
```

Create a mailbox for an existing user

You can also create user mailboxes for existing users that have an Active Directory user account but don't have a corresponding mailbox. This is known as *mailbox-enabling* existing users. After you mailbox-enable an existing user, the user can send and receive email messages.

Use the EAC to create a mailbox for an existing user

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. Click **New** > **User mailbox**.
3. On the **New user mailbox** page, in the **Alias** box, type the user's alias, which specifies the email alias for the user. The user's alias is the portion of the email address on the left side of the at (@) symbol. It must be unique in the forest.

Note:
If you leave this box blank, the value from the user name portion of the **User logon name** is used for the email alias.

4. Click **Existing user**.
5. Click **Browse** to open the **Select User – Entire Forest** dialog box. This dialog box displays a list of Active Directory user accounts in the forest that aren't mail-enabled or don't have Exchange mailboxes. Select the user account you want to mail-enable, and then click **OK**.

When you create a mailbox for an existing user, you don't have to provide account information because this information already exists in Active Directory.

Note:
The Active Directory account that is associated with user mailboxes must reside in the same forest as the Exchange server. To create a mailbox for a user account that resides in a trusted forest, you have to create a linked mailbox. See Manage linked mailboxes.

6. Click **More options** to configure the following boxes. Otherwise, skip to Step 7 to save the new user mailbox.
   - **Specify the mailbox database** Use this option to specify a mailbox database instead of allowing Exchange to select a database for you. Click **Browse** to open the **Select Mailbox Database** dialog box. This dialog box lists all the mailbox databases in your Exchange organization. By default, the mailbox databases are sorted by name. You can also click the title of the corresponding column to sort the databases by server name or version. Select the mailbox database you want to use, and then click **OK**.
   - **Create local archive storage for this user** Select this check box to create an archive mailbox for the mailbox. If you create an archive mailbox, mailbox items will be moved automatically from the primary mailbox to the archive, based on the default retention policy settings or those you define.

Click **Browse** to select a database that resides in the local forest to store the archive mailbox.

To learn more, see In-Place Archiving.

   - **Address book policy** Use this option to specify an address book policy (ABP) for the mailbox. ABPs contain a global address list (GAL), an offline address book (OAB), a room list,

and a set of address lists. When assigned to mailbox users, an ABP provides them with access to a customized GAL in Outlook and Outlook Web App. To learn more, see Address book policies.

In the drop-down list, select the policy that you want associated with this mailbox.

7. When you're finished, click **Save** to create the mailbox.

Use the Shell to create a mailbox for an existing user

This example creates a mailbox for the existing user estherv@contoso.com on the Exchange database named UsersMailboxDatabase.

```
Enable-Mailbox estherv@contoso.com -Database
UsersMailboxDatabase
```

You can also use the **Enable-Mailbox** cmdlet to mail-enable multiple users. You can do this by piping the results of the **Get-User** cmdlet to the **Enable-Mailbox** cmdlet. When you run the **Get-User** cmdlet, you must return only users that aren't already mail-enabled. To do this, you need to specify the value User with the *RecipientTypeDetails* parameter. You can also limit the results returned by using the *Filter* parameter to request only users that meet the criteria you specify. You then pipe the results to the **Enable-Mailbox** cmdlet.

For example, the following command mailbox-enables users who aren't already mail-enabled and that have a value in the **UserPrincipalName** property, which helps ensure that you don't inadvertently convert a system account to a mailbox.

```
Get-User -RecipientTypeDetails User -Filter
{ UserPrincipalName -ne $Null } | Enable-Mailbox
```

For syntax and parameter information, see Enable-Mailbox and Get-User.

For more information about pipelining, see Pipelining.

How do you know this worked?

To verify that you've successfully created a mailbox for an existing user, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**. The new mailbox-enabled user is displayed in the mailbox list. Under **Mailbox Type**, the type is **User**.
- In the Shell, run the following command to display information about the new mailbox-enabled user.

```
Get-Mailbox <Name> | FL
Name,RecipientTypeDetails,PrimarySmtpAddress
```

Note that value for the *RecipientTypeDetails* property is `UserMailbox`.

Manage user mailboxes

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-27*

After you create a user mailbox, you can make changes and set additional properties by using the EAC or the Shell.

You can also change properties for multiple user mailboxes at the same time. For more information, see Bulk edit user mailboxes.

What do you need to know before you begin?

- Estimated time to complete each user mailbox task: 2 to 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Change user mailbox properties

Use the EAC to change user mailbox properties

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to change the properties for, and then click **Edit** .
3. On the mailbox properties page, click one of the following sections to view or change properties.
   - General
   - Mailbox Usage
   - Contact Information
   - Organization
   - Email Address
   - Mailbox Features

- Member Of
- MailTip
- Mailbox Delegation

General

Use the **General** section to view or change basic information about the user.

- **First name**, **Initials**, **Last name**
- *** Name** This is the name that's listed in Active Directory. If you change this name, it can't exceed 64 characters.
- *** Display name** This name appears in your organization's address book, on the To: and From: lines in email, and in the Mailbox list. This name can't contain empty spaces before or after the display name.
- *** Alias** This specifies the email alias for the user. The user's alias is the portion of the email address on the left side of the at (@) symbol. It must be unique in the forest.
- *** User logon name** This is the name that the user uses to sign in to their mailbox and to log on to the domain. Typically the user logon name consists of the user's alias on the left side of the @ symbol, and the domain name in which the user account resides on the right side of the @ symbol.

Note:
This box is labeled **User ID** In Exchange Online.

- **Require password change on next logon** Select this check box if you want the user to reset their password the next time they sign in to their mailbox.

Note:
This check box isn't available in Exchange Online.

- **Hide from address lists** Select this check box to prevent the recipient from appearing in the address book and other address lists that are defined in your Exchange organization. After you select this check box, users can still send messages to the recipient by using the email address.

Click **More options** to view or change these additional properties:

- **Organizational unit** This read-only box displays the organizational unit (OU) that contains the user account. You have to use Active Directory Users and Computers to move the user account to a different OU.

Note:
This box isn't available in Exchange Online.

- **Mailbox database** This read-only box displays the name of the mailbox database that hosts the mailbox. To move the mailbox to a different database, select it in the mailbox list, and then click **Move mailbox to another database** in the Details pane.

Note:
This option isn't available in Exchange Online.

- **Custom attributes** This section displays the custom attributes defined for the user mailbox. To specify custom attribute values, click **Edit**. You can specify up to 15 custom attributes for the

recipient.

Mailbox Usage

Use the **Mailbox Usage** section to view or change the mailbox storage quota and deleted item retention settings for the mailbox. These settings are configured by default when the mailbox is created. They use the values that are configured for the mailbox database and apply to all mailboxes in that database. You can customize these settings for each mailbox instead of using the mailbox database defaults.

- **Last logon** This read-only box displays the last time that the user signed in to their mailbox.
- **Mailbox usage** This area shows the total size of the mailbox and the percentage of the total mailbox quota that has been used.

Note:

To obtain the information that's displayed in the previous two boxes, the EAC queries the mailbox database that hosts the mailbox. If the EAC is unable to communicate with the Exchange store that contains the mailbox database, these boxes will be blank. A warning message is displayed if the user hasn't signed in to the mailbox for the first time.

Click **More options** to view or change the mailbox storage quota and the deleted item retention settings for the mailbox.

Note:

These settings aren’t available in the EAC in Exchange Online.

- **Storage quota settings** To customize these settings for the mailbox and not use the mailbox database defaults, click **Customize the settings for this mailbox**, type a new value, and then click **Save**.

The value range for any of the storage quota settings is from 0 through 2047 gigabytes (GB).

  - **Issue a warning at (GB)** This box displays the maximum storage limit before a warning is issued to the user. If the mailbox size reaches or exceeds the value specified, Exchange sends a warning message to the user.
  - **Prohibit send at (GB)** This box displays the *prohibit send* limit for the mailbox. If the mailbox size reaches or exceeds the specified limit, Exchange prevents the user from sending new messages and displays a descriptive error message.
  - **Prohibit send and receive at (GB)** This box displays the *prohibit send and receive* limit for the mailbox. If the mailbox size reaches or exceeds the specified limit, Exchange prevents the mailbox user from sending new messages and won't deliver any new messages to the mailbox. Any messages sent to the mailbox are returned to the sender with a descriptive error message.

- **Deleted item retention settings** To customize these settings for the mailbox and not use the mailbox database defaults, click **Customize the settings for this mailbox**, type a new value, and then click **Save**.
  - **Keep deleted items for (days)** This box displays the length of time that deleted items are retained before they are permanently deleted and can’t be recovered by the user. When the mailbox is created, this value is based on the deleted item retention settings configured for the mailbox database. By default, a mailbox database is configured to retain deleted items for 14

days. The value range for this property is from 0 through 24855 days.

- **Don’t permanently delete items until the database is backed up** Select this check box to prevent mailboxes and email messages from being deleted until after the mailbox database on which the mailbox is located has been backed up.

Contact Information

Use the **Contact Information** section to view or change the user's contact information. The information on this page is displayed in the address book. Click **More options** to display additional boxes.

Tip:
You can use the **State/Province** box to create recipient conditions for dynamic distribution groups, email address policies, or address lists.

Mailbox users can use Outlook or Outlook Web App to view and change their own contact information. But they can’t change the information in the **Notes** and **Web page** boxes.

Organization

Use the **Organization** section to record detailed information about the user's role in the organization. This information is displayed in the address book. Also, you can create a virtual organization chart that is accessible from email clients such as Outlook.

- **Title** Use this box to view or change the recipient's title.
- **Department** Use this box to view or change the department in which the user works. You can use this box to create recipient conditions for dynamic distribution groups, email address policies, or address lists.
- **Company** Use this box to view or change the company for which the user works. You can use this box to create recipient conditions for dynamic distribution groups, email address policies, or address lists.
- **Manager** To add a manager, click **Browse**. In **Select Manager**, select a person, and then click **OK**.
- **Direct reports** You can't modify this box. A *direct report* is a user who reports to a specific manager. If you’ve specified a manager for the user, that user appears as a direct report in the details of the manager's mailbox. For example, Kari manages Chris and Kate, so Kari's mailbox is specified in the **Manager** box of Chris's mailbox and Kate's mailbox, and Chris and Kate appear in the **Direct reports** box in the properties of Kari's mailbox.

Email Address

Use the **Email Address** section to view or change the email addresses associated with the user mailbox. This includes the user’s primary SMTP address and any associated proxy addresses. The primary SMTP address (also known as the *default reply address*) is displayed in bold text in the address list, with the uppercase **SMTP** value in the **Type** column.

- **Add** Click **Add +** to add a new email address for this mailbox. Select one of following address types:

- **SMTP** This is the default address type. Click this button and then type the new SMTP address in the *** Email address** box.
- **EUM** An EUM (Exchange Unified Messaging) address is used by the Microsoft Exchange Unified Messaging service to locate UM-enabled users within an Exchange organization. EUM addresses consist of the extension number and the UM dial plan for the UM-enabled user. Click this button and type the extension number in the **Address/Extension** box. Then click **Browse** and select a dial plan for the user.
- **Custom address type** Click this button and type one of the supported non-SMTP email address types in the *** Email address** box.

Note:

With the exception of X.400 addresses, Exchange doesn't validate custom addresses for proper formatting. You must make sure that the custom address you specify complies with the format requirements for that address type.

- **Make this the reply address** In Exchange Online, you can select this check box to make the new email address the primary SMTP address for the mailbox. This check box isn't available in the EAC in Exchange 2013.

- **Automatically update email addresses based on the email address policy applied to this recipient** Select this check box to have the recipient's email addresses automatically updated based on changes made to email address policies in your organization. This box is selected by default.

Note:

This check box isn't available in Exchange Online.

- **Make this the reply address**

Mailbox Features

Use the **Mailbox Features** section to view or change the following mailbox features and settings:

- **Sharing policy** This box shows the sharing policy applied to the mailbox. A sharing policy controls how users in your organization can share calendar and contact information with users outside your Exchange organization. The Default Sharing Policy is assigned to mailboxes when they are created. To change the sharing policy that's assigned to the user, select a different one from the drop-down list.
- **Role assignment policy** This box shows the role assignment policy assigned to the mailbox. The role assignment policy specifies the role-based access control (RBAC) roles that are assigned to the user and control what specific mailbox and distribution group configuration settings users can modify. To change the role assignment policy that's assigned to the user, select a different one from the drop-down list.
- **Retention policy** This box shows the retention policy assigned to the mailbox. A retention policy is a group of retention tags that are applied to the user's mailbox. They allow you to control how long to keep items in users' mailboxes and define what action to take on items that have reached a certain age. A retention policy isn't assigned to mailboxes when they are created. To assign a retention policy to the user, select one from the drop-down list.

- **Address book policy** This box shows the address book policy applied to the mailbox. An address book policy allows you to segment users into specific groups to provide customized views of the address book. To apply or change the address book policy applied to the mailbox, select one from the drop-down list.
- **Unified Messaging** This feature is disabled by default. When you enable Unified Messaging (UM), the user will be able to use your organization's UM features and a default set of UM properties are applied to the user. Click **Enable** to enable UM for the mailbox. For information about how to enable UM, see Enable a user for voice mail.

Note:
A UM dial plan and a UM mailbox policy must exist before you can enable UM.

- **Mobile Devices** Use this section to view and change the settings for Exchange ActiveSync, which is enabled by default. Exchange ActiveSync enables access to an Exchange mailbox from a mobile device. Click **Disable Exchange ActiveSync** to disable this feature for the mailbox.
- **Outlook Web App** This feature is enabled by default. Outlook Web App enables access to an Exchange mailbox from a web browser. Click **Disable** to disable Outlook Web App for the mailbox. Click **Edit details** to add or change an Outlook Web App mailbox policy for the mailbox.
- **IMAP** This feature is enabled by default. Click **Disable** to disable IMAP for the mailbox.
- **POP3** This feature is enabled by default. Click **Disable** to disable POP3 for the mailbox.
- **MAPI** This feature is enabled by default. MAPI enables access to an Exchange mailbox from a MAPI client such as Outlook. Click **Disable** to disable MAPI for the mailbox.
- **Litigation hold** This feature is disabled by default. Litigation hold preserves deleted mailbox items and records changes made to mailbox items. Deleted items and all instances of changed items are returned in a discovery search. Click **Enable** to put the mailbox on litigation hold. If the mailbox is on litigation hold, click **Disable** to remove the litigation hold. Mailboxes on litigation hold are inactive mailboxes and can't be deleted. To delete the mailbox, remove the litigation hold. If the mailbox is on litigation hold, click **Edit details** to view and change the following litigation hold settings:
  - **Hold date** This read-only box indicates the date and time when the mailbox was put on litigation hold.
  - **Put on hold by** This read-only box indicates the user who put the mailbox on litigation hold.
  - **Note** Use this box to notify the user about the litigation hold, explain why the mailbox is on litigation hold, or provide additional guidance to the user, such as informing them that the litigation hold won't affect their day-to-day use of email.
  - **URL** Use this box to provide a URL to a website that provides information or guidance about the litigation hold on the mailbox.

Note:
The text from these boxes appears in the user's mailbox only if they are using Outlook 2010 or later versions. It doesn't appear in Outlook Web App or other email clients. To view the text from the Note and URL boxes in Outlook, click the **File** tab, and on the **Info** page, under **Account Settings**, you'll see the litigation hold comment.

- **Archiving** If an archive mailbox doesn't exist for the user, this feature is disabled. To enable an

archive mailbox, click **Enable**. If the user has an archive mailbox, the size of the archive mailbox and usage statistics are displayed. Click **Edit details** to view and change the following archive mailbox settings:

- **Status** This read-only box indicates whether an archive mailbox exists.
- **Database** This read-only box shows the name of the mailbox database that hosts the archive mailbox. This box isn't available in Exchange Online.
- **Name** Type the name of the archive mailbox in this box. This name is displayed under the folder list in Outlook or Outlook Web App.
- **Archive quota (GB)** This box shows the total size of the archive mailbox. To change the size, type a new value in the box or select a value from the drop-down list.
- **Issue warning at (GB)** This box shows the maximum storage limit for the archive mailbox before a warning is issued to the user. If the archive mailbox size reaches or exceeds the value specified, Exchange sends a warning message to the user. To change this limit, type a new value in the box or select a value from the drop-down list.

Note:
The archive quota and the issue warning quota for the archive mailbox can't be changed in Exchange Online.

- **Delivery Options** Use to forward email messages sent to the user to another recipient and to set the maximum number of recipients that the user can send a message to. Click **View details** to view and change these settings.
  - **Forwarding address** Select the **Enable forwarding** check box and then click **Browse** to display the **Select Mail User and Mailbox** page. Use this page to select a recipient to whom you want to forward all email messages that are sent to this mailbox.
  - **Deliver message to both forwarding address and mailbox** Select this check box so that messages will be delivered to both the forwarding address and the user's mailbox.
  - **Recipient limit** This setting controls the maximum number of recipients the user can send a message to. Select the **Maximum recipients** check box to limit the number of recipients allowed in the To:, Cc:, and Bcc: boxes of an email message and then specify the maximum number of recipients.

Note:
For on-premises Exchange organizations, the recipient limit is unlimited. For Exchange Online organizations, the limit is 500 recipients.

- **Message Size Restrictions** These settings control the size of messages that the user can send and receive. Click **View details** to view and change maximum size for sent and received messages.

Note:
These settings can't be changed in Exchange Online.

  - **Sent messages** To specify a maximum size for messages sent by this user, select the **Maximum message size (KB)** check box and type a value in the box. The message size must be between 0 and 2,097,151 KB. If the user sends a message larger than the specified size, the message will be returned to the user with a descriptive error message.

- **Received messages** To specify a maximum size for messages received by this user, select the **Maximum message size (KB)** check box and type a value in the box. The message size must be between 0 and 2,097,151 KB. If the user receives a message larger than the specified size, the message will be returned to the sender with a descriptive error message.

- **Message Delivery Restrictions** These settings control who can send email messages to this user. Click **View details** to view and change these restrictions.
  - **Accept messages from** Use this section to specify who can send messages to this user.
    - **All senders** Select this option to specify that the user can accept messages from all senders. This includes both senders in your Exchange organization and external senders. This option is selected by default. This option includes external users only if you clear the **Require that all senders are authenticated** check box. If you select this check box, messages from external users will be rejected.
    - **Only senders in the following list** Select this option to specify that the user can accept messages only from a specified set of senders in your Exchange organization. Click **Add +** to display the **Select Recipients** page, which displays a list of all recipients in your Exchange organization. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search** 🔍.
    - **Require that all senders are authenticated** Select this option to prevent anonymous users from sending messages to the user.
  - **Reject messages from** Use this section to block people from sending messages to this user.
    - **No senders** Select this option to specify that the mailbox won't reject messages from any senders in the Exchange organization. This option is selected by default.
    - **Senders in the following list** Select this option to specify that the mailbox will reject messages from a specified set of senders in your Exchange organization. Click **Add+** to display the **Select Recipients** page, which displays a list of all recipients in your Exchange organization. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search** 🔍.

Member Of

Use the **Member Of** section to view a list of the distribution groups or security groups to which this user belongs. You can't change membership information on this page. Note that the user may match the criteria for one or more dynamic distribution groups in your organization. However, dynamic distribution groups aren't displayed on this page because their membership is calculated each time they are used.

MailTip

Use the **MailTip** section to add a MailTip to alert users of potential issues if they send a message to this recipient. A MailTip is text that is displayed in the InfoBar when this recipient is added to the To, Cc, or Bcc boxes of a new email message.

Note:

MailTips can include HTML tags, but scripts aren't allowed. The length of a custom MailTip can't exceed 175 displayed characters. HTML tags aren't counted in the limit.

Mailbox Delegation

Use the **Mailbox Delegation** section to assign permissions to other users (also called *delegates*) to allow them to sign in to the user's mailbox or send messages on behalf of the user. You can assign the following permissions:

- **Send As** This permission allows users other than the mailbox owner to use the mailbox to send messages. After this permission is assigned to a delegate, any message that a delegate sends from this mailbox will appear as if it was sent by the mailbox owner. However, this permission doesn't allow a delegate to sign in to the user's mailbox.
- **Send on Behalf Of** This permission also allows a delegate to use this mailbox to send messages. However, after this permission is assigned to a delegate, the **From:** address in any message sent by the delegate indicates that the message was sent by the delegate on behalf of the mailbox owner.
- **Full Access** This permission allows a delegate to sign in to the user's mailbox and view the contents of the mailbox. However, after this permission is assigned to a delegate, the delegate can't send messages from the mailbox. To allow a delegate to send email from the user's mailbox, you still have to assign the delegate the Send As or the Send on Behalf Of permission.

To assign permissions to delegates, click **Add +** under the appropriate permission to display a page that displays a list of all recipients in your Exchange organization that can be assigned the permission. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search** .

Use the Shell to change user mailbox properties

Use the **Get-Mailbox** and **Set-Mailbox** cmdlets to view and change properties for user mailboxes. One advantage of using the Shell is the ability to change the properties for multiple mailboxes. For information about what parameters correspond to mailbox properties, see the following topics:

- Get-Mailbox
- Set-Mailbox

Here are some examples of using the Shell to change user mailbox properties.

This example shows how to forward Pat Coleman's email messages to Sunil Koduri's (sunilk@contoso.com) mailbox.

```
Set-Mailbox -Identity patc -DeliverToMailboxAndForward
$true -ForwardingAddress sunilk@contoso.com
```

This example uses the **Get-Mailbox** command to find all user mailboxes in the organization, and then uses the **Set-Mailbox** command to set the recipient limit to 500 recipients allowed in the To:,

Cc:, and Bcc: boxes of an email message.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'UserMailbox')} | Set-Mailbox -
RecipientLimits 500
```

This example uses the **Get-Mailbox** command to find all the mailboxes in the Marketing organizational unit, and then uses the **Set-Mailbox** command to configure these mailboxes. The custom warning, prohibit send, and prohibit send and receive limits are set to 200 megabytes (MB), 250 MB, and 280 MB respectively, and the mailbox database's default limits are ignored. This command can be used to configure a specific set of mailboxes to have larger or smaller limits than other mailboxes in the organization.

```
Get-Mailbox -OrganizationalUnit "Marketing" | Set-Mailbox -
IssueWarningQuota 209715200 -ProhibitSendQuota 262144000 -
ProhibitSendReceiveQuota 293601280 -
UseDatabaseQuotaDefaults $false
```

This example uses the **Get-Mailbox** cmdlet to find all users in the Customer Service department, and then uses the **Set-Mailbox** cmdlet to change the maximum message size for sending messages to 2 MB.

```
Get-Mailbox -Filter "Department -eq 'Customer Service'" |
Set-Mailbox -MaxSendSize 2097152
```

This example sets the MailTip translation in French and Chinese.

```
Set-Mailbox john@contoso.com -MailTipTranslations ("FR:
C'est la langue française", "CHT: 這是漢語語言")
```

How do you know this worked?

To verify that you've successfully changed properties for a user mailbox, do the following:

- In the EAC, select the mailbox and then click **Edit** ✎ to view the property or feature that you changed. Depending on the property that you changed, it might be displayed in the Details pane for the selected mailbox.
- In the Shell, use the **Get-Mailbox** cmdlet to verify the changes. One advantage of using the Shell is that you can view multiple properties for multiple mailboxes. In the example above where the recipient limit was changed, run the following command to verify the new value.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'UserMailbox')} | fl
Name,RecipientLimits
```

For the example above where the message limits were changed, run this command.

```
Get-Mailbox -OrganizationalUnit "Marketing" | fl
Name,IssueWarningQuota,ProhibitSendQuota,ProhibitSendReceiv
eQuota,UseDatabaseQuotaDefaults
```

Bulk edit user mailboxes

You can use the EAC to change the properties for multiple user mailboxes. When you select two or more user mailboxes from the mailbox list in the EAC, the properties that can be bulk edited are displayed in the Details pane. When you change one of these properties, the change is applied to all selected mailboxes.

Here's a list of the user mailbox properties and features that can be bulk edited. Note that not all properties in each area are available to be changed.

- **Contact Information** Change shared properties such as street, postal code, and city name.
- **Organization** Change shared properties such as department name, company name, and the manager that the selected users report to.
- **Custom attributes** Change or add values for custom attributes 1 – 15.
- **Mailbox quota** Change the mailbox quota values and the retention period for deleted items. This isn't available in Exchange Online.
- **Email connectivity** Enable or disable Outlook Web App, POP3, IMAP, MAPI, and Exchange ActiveSync.
- **Archive** Enable or disable the archive mailbox.
- **Retention policy, role assignment policy, and sharing policy** Update the settings for each of these mailbox features.
- **Move mailboxes to another database** Move the selected mailboxes to a different database.
- **Delegate permissions** Assign permissions to users or groups that allow them to open or send messages from other mailboxes. You can assign Full, Send As and Send on Behalf permissions to users or groups. Check out Manage Permissions for Recipients for more details.

Note:

The estimated time to complete this task is 2 minutes, but may take longer if you change multiple properties or features.

Use the EAC to bulk edit user mailboxes

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of mailboxes, select two or more mailboxes.

Tip:

You can select multiple adjacent mailboxes by holding down the Shift key and clicking the first mailbox, and then clicking the last mailbox you want to edit. You can also select multiple non-adjacent mailboxes by holding down the Ctrl key and clicking each mailbox that you want to edit.

3. In the Details pane, under **Bulk Edit**, select the mailbox properties or feature that you want to edit.
4. Make the changes on the properties page and then save your changes.

How do you know this worked?

To verify that you've successfully bulk edited user mailboxes, do one of the following:

- In the EAC, select each of the mailboxes that you bulk edited and then click **Edit** to view the property or feature that you changed.
- In the Exchange Management Shell, use the **Get-Mailbox** cmdlet to verify the changes. One advantage of using the Shell is that you can view multiple properties for multiple mailboxes. For example, say you used the bulk edit feature in the EAC to enable the archive mailbox and assign a retention policy to all users in your organization. To verify these changes, you could run the following command:

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'UserMailbox')} | fl
Name,ArchiveDatabase,RetentionPolicy
```

For more information about the available parameters for the **Get-Mailbox** cmdlet, see Get-Mailbox.

Add or remove email addresses for a mailbox

Exchange Server 2013 > Recipients > Manage user mailboxes >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-06-07*

You can use the EAC or the Shell to add or remove an email address for a user mailbox. You can configure more than one email address for the same mailbox. The additional addresses are called *proxy addresses*. A proxy address lets a user receive email that's sent to a different email address. Any email message sent to the user's proxy address is delivered to their primary email address, which is also known as the *primary SMTP address* or the *default reply address*.

Note:
The procedures in this topic show how to add or remove email addresses for a user mailbox. You can use similar procedures to add or remove email addresses for other recipient types.

For additional management tasks related to managing recipients, see the "Recipients documentation" table in Recipients.

What do you need to know before you begin?

- Estimated time to complete each procedure: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Add an email address to a user mailbox

Use the EAC to add an email address

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to add an email address to, and then click **Edit** .
3. On the mailbox properties page, click **Email Address**.

Note:

On the **Email Address** page, the primary SMTP address is displayed in bold text in the address list, with the uppercase **SMTP** value in the **Type** column.

4. Click **Add +**, and then click **SMTP** to add an SMTP email address to this mailbox.

Note:

SMTP is the default email address type. You can also add Exchange Unified Messaging (EUM) addresses or custom addresses to a mailbox. For more information, see "Change user mailbox properties" in the Manage user mailboxes topic.

5. Type the new SMTP address in the **Email address** box, and then click **OK**.

The new address is displayed in the list of email addresses for the selected mailbox.

6. Click **Save** to save the change.

Use the Shell to add an email address

The email addresses associated with a mailbox are contained in the *EmailAddresses* property for the mailbox. Because it can contain more than one email address, the *EmailAddresses* property is known as a *multivalued* property. The following examples show different ways to modify a multivalued property.

This example shows how to add an SMTP address to the mailbox of Dan Jump.

```
Set-Mailbox "Dan Jump" -EmailAddresses
@{add="dan.jump@northamerica.contoso.com"}
```

This example shows how to add multiple SMTP addresses to a mailbox.

```
Set-Mailbox "Dan Jump" -EmailAddresses
@{add="dan.jump@northamerica.contoso.com","danj@tailspintoy
s.com"}
```

For more information about how to use this method of adding and removing values for multivalued properties, see Modifying multivalued properties.

This example shows another way to add email addresses to a mailbox by specifying all addresses associated with the mailbox. In this example, danj@tailspintoys.com is the new email address that you want to add. The other two email addresses are existing addresses. The address with the case-sensitive qualifier SMTP is the primary SMTP address. You have to include all email addresses for the mailbox when you use this command syntax. If you don't, the addresses specified in the command will overwrite the existing addresses.

```
Set-Mailbox "Dan Jump" -EmailAddresses
SMTP:dan.jump@contoso.com,dan.jump@northamerica.contoso.com
,danj@tailspintoys.com
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To verify that you've successfully added an email address to a mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**, click the mailbox, and then click **Edit** ✎.
- On the mailbox properties page, click **Email Address**.
- In the list of email addresses for the mailbox, verify that the new email address is included.

Or

- Run the following command in the Shell.

```
Get-Mailbox <identity> | fl EmailAddresses
```

- Verify that the new email address is included in the results.

Remove an email address from a user mailbox

Use the EAC to remove an email address

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to remove an email address from, and then click **Edit** ✎.

3. On the mailbox properties page, click **Email Address**.
4. In the list of email addresses, select the address you want to remove, and then click **Remove** —.
5. Click **Save** to save the change.

Use the Shell to remove an email address

This example shows how to remove an email address from the mailbox of Janet Schorr.

```
Set-Mailbox "Janet Schorr" -EmailAddresses
@{remove="janets@corp.contoso.com"}
```

This example shows how to remove multiple addresses from a mailbox.

```
Set-Mailbox "Janet Schorr" -EmailAddresses
@{remove="janet.schorr@corp.contoso.com","janets@tailspinto
ys.com"}
```

For more information about how to use this method of adding and removing values for multivalued properties, see Modifying multivalued properties.

You can also remove an email address by omitting it from the command to set email addresses for a mailbox. For example, let's say Janet Schorr's mailbox has three email addresses: janets@contoso.com (the primary SMTP address), janets@corp.contoso.com, and janets@tailspintoys.com. To remove the address janets@corp.contoso.com, you would run the following command.

```
Set-Mailbox "Janet Schorr" -EmailAddresses
SMTP:janets@contoso.com,janets@tailspintoys.com
```

Because janets@corp.contoso.com was omitted in the previous command, it's removed from the mailbox.

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To verify that you've successfully removed an email address from a mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**, click the mailbox, and then click **Edit** ✎.
- On the mailbox properties page, click **Email Address**.
- In the list of email addresses for the mailbox, verify that the email address isn't included.

Or

- Run the following command in the Shell.

```
Get-Mailbox <identity> | fl EmailAddresses
```

- Verify that the email address isn't included in the results.

Use the Shell to add email addresses to multiple mailboxes

You can add a new email address to multiple mailboxes at one time by using the Shell and a comma separated values (CSV) file.

This example imports data from C:\Users\Administrator\Desktop\AddEmailAddress.csv, which has the following format.

```
Mailbox,NewEmailAddress
Dan Jump,danj@northamerica.contoso.com
David Pelton,davidp@northamerica.contoso.com
Kim Akers,kima@northamerica.contoso.com
Janet Schorr,janets@northamerica.contoso.com
Jeffrey Zeng,jeffreyz@northamerica.contoso.com
Spencer Low,spencerl@northamerica.contoso.com
Toni Poe,tonip@northamerica.contoso.com
...
```

Run the following command to use the data in the CSV file to add the email address to each mailbox specified in the CSV file.

```
Import-CSV "C:\Users\Administrator\Desktop
\AddEmailAddress.csv" | ForEach {Set-Mailbox $_.Mailbox -
EmailAddresses @{add=$_.NewEmailAddress}}
```

Note:

The column names in the first row of this CSV file (`Mailbox,NewEmailAddress`) are arbitrary. Whatever you use for column names, make sure you use the same column names in the Shell command.

How do you know this worked?

To verify that you've successfully added an email address to multiple mailboxes, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**, click a mailbox that you added the address to, and then click **Edit** ✎.
- On the mailbox properties page, click **Email Address**.
- In the list of email addresses for the mailbox, verify that the new email address is included.

Or

- Run the following command in the Shell, using the same CSV file that you used to add the new email address.

```
Import-CSV "C:\Users\Administrator\Desktop
```

```
\AddEmailAddress.csv" | ForEach {Get-Mailbox $_.Mailbox |
fl Name,EmailAddresses}
```

- Verify that the new email address is included in the results for each mailbox.

Configure email forwarding for a mailbox

Exchange Server 2013 > Recipients > Manage user mailboxes >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-26*

As an Office 365 admin of an Office 365 Enterprise or Office 365 Midsize organization, you might have company requirements to set up email forwarding for a user's mailbox. Email forwarding lets you to set up a mailbox to forward email messages sent to that mailbox to another user's mailbox in or outside of your organization.

Use the Exchange Admin Center and Exchange Management Shell

You can use either the Exchange Admin Center (EAC) or Exchange Management Shell to set up email forwarding.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" entry in the Recipients Permissions topic.

Use the Exchange Admin Center to configure email forwarding

1. In the Exchange Admin Center, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click or tap the mailbox that you want to configure mail forwarding for, and then click or tap **Edit** ✎.
3. On the mailbox properties page, click **Mailbox Features**.
4. Under **Mail Flow**, select **View details** to view or change the setting for forwarding email messages.

On this page, you can set the maximum number of recipients that the user can send a message to. For on-premises Exchange organizations, the recipient limit is unlimited. For Exchange Online

organizations, the limit is 500 recipients.

5. Check the **Enable forwarding** check box, and then click or tap **Browse**.
6. On the **Select Recipient** page, select a user you want to forward all email to. Select the **Deliver message to both forwarding address and mailbox** check box if you want both the recipient and the forwarding email address to get copies of the emails sent. Click or tap **OK**, and then click or tap **Save**.

Note:
What if you want to forward emails to an email address outside your organization? You can use Exchange Management Shell to do this. See the example in the "Use Exchange Management Shell to configure mail forwarding".

Use Exchange Management Shell to configure mail forwarding

Haven't used Exchange Management Shell much? Check out the Exchange Management Shell topic to learn more. Take a look at the Get-Mailbox and Set-Mailbox topics for more details on the cmdlets used here.

This example delivers email to the mailbox of Douglas Kohn and, at the same time, forwards all mail sent to Douglas Kohn to douglaskohn.parents@fineartschool.net.

```
Set-Mailbox -Identity "Douglas Kohn" -
DeliverToMailboxAndForward $true -ForwardingSMTPAddress
"douglaskohn.parents@fineartschool.net"
```

This example forwards all email sent to the mailbox of Ken Sanchez, an employee of Contoso Suites, to one of his coworkers, pilarp@contoso.com.

```
Set-Mailbox -Identity "Ken Sanchez" -ForwardingSMTPAddress
"pilarp@contoso.com"
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To make sure that you've successfully configured email forwarding, do one of the following:

1. In the Exchange Admin Center, go to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click or tap the mailbox that you configured email forwarding for, and then click **Edit** ✎.
3. On the mailbox properties page, click or tap **Mailbox Features**.
4. Under **Mail Flow**, click or tap **View details** to view the mail forwarding settings.

Or

Run the following command in the Shell.

```
Get-Mailbox <identity> | fl
ForwardingSMTPAddress,DeliverToMailboxandForward
```

Make sure that the forwarding address is listed in the *ForwardingSMTPAddress* parameter. Also, if the *DeliverToMailboxAndForward* parameter is set to `$true`, messages will be delivered to the mailbox and to the forwarding address. If the parameter is set to `$false`, messages are delivered only to the forwarding address.

End users

Check out the following topics on how to forward your email to another email address by using Outlook and Outlook Web App.

- Forward email to another email account
- Manage email messages by using rules

Additional information

For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Configure message delivery restrictions for a mailbox

Exchange Server 2013 > Recipients > Manage user mailboxes >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-29*

You can use the EAC or the Shell to place restrictions on whether messages are delivered to individual recipients. Message delivery restrictions are useful to control who can send messages to users in your organization. For example, you can configure a mailbox to accept or reject messages sent by specific users or to accept messages only from users in your Exchange organization.

The message delivery restrictions covered in this topic apply to all recipient types. To learn more about the different recipient types, see Recipients.

For additional management tasks related to recipients, see the following topics:

- Manage user mailboxes
- Manage Distribution Groups
- Manage dynamic distribution groups
- Manage mail users
- Manage mail contacts

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure message delivery restrictions

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to configure message delivery restrictions for, and then click **Edit** .
3. On the mailbox properties page, click **Mailbox Features**.
4. Under **Message Delivery Restrictions**, click **View details** to view and change the following delivery restrictions:
   - **Accept messages from** Use this section to specify who can send messages to this user.
     - **All senders** This option specifies that the user can accept messages from all senders. This includes both senders in your Exchange organization and external senders. This is the default option. It includes external users only if you clear the **Require that all senders are authenticated** check box. If you select this check box, messages from external users will be rejected.
     - **Only senders in the following list** This option specifies that the user can accept messages only from a specified set of senders in your Exchange organization. Click **Add +** to display a list of all recipients in your Exchange organization. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search** .
     - **Require that all senders are authenticated** This option prevents anonymous users from sending messages to the user. This includes external users that are outside of your Exchange

organization.

- **Reject messages from** Use this section to block people from sending messages to this user.
  - **No senders** This option specifies that the mailbox won't reject messages from any senders in the Exchange organization. This is the default option.
  - **Senders in the following list** This option specifies that the mailbox will reject messages from a specified set of senders in your Exchange organization. Click **Add +** to display a list of all recipients in your Exchange organization. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search**.

5. Click **OK** to close the **Message Delivery Restrictions** page, and then click **Save** to save your changes.

Use the Shell to configure message delivery restrictions

The following examples show how to use the Shell to configure message delivery restrictions for a mailbox. For other recipient types, use the corresponding **Set-** cmdlet with the same parameters.

This example configures the mailbox of Robin Wood to accept messages only from the users Lori Penor, Jeff Phillips, and members of the distribution group Legal Team 1.

```
Set-Mailbox -Identity "Robin Wood" -AcceptMessagesOnlyFrom
"Lori Penor","Jeff Phillips" -
AcceptMessagesOnlyFromDLMembers "Legal Team 1"
```

Note:

If you're configuring a mailbox to accept messages only from individual senders, you have to use the *AcceptMessagesOnlyFrom* parameter. If you're configuring a mailbox to accept messages only from senders that are members of a specific distribution group, use the *AcceptMessagesOnlyFromDLMembers* parameter.

This example adds the user named David Pelton to the list of users whose messages will be accepted by the mailbox of Robin Wood.

```
Set-Mailbox -Identity "Robin Wood" -AcceptMessagesOnlyFrom
@{add="David Pelton"}
```

This example configures the mailbox of Robin Wood to require all senders to be authenticated. This means the mailbox will only accept messages sent by other users in your Exchange organization.

```
Set-Mailbox -Identity "Robin Wood" -
RequireSenderAuthenticationEnabled $true
```

This example configures the mailbox of Robin Wood to reject messages from the users Joe Healy, Terry Adams, and members of the distribution group Legal Team 2.

```
Set-Mailbox -Identity "Robin Wood" -RejectMessagesFrom "Joe
Healy","Terry Adams" -RejectMessagesFromDLMembers "Legal
Team 2"
```

This example configures the mailbox of Robin Wood to also reject messages sent by members of the group Legal Team 3.

```
Set-Mailbox -Identity "Robin Wood" -
RejectMessagesFromDLMembers @{add="Legal Team 3"}
```

Note:

If you're configuring a mailbox to reject messages from individual senders, you have to use the *RejectMessagesFrom* parameter. If you're configuring a mailbox to reject messages from senders that are members of a specific distribution group, use the *RejectMessagesFromDLMembers* parameter.

For detailed syntax and parameter information related to configuring delivery restrictions for different types of recipients, see the following topics:

- Set-DistributionGroup
- Set-DynamicDistributionGroup
- Set-Mailbox
- Set-MailContact
- Set-MailUser

How do you know this worked?

To verify that you've successfully configured message delivery restrictions for a user mailbox, do one the following:

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to verify the message delivery restrictions for, and then click **Edit** .
3. On the mailbox properties page, click **Mailbox Features**.
4. Under **Message Delivery Restrictions**, click **View details** to verify the delivery restrictions for the mailbox.

Or

Run the following command in the Shell.

```
Get-Mailbox <identity> | fl
AcceptMessagesOnlyFrom,AcceptMessagesOnlyFromDLMembers,Reje
ctMessagesFrom,RejectMessagesFromDLMembers,RequireSenderAut
henticationEnabled
```

Configure message size limits for a mailbox

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-12*

You can use the EAC and the Shell to configure message size limits for a user mailbox. These limits control the size of messages that a user can send and receive. By default, when a mailbox is created, there isn't a size limit for sent and received messages.

Keep in mind that there are other settings in an Exchange organization that determine the maximum message size a mailbox can send and receive (for example, the maximum message size configured on a Mailbox server). To learn more about the message size restrictions in Exchange, including the types of message size limits, their scope, and the order of precedence, see Message size limits.

For additional management tasks related to user mailboxes, see Manage user mailboxes.

Note:

You can also control the size of messages sent and received by mail users and from shared mailboxes.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure message size limits

1. In the EAC, navigate to **Recipients** > **Mailboxes**.

2. In the list of user mailboxes, click the mailbox that you want to change the message size limits for, and then click **Edit** ✎.
3. On the mailbox properties page, click **Mailbox Features**.
4. Under **Message Size Restrictions**, click **View details** to view and change the following message size limits:
   - **Sent messages** To specify a maximum size for messages sent by this user, select the **Maximum message size (KB)** check box and type a value in the box. The message size must be between 0 and 2,097,151 KB. If the user sends a message larger than the specified size, the message will be returned to the user with a descriptive error message.
   - **Received messages** To specify a maximum size for messages received by this user, select the **Maximum message size (KB)** check box and type a value in the box. The message size must be between 0 and 2,097,151 KB. If the user receives a message larger than the specified size, the message will be returned to the sender with a descriptive error message.
5. Click **OK**, and then click **Save** to save your changes.

Use the Shell to configure message size limits

This example sets the maximum size for sent messages to 25 MB and the maximum size for received messages to 35 MB for the mailbox of Debra Garcia.

```
Set-Mailbox "Debra Garcia" -MaxSendSize 25mb -
MaxReceiveSize 35mb
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To verify that you've successfully configured message size limits for a mailbox, do one of the following:

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to verify the message size limits for, and then click **Edit** ✎.
3. On the mailbox properties page, click **Mailbox Features**.
4. Under **Message Size Restrictions**, click **View details** to verify the message size limits for the mailbox.

Or

Run the following command in the Shell.

```
Get-Mailbox <identity> | fl MaxSendSize,MaxReceiveSize
```

Configure storage quotas for a mailbox

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-29*

You can use the EAC or the Shell to customize the mailbox storage quotas for specific mailboxes. Storage quotas let you control the size of mailboxes and manage the growth of mailbox databases. When a mailbox reaches or exceeds a specified storage quota, Exchange sends a descriptive notification to the mailbox owner.

Storage quotas are typically configured on a per-database basis. This means that the quotas configured for a mailbox database apply to all mailboxes in that database. For more information about managing per-database mailbox settings, see Manage mailbox databases in Exchange 2013.

This topic shows you how to customize storage settings for a specific mailbox instead of using the storage settings from the mailbox database. For additional management tasks related to user mailboxes, see Manage user mailboxes.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure storage quotas for a mailbox

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to change the storage quotas for, and then click **Edit** .
3. On the mailbox properties page, click **Mailbox Usage**, and then click **More options**.
4. Click **Customize the settings for this mailbox**, and then configure the following boxes. The value range for any of the storage quota settings is from 0 through 2047 gigabytes (GB).

- **Issue a warning at (GB)** This box displays the maximum storage limit before a warning is issued to the user. If the mailbox size reaches or exceeds the value specified, Exchange sends a warning message to the user.

Important:
The message associated with the **Issue warning** quota won't be sent to the user unless the value of this setting is greater than 50% of the value specified in the **Prohibit send** quota. For example, if you set the **Prohibit send** quota to 8 MB, you must set the **Issue warning** quota to at least 4 MB. If you don't, the **Issue warning** quota message won't be sent.

- **Prohibit send at (GB)** This box displays the *prohibit send* limit for the mailbox. If the mailbox size reaches or exceeds the specified limit, Exchange prevents the user from sending new messages and displays a descriptive error message.
- **Prohibit send and receive at (GB)** This box displays the *prohibit send and receive* limit for the mailbox. If the mailbox size reaches or exceeds the specified limit, Exchange prevents the mailbox user from sending new messages and won't deliver any new messages to the mailbox. Any messages sent to the mailbox are returned to the sender with a descriptive error message.

5. Click **Save** to save your changes.

Use the Shell to configure storage quotas for a mailbox

This example sets the issue warning, prohibit send, and prohibit send and receive quotas for Joe Healy's mailbox to 24.5 GB, 24.75 GB, and 25 GB respectively.

Note:
To ensure that the custom settings for the mailbox are used rather than the mailbox database defaults, you must set the *UseDatabaseQuotaDefaults* parameter to `$false`.

```
Set-Mailbox -Identity "Joe Healy" -IssueWarningQuota 24.5gb
-ProhibitSendQuota 24.75gb -ProhibitSendReceiveQuota 25gb -
UseDatabaseQuotaDefaults $false
```

This example sets the issue warning, prohibit send, and prohibit send and receive quotas for Ayla Kol's mailbox to 900 megabytes (MB), 950 MB, and 1 GB respectively, and configures the mailbox to use custom settings.

```
Set-Mailbox -Identity "Ayla Kol" -IssueWarningQuota 900mb -
ProhibitSendQuota 950mb -ProhibitSendReceiveQuota 1gb -
UseDatabaseQuotaDefaults $false
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To verify that you've successfully set the storage quotas for a mailbox, do one of the following:

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to verify the storage quotas for, and then click **Edit** ✎.
3. On the mailbox properties page, click **Mailbox Usage**, and then click **More options**.
4. Verify that **Customize the settings for this mailbox** is selected.
5. Verify the storage quota settings.

Or

Run the following command in the Shell.

```
Get-Mailbox <identity> | fl
IssueWarningQuota,ProhibitSendQuota,ProhibitSendReceiveQuot
a,UseDatabaseQuotaDefaults
```

Convert a Mailbox

Exchange Server 2013 > Recipients > Manage user mailboxes >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-01-08*

Converting a mailbox to a different type of mailbox is very similar to the experience in Exchange 2010. You must still use the Set-Mailbox cmdlet in the Shell to do the conversion.

You can convert the following mailboxes from one type to another:

- User mailbox to resource mailbox
- Shared mailbox to user mailbox
- Shared mailbox to resource mailbox
- Resource mailbox to user mailbox
- Resource mailbox to shared mailbox

Use the Shell to convert a mailbox

Estimated time to complete: 5 minutes.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

This example converts the shared mailbox, MarketingDept1 to a user mailbox.

```
Set-Mailbox MarketingDept1 -Type Regular
```

You can use the following values for the *Type* parameter:

- Regular
- Room
- Equipment
- Shared

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To verify that you have successfully converted the mailbox, run the following Shell command:

```
Get-Mailbox -Identity MarketingDept1 | Format-List
RecipientTypeDetails
```

The value for *RecipientTypeDetails* should be *UserMailbox*.

For detailed syntax and parameter information, see Get-Mailbox.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Enable or disable Exchange ActiveSync for a mailbox

Exchange Server 2013 > Recipients > Manage user mailboxes >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-13*

You can use the EAC or the Shell to enable or disable Microsoft Exchange ActiveSync for a user mailbox. Exchange ActiveSync is a client protocol that lets users synchronize a mobile device with their Exchange mailbox. Exchange ActiveSync is enabled by default when a user mailbox is created. To learn more, see Exchange ActiveSync.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange ActiveSync settings" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see

Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or disable Exchange ActiveSync

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to enable or disable Exchange ActiveSync for, and then click **Edit** .
3. On the mailbox properties page, click **Mailbox Features**.
4. Under **Mobile Devices**, do one of the following:
   - To disable Exchange ActiveSync click **Disable Exchange ActiveSync**.

A warning appears asking if you're sure you want to disable Exchange ActiveSync. Click **Yes**.

   - To enable Exchange ActiveSync, click **Enable Exchange ActiveSync**.
5. Click **Save** to save your change.

Note:

You can enable and disable Exchange ActiveSync for multiple user mailboxes by using the EAC bulk edit feature. For more information about how to do this, see the "Bulk edit user mailboxes" section in Manage user mailboxes.

Use the Shell to enable or disable Exchange ActiveSync

This example disables Exchange ActiveSync for the mailbox of Yan Li.

```
Set-CASMailbox -Identity "Yan Li" -ActiveSyncEnabled $false
```

This example enables Exchange ActiveSync for the mailbox of Elly Nkya.

```
Set-CASMailbox -Identity Ellyn@contoso.com -
ActiveSyncEnabled $true
```

For detailed syntax and parameter information, see Set-CASMailbox.

How do you know this worked?

To verify that you've successfully enabled or disabled Exchange ActiveSync for a user mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**, click the mailbox, and then click **Edit** .
- On the mailbox properties page, click **Mailbox Features**.

- Under **Mobile Devices**, verify whether Exchange ActiveSync is enabled or disabled.

Or

- Run the following command in the Shell.

```
Get-CASMailbox <identity>
```

If Exchange ActiveSync is enabled, the value for the *ActiveSyncEnabled* property is `True`. If Exchange ActiveSync is disabled, the value is `False`.

Enable or disable MAPI for a mailbox

Exchange Server 2013 > Recipients > Manage user mailboxes >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-26*

You can use the EAC or the Shell to enable or disable MAPI for a user mailbox. When MAPI is enabled, a user's mailbox can be accessed by Microsoft Outlook or other MAPI email clients. When MAPI is disabled, it can't be accessed by Outlook or other MAPI clients. But the mailbox will continue to receive email messages, and a user can access it to send and receive email by using Outlook Web App, a POP email client, or an IMAP client, assuming that the mailbox is enabled to support access by those clients.

Note:

Support for Outlook Web App and MAPI, POP3, and IMAP4 email clients is enabled by default when a user mailbox is created.

For additional management tasks related to managing email client access to a mailbox, see the following topics:

- Enable or disable Outlook Web App for a mailbox
- Enable or disable IMAP4 access for a user
- Enable or disable POP3 access for a user

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Client Access user settings" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or disable MAPI

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to enable or disable MAPI, and then click **Edit** ✎.
3. On the mailbox properties page, click **Mailbox Features**.
4. Under **Email Connectivity**, do one of the following.
   - To disable MAPI, under **MAPI: Enabled**, click **Disable**.

A warning appears asking if you're sure you want to disable MAPI. Click **Yes**.

   - To enable MAPI, under **MAPI: Disabled**, click **Enable**.
5. Click **Save** to save your change.

Use the Shell to enable or disable MAPI

This example disables MAPI for the mailbox of Ken Sanchez.

```
Set-CASMailbox -Identity "Ken Sanchez" -MAPIEnabled $false
```

This example enables MAPI for the mailbox of Esther Valle.

```
Set-CASMailbox -Identity "Esther Valle" -MAPIEnabled $true
```

For detailed syntax and parameter information, see Set-CASMailbox.

How do you know this worked?

To verify that you've successfully enabled or disabled MAPI for a user mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**, click the mailbox, and then click **Edit** ✎.
- On the mailbox properties page, click **Mailbox Features**.
- Under **Email Connectivity**, verify whether MAPI is enabled or disabled.

Or

- Run the following command in the Shell.

```
Get-CASMailbox <identity>
```

If MAPI is enabled, the value for the *MapiEnabled* property is `True`. If MAPI is disabled, the value is `False`.

Enable or disable Outlook Web App for a mailbox

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-14*

You can use the EAC or the Shell to enable or disable Outlook Web App for a user mailbox. When Outlook Web App is enabled, a user can use Outlook Web App to send and receive email. When Outlook Web App is disabled, the mailbox will continue to receive email messages, and a user can access it to send and receive email by using a MAPI client, such as Microsoft Outlook, or with a POP or IMAP email client, assuming that the mailbox is enabled to support access by those clients.

Note:
Support for Outlook Web App and MAPI, POP3, and IMAP4 email clients is enabled by default when a user mailbox is created.

For additional management tasks related to managing email client access to a mailbox, see the following topics:

- Enable or disable MAPI for a mailbox
- Enable or disable IMAP4 access for a user
- Enable or disable POP3 access for a user

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Client Access user settings" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or disable Outlook Web App

1. In the EAC, navigate to **Recipients** > **Mailboxes**.

2. In the list of user mailboxes, click the mailbox that you want to enable or disable Outlook Web App for, and then click **Edit** .
3. On the mailbox properties page, click **Mailbox Features**.
4. Under **Email Connectivity**, do one of the following:
   - To disable Outlook Web App, under **Outlook Web App: Enabled**, click **Disable**.

A warning appears asking if you're sure you want to disable Outlook Web App. Click **Yes**.

   - To enable Outlook Web App, under **Outlook Web App: Disabled**, click **Enable**.
5. Click **Save** to save your change.

Note:

You can enable and disable Outlook Web App for multiple user mailboxes by using the EAC bulk edit feature. For more information about how to do this, see the "Bulk edit user mailboxes" section in Manage user mailboxes.

Use the Shell to enable or disable Outlook Web App

This example disables Outlook Web App for the mailbox of Yan Li.

```
Set-CASMailbox -Identity "Yan Li" -OWAEnabled $false
```

This example enables Outlook Web App for the mailbox of Elly Nkya.

```
Set-CASMailbox -Identity Ellyn@contoso.com -OWAEnabled
$true
```

For detailed syntax and parameter information, see Set-CASMailbox.

How do you know this worked?

To verify that you've successfully enabled or disabled Outlook Web App for a user mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**, click the mailbox, and then click **Edit** .
- On the mailbox properties page, click **Mailbox Features**.
- Under **Email Connectivity**, verify whether Outlook Web App is enabled or disabled.

Or

- Run the following command in the Shell.

```
Get-CASMailbox <identity>
```

If Outlook Web App is enabled, the value for the *OWAEnabled* property is `True`. If Outlook Web App is disabled, the value is `False`.

Enable single item recovery for a mailbox

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-11*

You can use the Shell to enable single item recovery on a mailbox. By default, single item recovery is disabled when a mailbox is created. If single item recovery is enabled, messages that are permanently deleted by the user are retained in the Recoverable Items folder of the mailbox until the deleted item retention period expires. If a message is changed by a user or a process, copies of the original item are also retained. In an on-premises Exchange organization, the mailbox uses the deleted item retention settings of the mailbox database, by default. The deleted item retention period for a mailbox database is set to 14 days, but you can override the default by configuring this setting on a per-mailbox basis.

If a mailbox is placed on In-Place Hold or litigation hold, deleted items are retained until the hold is removed.

To learn more about single item recovery, see Recoverable Items folder. To recover messages deleted by the user before the deleted item retention period expires, see Perform single item recovery.

Note:
In Exchange Online, the deleted item retention period is set to 14 days, by default. You can change this setting to a maximum of 30 days.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Retention and legal holds" entry in the Recipients Permissions topic.
- You can't use the Exchange admin center (EAC) to enable single item recovery. But in an on-premises Exchange organization, you can use the EAC to change the deleted item retention period. For details, see Configure Deleted Item retention and Recoverable Items quotas.

In Exchange Online, you have to use the Shell to change the deleted item retention period. For details, see **Change the Deleted Item Retention Period for a Mailbox in Exchange Online**.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to enable single item recovery

This example enables single item recovery for the mailbox of April Summers.

```
Set-Mailbox -Identity "April Summers" -
SingleItemRecoveryEnabled $true
```

This example enables single item recovery for the mailbox of Pilar Pinilla and sets the number of days that deleted items are retained to 30 days.

```
Set-Mailbox -Identity "Pilar Pinilla" -
SingleItemRecoveryEnabled $true -RetainDeletedItemsFor 30
```

This example enables single item recovery for all user mailboxes in the organization.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'UserMailbox')} | Set-Mailbox -
SingleItemRecoveryEnabled $true
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To verify that you've enabled single item recovery for a mailbox and display the value for how long deleted items will be retained (in days), run the following command.

```
Get-Mailbox <Name> | FL
SingleItemRecoveryEnabled,RetainDeletedItemsFor
```

Manage linked mailboxes

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013*

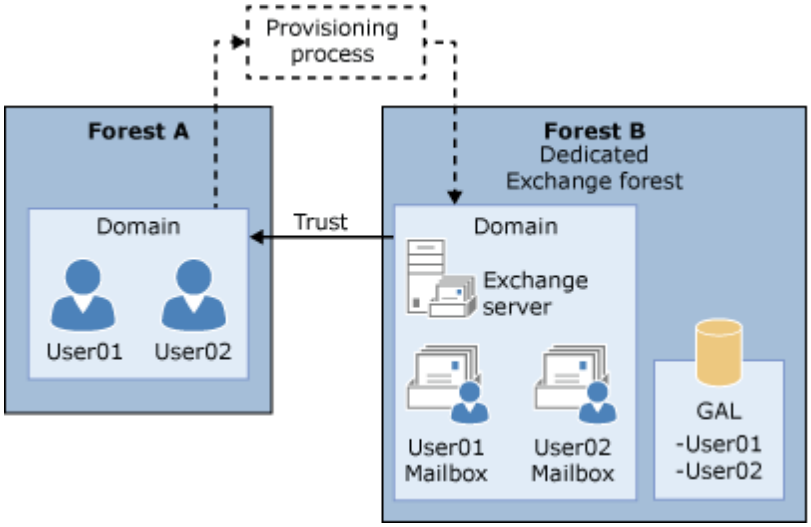
Topic Last Modified: *2012-11-27*

Linked mailboxes are mailboxes that are accessed by users in a separate, trusted forest. Linked mailboxes may be necessary for organizations that deploy Exchange in a resource forest. The

resource forest scenario allows an organization to centralize Exchange in a single forest, while allowing access to the Exchange organization with user accounts that are located in one or more trusted forests (called *account forests*). The user account that accesses the linked mailbox doesn't exist in the forest where Exchange is deployed. Therefore, a disabled user account that exists in the same forest as Exchange is created and associated with the corresponding linked mailbox.

The following figure illustrates the relationship between the linked user account used to access the linked mailbox (located in the account forest) and the disabled user account in the Exchange resource forest that's associated with the linked mailbox.

Linked mailboxes



Note:

A trust between the Exchange forest and at least one account forest must be set up before you can create linked mailboxes. At a minimum, you must set up a one-way, outgoing trust so that the Exchange forest trusts the account forest. For more information, see Learn more about setting up a forest trust to support linked mailboxes.

What do you need to know before you begin?

- Estimated time to complete: 2 to 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- A user account (called the *linked master account*) must exist in the account forest before you can create a linked mailbox. This is because the linked mailbox is associated with a user in the account forest.
- If you've configured a one-way outgoing trust where the Exchange forest trusts the account forest, you'll need administrator credentials in the account forest to create a linked mailbox.

To create a linked mailbox without being prompted for administrator credentials in the account forest, you have to create a two-way trust, or create another one-way outgoing trust where the account forest also trusts the Exchange forest. This step also requires administrator credentials in the account forest.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Create a linked mailbox

Use the EAC to create a linked mailbox

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. Click **New** > **Linked mailbox**.
3. On the **New linked mailbox** page, in the **Trusted forest or domain** box, select the name of the account forest that contains the user account that you're creating the linked mailbox for. Click **Next**.
4. If your organization has configured a one-way outgoing trust where the Exchange forest trusts the account forest, you're prompted for administrator credentials in the account forest so that you can gain access to a domain controller in the trusted forest. Type the user name and password for an administrator account in the account forest, and then click **Next**.

Note:
You won't be prompted for administrator credentials if you've created a two-way trust or have created another one-way outgoing trust where the account forest trusts the Exchange forest.

5. Complete the following boxes on the **Select linked master account** page.
   - **Linked domain controller** Select a domain controller in the account forest. Exchange will connect to this domain controller to retrieve the list of user accounts in the account forest so that you can select the linked master account.
   - **Linked master account** Click **Browse**, select a user account in the account forest, and then click **OK**. The new linked mailbox will be associated with this account.
6. Click **Next** and complete the following boxes on the **Enter general information** page.
   - *** Name** Use this box to type a name for the user. This is the name used as the display name in the EAC and your organization's address book, and the name that's listed in Active Directory. This name is required.
   - **Organizational unit** You can select an organizational unit (OU) other than the default (which is the recipient scope). If the recipient scope is set to the forest, the default value is set to the Users container in the Active Directory domain that contains the computer on which the EAC is running. If the recipient scope is set to a specific domain, the Users container in that domain is selected by default. If the recipient scope is set to a specific OU, that OU is selected by default.

To select a different OU, click **Browse**. The dialog box displays all OUs in the Exchange forest that are within the specified scope. Select the OU you want, and then click **OK**.

- *** User logon name** Use this box to type the user logon name, which is required to create a linked mailbox. Type the user name here. This name will be used in the left portion of the email address for the linked mailbox if you don't specify an alias.

Note:

Because the user account that is created in the Exchange forest is disabled when you create a linked mailbox, the user doesn't use the user logon name to sign in to the linked mailbox. They sign in using their credentials from the account forest.

7. Click **More options** to configure the following boxes. Otherwise, skip to Step 8 to save the new linked mailbox.
   - **Alias** Type the alias, which specifies the email alias for the linked mailbox. The user's alias is the portion of the email address on the left side of the at (@) symbol. It must be unique in the forest.

Note:

If you leave this box blank, the value from the user name portion of the **User Logon Name** is used for the email alias.

   - **First name**, **Initials**, **Last name**
   - **Mailbox database** Use this option to specify a mailbox database instead of allowing Exchange to choose a database for you. Click **Browse** to open the **Select Mailbox Database** dialog box. This dialog box lists all the mailbox databases in your Exchange organization. By default, the mailbox databases are sorted by name. You can also click the title of the corresponding column to sort the databases by server name or version. Select the mailbox database you want to use, and then click **OK**.
   - **Address book policy** Use this option to specify an address book policy (ABP) for the linked mailbox. ABPs contain a global address list (GAL), an offline address book (OAB), a room list, and a set of address lists. When assigned to users, an ABP provides them with access to a customized GAL in Outlook and Outlook Web App. To learn more, see Address book policies.

In the drop-down list, select the policy that you want associated with this mailbox.

8. When you're finished, click **Save** to create the new linked mailbox.

Use the Shell to create a linked mailbox

This example creates a linked mailbox for Ayla Kol in the CONTOSO Exchange resource forest. The FABRIKAM domain is in the account forest. The administrator account FABRIKAM \administrator is used to access the linked domain controller.

```
New-Mailbox -Name "Ayla Kol" -LinkedDomainController
"DC1_FABRIKAM" -LinkedMasterAccount " FABRIKAM\aylak" -
OrganizationalUnit Users -UserPrincipalName
aylak@contoso.com -LinkedCredential:(Get-Credential
FABRIKAM\administrator)
```

For syntax and parameter information, see New-Mailbox.

How do you know this worked?

To verify that you've successfully created a linked mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**. The new linked mailbox is displayed in the mailbox list. Under **Mailbox Type**, the type is **Linked**.
- In the Shell, run the following command to display information about the new linked mailbox.

```
Get-Mailbox <Name> | FL
Name,RecipientTypeDetails,IsLinked,LinkedMasterAccount
```

Change linked mailbox properties

After you create a linked mailbox, you can make changes and set additional properties by using the Exchange Administration Center (EAC) or the Exchange Management Shell.

You can also change properties for multiple linked mailboxes at the same time. For more information, see the section, "Bulk edit user mailboxes" section in the Manage user mailboxes topic.

Important:
The estimated time to complete this task will vary based on the number of properties you want to view or change.

Use the EAC to change linked mailbox properties

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of mailboxes, click the linked mailbox that you want to change the properties for, and then click **Edit** .
3. On the mailbox properties page, click one of the following sections to view or change properties.
   - General
   - Mailbox Usage
   - Email Address
   - Mailbox Features
   - Member Of
   - MailTip
   - Mailbox Delegation

General

Use the **General** section to view or change basic information about the user.

- *** Linked mailbox name** This is the name that's listed in Active Directory. If you change this name, it can't exceed 64 characters.
- *** Display name** This name appears in your organization's address book, on the To: and From: lines in email, and in the Mailboxes list in the EAC. This name can't contain empty spaces before or after the display name.
- *** User logon name** For user mailboxes, this is the name that the user uses to sign in to their mailbox and to log on to the domain. For linked mailboxes, the corresponding user account that

is created in the Exchange forest when the linked mailbox was created is disabled. The user uses their credentials from the account forest to sign in to the linked mailbox.

If you change this name, it must be unique in your organization.

- **Linked master account** This read-only box displays the user (in the format domain\username format) from the account forest that is associated with the linked mailbox. To change the linked master account associated with the linked mailbox, you have to use the **Set-Mailbox** cmdlet in the Shell. If you change the linked master account, the user will have to use the credentials for the new linked master account to sign in to the linked mailbox. For the command syntax to change the linked master account, see Use the Shell to change linked mailbox properties.
- **Hide from address lists** Select this check box to prevent the linked mailbox from appearing in the address book and other address lists that are defined in your Exchange organization. After you select this check box, users can still send messages to this user by using the email address.

Click **More options** to view or change these additional properties:

- **Organizational unit** This read-only box displays the organizational unit (OU) that contains the user account. You have to use Active Directory Users and Computers to move the user account to a different OU.
- **Mailbox database** This read-only box displays the name of the mailbox database that hosts the mailbox. To move the mailbox to a different database, select it in the mailbox list, and then click **Move mailbox to a different database** in the Details pane.
- *** Alias** This specifies the email alias for the linked mailbox. The alias is the portion of the email address on the left side of the at (@) symbol. It must be unique in the forest.
- **First name**, **Initials**, **Last name**
- **Custom attributes** This section displays the custom attributes defined for the linked mailbox. To specify custom attribute values, click **Edit** . You can specify up to 15 custom attributes for the recipient.

Mailbox Usage

Use the **Mailbox Usage** section to view or change the mailbox storage quota and deleted item retention settings for the linked mailbox. These settings are configured by default when the linked mailbox is created. They use the values that are configured for the mailbox database and apply to all mailboxes in that database. You can customize these settings for each mailbox instead of using the mailbox database defaults.

- **Last logon** This read-only box displays the last time that the user signed in to the mailbox.
- **Mailbox usage** This area shows the total size of the mailbox and the percentage of the total mailbox quota that has been used.

Note:

To obtain the information that's displayed in the previous two boxes, the EAC queries the mailbox database that hosts the mailbox. If the EAC can’t communicate with the Exchange store that contains the mailbox database, these boxes will be blank. A warning message is displayed if the user hasn't signed in to the mailbox for the first time.

Click **More options** to view or change the mailbox storage quota and the deleted item retention

settings for the mailbox.

- **Storage quota settings** To customize these settings for the mailbox and not use the mailbox database defaults, click **Customize settings for this mailbox**, type a new value, and then click **Save**.

The value range for any of the storage quota settings is from 0 through 2047 gigabytes (GB).

  - **Issue a warning at (GB)** This box displays the maximum storage limit before a warning is issued to the user. If the mailbox size reaches or exceeds the value specified, Exchange sends a warning message to the user.
  - **Prohibit send at (GB)** This box displays the *prohibit send* limit for the mailbox. If the mailbox size reaches or exceeds the specified limit, Exchange prevents the user from sending new messages and displays a descriptive error message.
  - **Prohibit send and receive at (GB)** This box displays the *prohibit send and receive* limit for the mailbox. If the mailbox size reaches or exceeds the specified limit, Exchange prevents the mailbox user from sending new messages and won't deliver any new messages to the mailbox. Any messages sent to the mailbox are returned to the sender with a descriptive error message.

- **Deleted item retention settings** To customize these settings for the mailbox and not use the mailbox database defaults, click **Customize settings for this mailbox**, type a new value, and then click **Save**.
  - **Keep deleted items for (days)** This box displays the length of time that deleted items are retained before they're permanently deleted and can't be recovered by the user. When the mailbox is created, this length of time is based on the deleted item retention settings configured for the mailbox database. By default, a mailbox database is configured to retain deleted items for 14 days. The value range for this property is from 0 through 24855 days.
  - **Don't permanently delete items until the database is backed up** Select this check box to prevent mailboxes and email messages from being deleted until after the mailbox database on which the mailbox is located has been backed up.

Email Address

Use the **Email address** section to view or change the email addresses associated with the linked mailbox. This includes the user's primary SMTP addresses and any associated proxy addresses. The primary SMTP address (also known as the *default reply address*) is displayed in bold text in the address list, with the uppercase **SMTP** value in the **Type** column.

- **Add** Click **Add +** to add a new email address for this mailbox. Select one of following address types:
  - **SMTP** This is the default address type. Click this radio button and then type the new SMTP address in the *** Email address** box.
  - **EUM** An EUM (Exchange Unified Messaging) address is used by the Microsoft Exchange Unified Messaging service to locate UM-enabled users within an Exchange organization. EUM addresses consist of the extension number and the UM dial plan for the UM-enabled user. Click this radio button and type the extension number in the **Address/Extension** box. Then click **Browse** and select a dial plan for the user.
  - **Custom address type** Click this button and type one of the supported non-SMTP email

address types in the *** Email address** box.

Note:

With the exception of X.400 addresses, Exchange doesn't validate custom addresses for proper formatting. You must make sure that the custom address you specify complies with the format requirements for that address type.

- **Automatically update email addresses based on the email address policy applied to this recipient** Select this check box if you want the recipient's email addresses to be updated automatically when changes are made to email address policies in your organization. This box is selected by default.

Mailbox Features

Use the **Mailbox Features** section to view or change the following mailbox features and settings:

- **Sharing policy** This box shows the sharing policy applied to the mailbox. A sharing policy controls how users in your organization can share calendar and contact information with users outside your Exchange organization. The Default Sharing Policy is assigned to mailboxes when they are created. To change the sharing policy that's assigned to the user, select a different one from the drop-down list.
- **Role assignment policy** This box shows the role assignment policy assigned to the mailbox. The role assignment policy specifies the role-based access control (RBAC) roles that are assigned to the user and controls which mailbox and distribution group configuration settings users can modify. To change the role assignment policy that's assigned to the user, select a different one from the drop-down list.
- **Retention policy** This box shows the retention policy assigned to the mailbox. A retention policy is a group of retention tags that are applied to the user's mailbox. The tags allow you to control how long to keep items in users' mailboxes and define which action to take on items that have reached a certain age. A retention policy isn't assigned to mailboxes when they are created. To assign a retention policy to the user, select one from the drop-down list.
- **Address Book policy** This box shows the address book policy applied to the mailbox. An address book policy allows you to segment users into specific groups to provide customized views of the address book. To apply or change the address book policy that's applied to the mailbox, select one from the drop-down list.
- **Unified Messaging** This feature is disabled by default. When you enable Unified Messaging (UM) the user will be able to use your organization's UM features and a default set of UM properties are applied to the user. Click **Enable** to enable UM for the mailbox. For information about how to enable UM, see Enable a user for voice mail.

Note:

A UM dial plan and a UM mailbox policy must exist before you can enable UM.

- **Mobile Devices** Use this section to view and change the settings for Exchange ActiveSync, which is enabled by default. Exchange ActiveSync enables access to an Exchange mailbox from a mobile device. Click **Disable Exchange ActiveSync** to disable this feature for the mailbox.
- **Outlook Web App** This feature is enabled by default. Outlook Web App provides access to an

Exchange mailbox via a web browser. Click **Disable** to disable Outlook Web App for the mailbox. Click **Edit details** to add or change an Outlook Web App mailbox policy for the mailbox.

- **IMAP** This feature is enabled by default. Click **Disable** to disable IMAP for the mailbox.
- **POP3** This feature is enabled by default. Click **Disable** to disable POP3 for the mailbox.
- **MAPI** This feature is enabled by default. MAPI enables access to an Exchange mailbox from a MAPI client such as Outlook. Click **Disable** to disable MAPI for the mailbox.
- **Litigation hold** This feature is disabled by default. Litigation hold preserves deleted mailbox items and records changes made to mailbox items. Deleted items and all instances of changed items are returned in a discovery search. Click **Enable** to put the mailbox on litigation hold. If the mailbox is on litigation hold, click **Disable** to remove the litigation hold. If the mailbox is on litigation hold, click **Edit details** to view and change the following litigation hold settings:
  - **Hold date** This read-only box indicates date and time when the mailbox was put on litigation hold.
  - **Put on hold by** This read-only box indicates the user who put the mailbox on litigation hold.
  - **Note** Use this box to notify the user about the litigation hold, explain why the mailbox is on litigation hold, or provide additional guidance to the user, such as informing them that the litigation hold won't affect their day-to-day use of email.
  - **URL** Use this box to provide a URL to a website that provides information or guidance about the litigation hold on the mailbox.

Note:

The text from these boxes appears in the user's mailbox only if they're using Outlook 2010 or later versions. It doesn't appear in Outlook Web App or other email clients. To view the text from the Note and URL boxes in Outlook, click the **File** tab and, on the **Info** page, under **Account Settings**, you'll see the litigation hold comment.

- **Archiving** If an archive mailbox doesn't exist for the user, this feature is disabled. To enable an archive mailbox, click **Enable**. If the user has an archive mailbox, the size of the archive mailbox and usage statistics are displayed. Click **Edit details** to view and change the following archive mailbox settings:
  - **Status** This read-only box indicates whether an archive mailbox exists.
  - **Database** This read-only box shows the name of the mailbox database that hosts the archive mailbox.
  - **Name** Type the name of the archive mailbox in this box. This name is displayed under the folder list in Outlook or Outlook Web App.
  - **Quota usage** This read-only area shows the total size of the archive mailbox and the percentage of the total archive mailbox quota that has been used.
  - **Quota value (GB)** This box shows the total size of the archive mailbox. To change the size, type a new value in the box or select a value from the drop-down list.
  - **Issue warning at (GB)** This box shows the maximum storage limit for the archive mailbox before a warning is issued to the user. If the archive mailbox size reaches or exceeds the value specified, Exchange sends a warning message to the user. To change this limit, type a new value in the box or select a value from the drop-down list.
- **Delivery Options** Use Delivery Options to forward email messages sent to the user to another

recipient and to set the maximum number of recipients that the user can send a message to. Click **Edit details** to view and change these settings.

- **Forwarding address** Select the **Enable forwarding** check box and then click **Browse** to display the **Select Mail User and Mailbox** page. Use this page to select a recipient to whom you want to forward all email messages that are sent to this mailbox. Messages will be delivered to both the linked mailbox and the forwarding address.
- **Recipient limit** This setting controls the maximum number of recipients the user can send a message to. Select the **Maximum recipients** check box to limit the number of recipients allowed on the To:, Cc:, and Bcc: lines of an email message, and then specify the maximum number of recipients.

Note:

For on-premises Exchange organizations, the recipient limit is unlimited. For Exchange Online organizations, the limit is 500 recipients.

- **Message Size Restrictions** These settings control the size of messages that the user can send and receive. Click **Edit details** to view and change the maximum size for sent and received messages.
  - **Sent messages** To specify a maximum size for messages sent by this user, select the **Maximum message size (KB)** check box and type a value in the box. The message size must be between 0 and 2,097,151 KB. If the user sends a message larger than the specified size, the message will be returned to the user with a descriptive error message.
  - **Received messages** To specify a maximum size for messages received by this user, select the **Maximum message size (KB)** check box and type a value in the box. The message size must be between 0 and 2,097,151 KB. If the user receives a message larger than the specified size, the message will be returned to the sender with a descriptive error message.
- **Message Delivery Restrictions** These settings control who can send email messages to this user. Click **Edit details** to view and change these restrictions.
  - **Accept messages from** Use this section to specify who can send messages to this user.
    - **All senders** Select this option to specify that the user can accept messages from all senders. This includes both senders in your Exchange organization and external senders. This option is selected by default. This option includes external users only if you clear the **Require that all senders are authenticated** check box. If you select this check box, messages from external users will be rejected.
    - **Only senders in the following list** Select this option to specify that the user can accept messages only from a specified set of senders in your Exchange organization. Click **Add** to display the **Select Recipients** page, which displays a list of all recipients in your Exchange organization. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search**.
    - **Require that all senders are authenticated** Select this option to prevent anonymous users from sending messages to the user.
  - **Reject messages from** Use this section to block people from sending messages to this user.
    - **No senders** Select this option to specify that the mailbox won't reject messages from any

senders in the Exchange organization. This option is selected by default.

- **Senders in the following list** Select this option to specify that the mailbox will reject messages from a specified set of senders in your Exchange organization. Click **Add** to display the **Select Recipient** page, which displays a list of all recipients in your Exchange organization. Select the recipients you want to reject messages from, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search**.

Member Of

Use the **Member Of** section to view a list of the distribution groups or security groups to which this user belongs. You can't change membership information on this page. Note that the user may match the criteria for one or more dynamic distribution groups in your organization. However, dynamic distribution groups aren't displayed on this page because their membership is calculated each time they're used.

MailTip

Use the **MailTip** section to add a MailTip to alert users of potential issues if they send a message to this recipient. A MailTip is text that's displayed in the InfoBar when a recipient is added to the To, Cc, or Bcc lines of a new email message.

Note:
MailTips can include HTML tags, but scripts aren't allowed. The length of a custom MailTip can't exceed 175 displayed characters. HTML tags aren't counted in the limit.

Mailbox Delegation

Use the **Mailbox Delegation** section to assign permissions to other users (also called *delegates*) to allow them to sign in to the user's mailbox or send messages on behalf of the user. You can assign the following permissions:

- **Send As** This permission allows users other than the mailbox owner to use the mailbox to send messages. After this permission is assigned to a delegate, any message that a delegate sends from this mailbox will appear as if it was sent by the mailbox owner. However, this permission doesn't allow a delegate to sign in to the user's mailbox.
- **Send on Behalf Of** This permission also allows a delegate to use this mailbox to send messages. However, after this permission is assigned to a delegate, the **From:** address in any message sent by the delegate indicates that the message was sent by the delegate on behalf of the mailbox owner.
- **Full Access** This permission allows a delegate to sign in to the user's mailbox and view the contents of the mailbox. However, after this permission is assigned to a delegate, the delegate can't send messages from the mailbox. To allow a delegate to send email from the user's mailbox, you still have to assign the delegate the Send As or the Send on Behalf Of permission.

To assign permissions to delegates, click **Add** under the appropriate permission to display the **Select Recipient** page, which displays a list of all recipients in your Exchange organization that can

be assigned the permission. Select the recipients you want assign delegate permissions to, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search**.

Use the Shell to change linked mailbox properties

Use the **Get-Mailbox** and **Set-Mailbox** cmdlets to view and change properties for linked mailboxes. One advantage of using the Shell is the ability to change the properties for multiple linked mailboxes. For information about what parameters correspond to mailbox properties, see the following topics:

- Get-Mailbox
- Set-Mailbox

Here are some examples of using the Shell to change linked mailbox properties.

This example uses the **Get-Mailbox** command to find all the linked mailboxes in the organization.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'LinkedMailbox')}
```

This example uses the **Set-Mailbox** command to limit the number of recipients allowed on the To:, Cc:, and Bcc: lines of an email message to 500. This limit applies to all linked mailboxes in the organization.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'LinkedMailbox')} | Set-Mailbox
-RecipientLimits 500
```

This example changes the linked master account in the fabrikam.com account forest that is associated with a linked mailbox in an Exchange forest.

```
Set-Mailbox -Identity "Ayla Kol" -LinkedDomainController
DC1.fabrikam.com -LinkedMasterAccount "fabrikam\robinw" -
LinkedCredential:(Get-Credential fabrikam\administrator)
```

How do you know this worked?

To verify that you have successfully changed properties for a linked mailbox, do the following:

- In the EAC, select the linked mailbox and then click **Edit** to view the property or feature that you changed. Depending on the property that you changed, it might be displayed in the Details pane for the selected mailbox.
- In the Shell, use the **Get-Mailbox** cmdlet to verify the changes. One advantage of using the Shell is that you can view multiple properties for multiple linked mailboxes. In the example above where the recipient limit was changed, running the following command will verify the new value.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'LinkedMailbox')} | fl
Name,RecipientLimits
```

For the example above where the linked master account was changed, run the following command to verify the new value.

```
Get-Mailbox "Ayla Kol" | fl LinkedMasterAccount
```

Manage Distribution Groups

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-21*

Use the Exchange Administration Center (EAC) or the Exchange Management Shell to create a new distribution group in your Exchange organization or to mail-enable an existing group in Active Directory.

There are two types of groups that can be used to distribute messages:

- *Mail-enabled universal distribution groups* (also called *distribution groups*) can be used only to distribute messages.
- *Mail-enabled universal security groups* (also called *security groups*) can be used to distribute messages as well as to grant access permissions to resources in Active Directory. For more information, see Manage mail-enabled security groups.

It's important to note the terminology differences between Active Directory and Exchange. In Active Directory, a distribution group refers to any group that doesn't have a security context, whether it's mail-enabled or not. In contrast, in Exchange, all mail-enabled groups are referred to as distribution groups, whether they have a security context or not.

Note:
You can create or mail-enable only universal distribution groups. To convert a domain-local or a global group to a universal group, you can use the Set-Group cmdlet using the Shell. You may have mail-enabled groups that were migrated from previous versions of Exchange that are not universal groups. You can use the EAC or the Shell to manage these groups

What do you need to know before you begin?

- Estimated time to complete: 2 to 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

- If your organization has configured a group naming policy, it's applied only to groups created by users. When you or other administrators use the EAC to create distribution groups, the group naming policy is ignored and isn't applied to the group name. However, if you use the Shell to create or rename a distribution group, the policy is applied unless you use the *IgnoreNamingPolicy* parameter to override the group naming policy. For more information, see:
  - Create a distribution group naming policy
  - Override the distribution group naming policy
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Create a distribution group

Use the EAC to create a distribution group

1. In the EAC, navigate to **Recipients** > **Groups**.
2. Click **New +** > **Distribution group**.
3. On the **New distribution group** page, complete the following boxes:
   - *** Display name** Use this box to type the display name. This name appears in your organization's address book, on the To: line when email is sent to this group, and in the Groups list in the EAC. The display name is required and should be user-friendly so people recognize what it is. It also must be unique in the forest.
   - *** Alias** Use this box to type the name of the alias for the group. The alias can't exceed 64 characters and must be unique in the forest. When a user types the alias in the To: line of an email message, it resolves to the group's display name.
   - **Description** Use this box to describe the group so people know what the purpose of the group is. This description appears in the address book.
   - **Organizational unit** You can select an organizational unit (OU) other than the default (which is the recipient scope). If the recipient scope is set to the forest, the default value is set to the Users container in the Active Directory domain that contains the computer on which the EAC is running. If the recipient scope is set to a specific domain, the Users container in that domain is selected by default. If the recipient scope is set to a specific OU, that OU is selected by default.

To select a different OU, click **Browse**. The dialog box displays all OUs in the forest that are within the specified scope. Select the OU you want, and then click **OK**.

   - *** Owners** By default, the person who creates a group is the owner. All groups must have at least one owner. You can add owners by clicking **Add +**.
   - **Members** Use this section to add members and to specify whether approval is required for people to join or leave the group.

Group owners don't have to be members of the group. Use **Add group owners as members** to add or remove the owners as members.

To add members to the group, click **Add +**. When you've finished adding members, click **OK** to return to the **New distribution group** page.

Under **Choose whether owner approval is required to join the group**, specify whether approval is required for people to join the group. Select one of the following settings:

- **Open: Anyone can join this group without being approved by the group owners** This is the default setting.
- **Closed: Members can be added only by the group owners. All requests to join will be rejected automatically**
- **Owner Approval: All requests are manually approved or rejected by the group owners** If you select this option, the group owner or owners will receive an email message requesting approval to join the group.

Under **Choose whether the group is open to leave**, specify whether approval is required for people to leave the group. Select one of the following settings:

- **Open: Anyone can leave this group without being approved by the group owners** This is the default setting.
- **Closed: Members can be removed only by the group owners. All requests to leave will be rejected automatically**

4. When you've finished, click **Save** to create the distribution group.

Note:

By default, new distribution groups require that all senders be authenticated. This prevents external senders from sending messages to distribution groups. To configure a distribution group to accept messages from all senders, you must modify the message delivery restriction settings for that distribution group.

Use the Shell to create a distribution group

This example creates a distribution group with an alias **itadmin** and the name **IT Administrators**. The distribution group is created in the default OU, and anyone can join this group without approval by the group owners.

```
New-DistributionGroup -Name "IT Administrators" -Alias
itadmin -MemberJoinRestriction open
```

For more information about using the Shell to create distribution groups, see New-DistributionGroup.

How do you know this worked?

To verify that you've successfully created a distribution group, do one of the following:

- In the EAC, navigate to **Recipients** > **Groups**. The new distribution group is displayed in the group list. Under **Group Type**, the type is **Distribution group**.
- In the Shell, run the following command to display information about the new distribution group.

```
Get-DistributionGroup <Name> | FL
Name,RecipientTypeDetails,PrimarySmtpAddress
```

Change distribution group properties

Use the EAC to change distribution group properties

1. In the EAC, navigate to **Recipients** > **Groups**.
2. In the list of groups, click the distribution group that you want to view or change, and then click **Edit** .
3. On the group properties page, click one of the following sections to view or change properties.
   - General
   - Ownership
   - Membership
   - Membership approval
   - Delivery management
   - Message approval
   - Email options
   - MailTip
   - Group delegation

General

Use this section to view or change basic information about the group.

- *** Display name** This name appears in the address book, on the To: line when email is sent to this group, and in the Groups list. The display name is required and should be user-friendly so people recognize what it is. It also has to be unique in your domain.

If you've implemented a group naming policy, the display name has to conform to the naming format defined by the policy.

- *** Alias** This is the portion of the email address that appears to the left of the at (@) symbol. If you change the alias, the primary SMTP address for the group will also be changed, and contain the new alias. Also, the email address with the previous alias will be kept as a proxy address for the group.
- **Description** Use this box to describe the group so people know what the purpose of the group is. This description appears in the address book and in the Details pane in the EAC.
- **Hide this group from address lists** Select this check box if you don't want users to see this group in the address book. To send email to this group, a sender has to type the group's alias or email address on the To: or Cc: lines.

Tip:
Consider hiding security groups because they're typically used to assign permissions to group members and not to send email.

- **Organizational unit** This read-only box displays the organizational unit (OU) that contains the

distribution group. You have to use Active Directory Users and Computers to move the group to a different OU.

Ownership

Use this section to assign group owners. The group owner can add members to the group, approve or reject requests to join or leave the group, and approve or reject messages sent to the group. By default, the person who creates a group is the owner. All groups must have at least one owner.

You can add owners by clicking **Add +**. You can remove an owner by selecting the owner and then clicking **Remove −**.

Membership

Use this section to add or remove members. Group owners don't have to be members of the group. Under **Members**, you can add members by clicking **Add +**. You can remove a member by selecting a user in the member list and then clicking **Remove −**.

Membership approval

Use this section to specify whether approval is required for users to join or leave the group.

- **Choose whether owner approval is required to join the group** Select one of the following settings:
  - **Open: Anyone can join this group without being approved by the group owners**
  - **Closed: Members can be added only by the group owners. All requests to join will be rejected automatically**
  - **Owner Approval: All requests are approved or rejected by the group owners** If you select this option, the group owner or owners receive an email requesting approval to join the group.
- **Choose whether the group is open to leave** Select one of the following settings:
  - **Open: Anyone can leave this group without being approved by the group owners**
  - **Closed: Members can be removed only by the group owners. All requests to leave will be rejected automatically**

Delivery management

Use this section to manage who can send email to this group.

- **Only senders inside my organization** Select this option to allow only senders in your organization to send messages to the group. This means that if someone outside of your organization sends an email message to this group, it will be rejected. This is the default setting.
- **Senders inside and outside of my organization** Select this option to allow anyone to send messages to the group.

You can further limit who can send messages to the group by allowing only specific senders to send messages to this group. Click **Add +** and then select one or more recipients. If you add senders to this list, they are the only ones who can send mail to the group. Mail sent by anyone not in the list will be rejected.

To remove a person or a group from the list, select them in the list and then click **Remove −**.

Important:

If you've configured the group to allow only senders inside your organization to send messages to the group, email sent from a mail contact will be rejected, even if they are added to this list.

Message approval

Use this section to set options for moderating the group. Moderators approve or reject messages sent to the group before they reach the group members.

- **Messages sent to this group have to be approved by a moderator** This check box isn't selected by default. If you select this check box, incoming messages are reviewed by the group moderators before delivery. Group moderators can approve or reject incoming messages.
- **Group moderators** To add group moderators, click **Add +**. To remove a moderator, select the moderator, and then click **Remove −**. If you've selected "Messages sent to this group have to be approved by a moderator" and you don't select a moderator, messages to the group are sent to the group owners for approval.
- **Senders who don't require message approval** To add people or groups that can bypass moderation for this group, click **Add +**. To remove a person or a group, select the item, and then click **Remove −**.
- **Select moderation notifications** Use this section to set how users are notified about message approval.
  - **Notify all senders when their messages aren't approved** This is the default setting. Notify all senders, inside and outside your organization, when their message isn't approved.
  - **Notify senders in your organization when their messages aren't approved** When you select this option, only people or groups in your organization are notified when a message that they sent to the group isn't approved by a moderator.
  - **Don't notify anyone when a message isn't approved** When you select this option, notifications aren't sent to message senders whose messages aren't approved by the group moderators.

Email options

Use this section to view or change the email addresses associated with the group. This includes the group's primary SMTP addresses and any associated proxy addresses. The primary SMTP address (also known as the *reply address*) is displayed in bold text in the address list, with the uppercase **SMTP** value in the **Type** column.

- **Add** Click **Add +** to add a new email address for this mailbox. Select one of following address types:
  - **SMTP** This is the default address type. Click this button and then type the new SMTP address in the *** Email address** box.

Note:

To make the new address the primary SMTP address for the group, select the **Make this the reply address** check box.

  - **Custom address type** Click this button and type one of the supported non-SMTP email

address types in the * **Email address** box.

Note:
With the exception of X.400 addresses, Exchange doesn't validate custom addresses for correct formatting. You must make sure that the custom address you specify complies with the format requirements for that address type.

- **Edit** To change an email address associated with the group, select it in the list, and then click **Edit** .

Note:
To make an existing address the primary SMTP address for the group, select the **Make this the reply address** check box.

- **Remove** To delete an email address associated with the group, select it in the list, and then click **Remove** .
- **Automatically update email addresses based on the email address policy applied to this recipient** Select this check box to have the recipient's email addresses automatically updated based on changes made to email address policies in your organization. This box is selected by default.

MailTip

Use this section to add a MailTip to alert users of potential issues if they send a message to this group. A MailTip is text that's displayed in the InfoBar when this group is added to the To, Cc, or Bcc lines of a new email message. For example, you could add a MailTip to large groups to warn potential senders that their message will be sent to lots of people.

Note:
MailTips can include HTML tags, but scripts aren't allowed. The length of a custom MailTip can't exceed 175 displayed characters. HTML tags aren't counted in the limit.

Group delegation

Use this section to assign permissions to a user (called a *delegate*) to allow them to send messages as the group or send messages on behalf of the group. You can assign the following permissions:

- **Send As** This permission allows the delegate to send messages as the group. After this permission is assigned, the delegate has the option to add the group to the **From** line to indicate that the message was sent by the group.
- **Send on Behalf Of** This permission also allows a delegate to send messages on behalf of the group. After this permission is assigned, the delegate has the option to add the group in the **From** line. The message will appear to be sent by the group and will say that it was sent by the delegate on behalf of the group.

To assign permissions to delegates, click **Add** under the appropriate permission to display the **Select Recipient** page, which displays a list of all recipients in your Exchange organization that can be assigned the permission. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and

then clicking **Search**.

Use the Shell to change distribution group properties

Use the **Get-DistributionGroup** and **Set-DistributionGroup** cmdlets to view and change properties for distribution groups. Advantages of using the Shell are the ability to change the properties that aren't available in the EAC and to change properties for multiple groups. For information about which parameters correspond to distribution group properties, see the following topics:

- Get-DistributionGroup
- Set-DistributionGroup

Here are some examples of using the Shell to change distribution group properties.

This example changes the primary SMTP address (also called the reply address) for the Seattle Employees distribution group from employees@contoso.com to sea.employees@contoso.com. Also, the previous reply address will be kept as a proxy address.

```
Set-DistributionGroup "Seattle Employees" -EmailAddresses
SMTP:sea.employees@contoso.com,smtp:employees@contoso.com
```

This example limits the maximum message size that can be sent to all distribution groups in the organization to 10 megabytes (MB).

```
Get-DistributionGroup -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq
'MailUniversalDistributionGroup')} | Set-DistributionGroup
-MaxReceiveSize 10MB
```

This example enables moderation for the distribution group Customer Support and sets the moderator to Amy. In addition, this moderated distribution group will notify senders who send mail from within the organization if their messages aren't approved.

```
Set-DistributionGroup -Identity "Customer Support" -
ModeratedBy "Amy" -ModerationEnabled $true -
SendModerationNotifications 'Internal'
```

This example changes the user-created distribution group Dog Lovers to require the group manager to approve users' requests to join the group. In addition, by using the *BypassSecurityGroupManagerCheck* parameter, the group manager will not be notified that a change was made to the distribution group's settings.

```
Set-DistributionGroup -Identity "Dog Lovers" -
MemberJoinRestriction 'ApprovalRequired' -
BypassSecurityGroupManagerCheck
```

How do you know this worked?

To verify that you've successfully changed properties for a distribution group, do the following:

- In the EAC, select the group and then click **Edit** to view the property or feature that you changed. Depending on the property that you changed, it might be displayed in the Details pane for the selected group.
- In the Shell, use the **Get-DistributionGroup** cmdlet to verify the changes. One advantage of using the Shell is that you can view multiple properties for multiple groups. In the example above where the recipient limit was changed, run the following command to verify the new value.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'UserMailbox')} | fl
Name,RecipientLimits
```

For the example above where the message limits were changed, run this command.

```
Get-Mailbox -OrganizationalUnit "Marketing" | fl
Name,IssueWarningQuota,ProhibitSendQuota,ProhibitSendReceiv
eQuota,UseDatabaseQuotaDefaults
```

Create a distribution group naming policy

Exchange Server 2013 > Recipients > Manage Distribution Groups >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-01*

A *group naming policy* lets you standardize and manage the names of distribution groups created by users in your organization. You can require a specific prefix and suffix be added to the name for a distribution group when it's created, and you can block specific words from being used. This helps you minimize the use of inappropriate words in group names.

A group naming policy:

- Enforces a consistent naming strategy for groups created by users.
- Identifies distribution groups in the shared address book.
- Suggests the function or membership of the group.
- Identifies the type of users who are likely members of the group.
- Identifies the geographic region the group is used in.
- Blocks inappropriate words in group names.

How does a group naming policy work? When a user creates a group, they specify a name in the

Display Name field. After the group is created, Microsoft Exchange applies the group naming policy by adding any prefix or suffix that you've defined in the group naming policy. The full name is displayed in the distribution groups list in the Exchange Administration Center (EAC), the shared address book, and the To:, Cc:, and From: fields in email messages. If a user tries to use a word that you've blocked, they get an error message when they try to save the new group and are asked to remove the blocked word and save the group again.

Here are some examples of a group naming policy. In each, **<Group Name>** is a descriptive name provided by the person who creates the group. Exchange adds the prefixes and suffixes defined by the policy to the display name when the group is created.

- Text strings, with underscore characters, used for a single prefix (**DG**) and suffix (**Users**):

DG_<Group Name>_Users

- Multiple prefixes (**DG** and **Contoso**) and one suffix (**Users**), using text strings:

DG_Contoso_<Group Name>_Users

- An attribute (**Department**) used for the prefix:

Department_<Group Name>

For example, say that your school populates the Department attribute for faculty members. Here's an example of a group name created by a faculty member in the Psychology department:

Psychology_Cognitive201

In this example, the underscore character (_) is provided as the only text string in a second prefix to separate the department name from the group name.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Distribution Groups" entry in the Recipients Permissions topic.
- The maximum length for a group name is 64 characters. This includes the combined number of characters in the prefix, the group name provided by the user, and the suffix.
- The group naming policy is applied only to groups created by users. When you or other administrators use the EAC to create distribution groups, the group naming policy is ignored and not applied to the group name.
- Group names are created without spacing. We recommend that you use an underscore character (_) or some other placeholder between text strings, attributes, and the group name.
- You can use Windows PowerShell to override the group naming policy when you create and edit a distribution group. For more information, see Override the distribution group naming policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the EAC to create a group naming policy

1. In the EAC, select **Groups** > **More ⋯** > **Configure group naming policy**.
2. Under **Group Naming Policy**, configure the prefix by selecting either **Attribute** or **Text** in the pull-down menu.
   - **Attribute** Select the attribute and then click **OK**.
   - **Text** Type the text string and click **OK**.

Notice that the text string that you typed or the attribute you selected is displayed as a hyperlink. Click the hyperlink to change the text string or attribute.

3. Click **Add** to add additional prefixes.
4. For the suffix, in the pull-down menu, select either **Attribute** or **Text**, and configure the suffix.
5. Click **Add** to add additional suffixes.

After you add a prefix or suffix, notice that a preview of the group naming policy is displayed.

6. To delete a prefix or suffix from the policy, click **Remove −**.
7. Click **Blocked Words** to add or remove blocked words.
   - To add a word to the list, type the word to block and click **Add +**.
   - To remove a word from the list, select it and click **Remove**.
   - To edit an existing blocked word, select it and click **Edit**.
8. When you are finished, click **Save**.

How do you know this worked?

To verify that you've successfully created a group naming policy, do the following:

- In the EAC, select **Groups** > **More** > **Configure group naming policy**.

On the **Group naming policy** page, the group naming policy that you defined is displayed under **Preview of policy**.

- In Windows PowerShell, run the following command to display the group naming policy.

```
Get-OrganizationConfig | FL DistributionGroupNamingPolicy
```

Override the distribution group naming policy

Exchange Server 2013 > Recipients > Manage Distribution Groups >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-13*

The group naming policy for distribution groups is applied only to groups created by users. When

you or other administrators use the Exchange Administration Center (EAC) to create distribution groups, the group naming policy is ignored and not applied to the group name.

However, if you use the Exchange Management Shell to create or rename a distribution group, the group naming policy is applied to groups created by administrators unless you use the *IgnoreNamingPolicy* parameter to override the group naming policy.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Distribution Groups" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to override the group naming policy when you create a new group

To override the group naming policy, run the following command.

```
New-DistributionGroup -Name <Group Name> -
IgnoreNamingPolicy
```

For example, if the group naming policy for your organization is **DG_<Group Name>_Users**, run the following command to create a group named **All Administrators**.

```
New-DistributionGroup -Name "All Administrators" -
IgnoreNamingPolicy
```

When Microsoft Exchange creates this group, it uses **All Administrators** for both the *Name* and *DisplayName* parameters.

Use the Shell to override the group naming policy when you rename a group

To override the group naming policy when you rename an existing group with the Shell, run the following command.

```
Set-DistributionGroup -Identity <Old Group Name> -Name <New Group Name> -DisplayName <New Group Name> -IgnoreNamingPolicy
```

For example, let's say you created a group naming policy late one night and the next morning you realized you misspelled the text string in the prefix. The next morning, you see that a new group has already been created with the misspelled prefix. You can fix the group naming policy in the EAC, but you have to use the Shell to rename the group with the misspelled name. Run the following command.

```
Set-DistributionGroup -Identity "Goverment_Contracts_NWRegion" -Name "Government_ContractEstimates_NWRegion" -DisplayName "Government_ContractEstimates_NWRegion" -IgnoreNamingPolicy
```

Important:

Be sure to include the *DisplayName* parameter when you rename a group. If you don't, the old name is still displayed in the shared address book on the To:, Cc:, and From: lines in email messages.

How do you know this worked?

To verify that you've successfully created or renamed a distribution group that ignores the group naming policy, run the following commands.

```
Get-DistributionGroup <Name> | FL DisplayName

Get-OrganizationConfig | FL DistributionGroupNamingPolicy
```

If the format of the display name for the group is different than the one enforced by your organization's group naming policy, it worked.

Manage mail-enabled security groups

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-15*

A mail-enabled security group can be used to distribute messages as well as to grant access permissions to resources in Active Directory. For more information, see Recipients.

What do you need to know before you begin?

- Estimated time to complete: 2 to 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Create a mail-enabled security group

Use the EAC to create a security group

1. In the EAC, navigate to **Recipients** > **Groups**.
2. Click **New +** > **Security group**.
3. On the **New security group** page, complete the following fields:
   - *** Display name** Use this box to type the display name. This name appears in the shared address book, on the To: line when email is sent to this group, and in the Groups list in the EAC. The display name is required and should be user-friendly so people recognize what it is. It also must be unique in the forest.

Note:

If a group naming policy is applied, you must follow the naming constraints enforced for your organization. For more information, see Create a distribution group naming policy. If you want to override your organization's group naming policy, see Override the distribution group naming policy.

   - *** Alias** Use this box to type the alias for the security group. The alias can't exceed 64 characters and must be unique in the forest. When a user types the alias on the To: line of an email message, it resolves to the group's display name.
   - **Description** Use this box to describe the security group so people know what the purpose of the group is.
   - **Organizational unit** You can select an organizational unit (OU) other than the default (which is the recipient scope). If the recipient scope is set to the forest, the default value is set to the Users container in the Active Directory domain that contains the computer on which the EAC is running. If the recipient scope is set to a specific domain, the Users container in that domain is

selected by default. If the recipient scope is set to a specific OU, that OU is selected by default. To select a different OU, click **Browse**. The dialog box displays all OUs in the forest that are within the specified scope. Select the desired OU, and then click **OK**.

- *** Owners** By default, the person who creates a group is the owner. All groups must have at least one owner. You can add owners by clicking **Add**.
- **Members** Use this section to add members and to specify whether approval is required for people to join or leave the group.

Group owners don't have to be members of the group. Use **Add group owners as members** to add or remove the owners as members.

To add members to the group, click **Add +**. When you've finished adding members, click **OK** to return to the **New security group** page.

Select the **Owner approval is required** check box if you want the group owners to receive user requests to join the group. If you select this option, members can only be removed by the group owners.

4. When you've finished, click **Save** to create the security group.

Note:

By default, all new mail-enabled security groups require that all senders be authenticated. This prevents external senders from sending messages to mail-enabled security groups. To configure a mail-enabled security group to accept messages from all senders, you must modify the message delivery restriction settings for that group.

Use the Shell to create a security group

This example creates a security group with an alias fsadmin and the name File Server Managers. The security group is created in the default OU, and anyone can join this group with approval by the group owners.

```
New-DistributionGroup -Name "File Server Managers" -Alias
fsadmin -Type security
```

For more information about using the Shell to create mail-enabled security groups, see New-DistributionGroup.

How do you know this worked?

To verify that you've successfully created a mail-enabled security group, do one of the following:

- In the EAC, navigate to **Recipients** > **Groups**. The new mail-enabled security group is displayed in the group list. Under **Group Type**, the type is **Security group**.
- In the Shell, run the following command to display information about the new mail-enabled security group.

```
Get-DistributionGroup <Name> | FL
Name,RecipientTypeDetails,PrimarySmtpAddress
```

Change mail-enabled security group properties

Use the EAC to change mail-enabled security group properties

1. In the EAC, navigate to **Recipients** > **Groups**.
2. In the list of groups, click the security group that you want to view or change, and then click **Edit** .
3. On the group properties page, click one of the following sections to view or change properties.
   - General
   - Ownership
   - Membership
   - Membership approval
   - Delivery management
   - Message approval
   - Email options
   - MailTip
   - Group delegation

General

Use this section to view or change basic information about the group.

- *** Display name** This name appears in the address book, on the To: line when email is sent to this group, and in the Groups list. The display name is required and should be user-friendly so people recognize what it is. It also has to be unique in your domain.
- *** Alias** This is the portion of the email address that appears to the left of the at (@) symbol. If you change the alias, the primary SMTP address for the group will also be changed, and contain the new alias. Also, the email address with the previous alias will be kept as a proxy address for the group.
- **Description** Use this box to describe the group so people know what the purpose of the group is. This description appears in the address book and in the Details pane in the EAC.
- **Hide this group from address lists** Select this check box if you don't want users to see this group in the address book. If this check box is selected, a sender has to type the group's alias or email address on the To: or Cc: lines to send mail to the group.

Tip:
Consider hiding security groups because they're typically used to assign permissions to group members and not to send email.

- **Organizational unit** This read-only box displays the organizational unit (OU) that contains the security group. You have to use Active Directory Users and Computers to move the group to a different OU.

Ownership

Use this section to assign group owners. The group owner can add members to the group, and approve or reject requests to join the group. By default, the person who creates a group is the

owner. All groups must have at least one owner.

You can add owners by clicking **Add +**. You can remove an owner by selecting the owner and then clicking **Remove −**.

Membership

Use this section to add or remove members. Group owners don't have to be members of the group. Under **Members**, you can add members by clicking **Add +**. You can remove a member by selecting a user in the member list and then clicking **Remove −**.

Membership approval

Use this section to specify whether owner approval is required for users to join the group. If you select the **Owner approval is required** check box, the group owner or owners receive an email requesting approval to join the group. As previously mentioned, only owners can remove members from the group.

Delivery management

Use this section to manage who can send email to this group.

- **Only senders inside my organization** Select this option to allow only senders in your organization to send messages to the group. This means that if someone outside of your organization sends an email message to this group, it will be rejected. This is the default setting.
- **Senders inside and outside of my organization** Select this option to allow anyone to send messages to the group.

You can further limit who can send messages to the group by allowing only specific senders to send messages to this group. Click **Add +** and then select one or more recipients. If you add senders to this list, they are the only ones who can send mail to the group. Mail sent by anyone not in the list will be rejected.

To remove a person or a group from the list, select them in the list and then click **Remove −**.

Important:
If you've configured the group to allow only senders inside your organization to send messages to the group, email sent from a mail contact will be rejected, even if they're added to this list.

Message approval

Use this section to set options for moderating the group. Moderators approve or reject messages sent to the group before they reach the group members.

- **Messages sent to this group have to be approved by a moderator** This check box isn't selected by default. If you select this check box, incoming messages will be reviewed by the group moderators before delivery. Group moderators can approve or reject incoming messages.
- **Group moderators** To add group moderators, click **Add +**. To remove a moderator, select the moderator, and then click **Remove −**. If you've selected "Messages sent to this group have to be

approved by a moderator" and you don't select a moderator, messages to the group will be sent to the group owners for approval.

- **Senders who don't require message approval** To add people or groups that can bypass moderation for this group, click **Add +**. To remove a person or a group, select the item, and then click **Remove −**.
- **Select moderation notifications** Use this section to set how users are notified about message approval.
  - **Notify all senders when their messages aren't approved** This is the default setting. Senders inside and outside your organization will be notified when their messages aren't approved.
  - **Notify senders in your organization when their messages aren't approved** When you select this option, only people or groups in your organization are notified when a message that they sent to the group isn't approved by a moderator.
  - **Don't notify anyone when a message isn't approved** When you select this option, notifications aren't sent to message senders whose messages aren't approved by the group moderators.

Email options

Use this section to view or change the email addresses associated with the group. This includes the group's primary SMTP addresses and any associated proxy addresses. The primary SMTP address (also known as the *reply address*) is displayed in bold text in the address list, with the uppercase **SMTP** value in the **Type** column.

- **Add** Click **Add +** to add a new email address for this mailbox. Select one of following address types:
  - **SMTP** This is the default address type. Click this button and then type the new SMTP address in the *** Email address** box.

Note:

To make the new address the primary SMTP address for the group, select the **Make this the reply address** check box. This check box is displayed only when the **Automatically update email addresses based on the email address policy applied to this recipient** check box isn't selected.

  - **Custom address type** Click this button and type one of the supported non-SMTP email address types in the *** Email address** box.

Note:

With the exception of X.400 addresses, Exchange doesn't validate custom addresses for correct formatting. You must make sure that the custom address you specify complies with the format requirements for that address type.

- **Edit** To change an email address associated with the group, select it in the list, and then click **Edit** ✎.

Note:

To make an existing address the primary SMTP address for the group, select the **Make this the reply address** check box. As previously mentioned, this check box is displayed only when the **Automatically update email addresses based on the email address policy applied**

to this recipient check box isn't selected.

- **Remove** To delete an email address associated with the group, select it in the list, and then click **Remove** −.
- **Automatically update email addresses based on the email address policy applied to this recipient** Select this check box to have the recipient's email addresses automatically updated based on changes made to email address policies in your organization. By default, this box is selected.

MailTip

Use this section to add a MailTip to alert users of potential issues before they send a message to this group. A MailTip is text that's displayed in the InfoBar when this group is added to the To, Cc, or Bcc lines of a new email message. For example, you could add a MailTip to large groups to warn potential senders that their message will be sent to lots of people.

Note:

MailTips can include HTML tags, but scripts aren't allowed. The length of a custom MailTip can't exceed 175 displayed characters. HTML tags aren't counted in the limit.

Group delegation

Use this section to assign permissions to a user (called a *delegate*) to allow them to send messages as the group or send messages on behalf of the group. You can assign the following permissions:

- **Send As** This permission allows the delegate to send messages as the group. After this permission is assigned, the delegate has the option to add the group to the **From** line to indicate that the message was sent by the group.
- **Send on Behalf Of** This permission also allows a delegate to send messages on behalf of the group. After this permission is assigned, the delegate has the option to add the group in the **From** line. The message will appear to be sent by the group and will say that it was sent by the delegate on behalf of the group.

To assign permissions to delegates, click **Add** under the appropriate permission to display the **Select Recipient** page, which displays a list of all recipients in your Exchange organization that can be assigned the permission. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search** .

Use the Shell to change security group properties

Use the **Get-DistributionGroup** and **Set-DistributionGroup** cmdlets to view and change properties for security groups. Advantages of using the Shell are the ability to change the properties that aren't available in the EAC and to change properties for multiple security groups. For information about which parameters correspond to which distribution group properties, see the following topics:

- Get-DistributionGroup
- Set-DistributionGroup

Here are some examples of using the Shell to change security group properties.

This example displays a list of all security groups in the organization.

```
Get-DistributionGroup -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'MailUniversalSecurityGroup')}
```

This example changes the primary SMTP address (also called the reply address) for the Seattle Administrators security group from admins@contoso.com to seattle.admins@contoso.com. The previous reply address will be kept as a proxy address.

```
Set-DistributionGroup "Seattle Employees" -EmailAddresses
SMTP:sea.admins@contoso.com,smtp:admins@contoso.com
```

This example hides all security groups in the organization from the address book.

```
Get-DistributionGroup -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'MailUniversalSecurityGroup')} |
Set-DistributionGroup -HiddenFromAddressListsEnabled $true
```

How do you know this worked?

To verify that you've successfully changed properties for a security group, do the following:

- In the EAC, select the group and then click **Edit** to view the property or feature that you changed. Depending on the property that you changed, it might be displayed in the Details pane for the selected group.
- In the Shell, use the **Get-DistributionGroup** cmdlet to verify the changes. One advantage of using the Shell is that you can view multiple properties for multiple groups. In the example above where all security groups were hidden from the address book, run the following command to verify the new value.

```
Get-DistributionGroup -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'MailUniversalSecurityGroup')} |
 fl Name,HiddenFromAddressListsEnabled
```

Manage dynamic distribution groups

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-15*

Dynamic distribution groups are mail-enabled Active Directory group objects that are created to expedite the mass sending of email messages and other information within a Microsoft Exchange organization.

Unlike regular distribution groups that contain a defined set of members, the membership list for dynamic distribution groups is calculated each time a message is sent to the group, based on the filters and conditions that you define. When an email message is sent to a dynamic distribution group, it's delivered to all recipients in the organization that match the criteria defined for that group.

Important:

A dynamic distribution group includes any recipient in Active Directory with attribute values that match its filter. If a recipient's properties are modified to match the filter, the recipient could inadvertently become a group member and start receiving messages that are sent to the group. Well-defined, consistent account provisioning processes will reduce the chances of this issue occurring.

What do you need to know before you begin?

- Estimated time to complete: 2 to 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Dynamic distribution groups" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Create a dynamic distribution group

Use the EAC to create a dynamic distribution group

1. In the EAC, navigate to **Recipients** > **Groups** > **New** > **Dynamic distribution group**.
2. On the **New dynamic distribution group** page, complete the following boxes:
   - *** Display name** Use this box to type the display name. This name appears in the shared address book, on the To: line when email is sent to this group, and in the Groups list in the EAC. The display name is required and should be user-friendly so people recognize what it is. It also must be unique in the forest.

Note:

Group naming policy isn't applied to dynamic distribution groups.

- *** Alias** Use this box to type the name of the alias for the group. The alias cannot exceed 64 characters and must be unique in the forest. When a user types the alias in the To: line of an email message, it resolves to the group's display name.
- **Description** Use this box to describe the group so people know what the purpose of the group is. This description appears in the shared address book.
- **Organizational unit** You can select an organizational unit (OU) other than the default (which is the recipient scope). If the recipient scope is set to the forest, the default value is set to the Users container in the Active Directory domain that contains the computer on which the EAC is running. If the recipient scope is set to a specific domain, the Users container in that domain is selected by default. If the recipient scope is set to a specific OU, that OU is selected by default.

To select a different OU, click **Browse**. The dialog box displays all OUs in the forest that are within the specified scope. Select the OU you want, and then click **OK**.

- **Owner** An owner for a dynamic distribution group is optional. You can add owners by clicking **Browse** and then selecting users from the list.

3. Use the **Members** section to specify the types of recipients for the group and set up rules that will determine membership. Select one of the following boxes:
   - **All recipient types** Choose this option to send messages that meet the criteria defined for this group to all recipient types.
   - **Only the following recipient types** Messages that meet the criteria defined for this group will be sent to one or more of the following recipient types:
     - **Users with Exchange mailboxes** Select this check box if you want to include users that have Exchange mailboxes. Users that have Exchange mailboxes are those that have a user domain account and a mailbox in the Exchange organization.
     - **Users with external email addresses** Select this check box if you want to include users that have external email addresses. Users that have external email accounts have user domain accounts in Active Directory, but use email accounts that are external to the organization. This enables them to be included in the global address list (GAL) and added to distribution lists.
     - **Resource mailboxes** Select this check box if you want to include Exchange resource mailboxes. Resource mailboxes allow you to administer company resources through a mailbox, such as a conference room or a company vehicle.
     - **Contacts with external email addresses** Select this check box if you want to include contacts that have external email addresses. Contacts that have external email addresses don’t have user domain accounts in Active Directory, but the external email address is available in the GAL.
     - **Mail-enabled groups** Select this check box if you want to include security groups or distribution groups that have been mail-enabled. Mail-enabled groups are similar to distribution groups. Email messages that are sent to a mail-enabled group account will be delivered to several recipients.
4. Click **Add a rule** to define the criteria for membership in this group.
5. Select one of the following recipient attributes from the drop-down list and provide a value. If the value for the selected attribute matches that value you define, the recipient receives a

message sent to this group.

Attribute	Send message to a recipient if...
Recipient container	The recipient object resides in the specified domain or OU.
State or province	The specified value matches the recipient's State or province property.
Company	The specified value matches the recipient's Company property.
Department	The specified value matches the recipient's Department property.
Custom attributeN (where N is a number from 1 to 15)	The specified value matches the recipient's CustomAttributeN property.

Important:

The values that you enter for the selected attribute must exactly match those that appear in the recipient's properties. For example, if you enter **Washington** for **State or province**, but the value for the recipient's property is **WA**, the condition will not be met. Also, text-based values that you specify aren't case-sensitive. For example, if you specify **Contoso** for the **Company** attribute, messages will be sent to a recipient if this value is **contoso**.

6. In the **Specify words or phrases** window, type the value in the text box. Click **Add** and then click **OK**.
7. To add another rule to define the criteria for membership, click **Add a rule** under the previous rule that you created.

Important:

If you add multiple rules to define membership, a recipient must meet the criteria of each rule to receive a message sent to the group. In other words, each rule is connected with the Boolean operator **AND**.

8. When you've finished, click **Save** to create the dynamic distribution group.

Note:

If you want to specify rules for attributes other than the ones available in the EAC, you must use the Shell to create a dynamic distribution group. Keep in mind that the filter and condition settings for dynamic distribution groups that have custom recipient filters can be managed only by using the Shell. For an example of how to create a dynamic distribution group with a custom query, see the next section on using the Shell to create a dynamic distribution group.

Use the Shell to create a dynamic distribution group

This example creates the dynamic distribution group "Mailbox Users DDG" that contains only mailbox users.

```
New-DynamicDistributionGroup -IncludedRecipients
MailboxUsers -Name "Mailbox Users DDG" -OrganizationalUnit
Users
```

This example creates a dynamic distribution group with a custom recipient filter. The dynamic distribution group contains all mailbox users on a server called Server1.

```
New-DynamicDistributionGroup -Name "Mailbox Users on
Server1" -OrganizationalUnit Users -RecipientFilter
{((RecipientTypeDetails -eq 'UserMailbox' -and ServerName -
eq 'Server1'))}
```

This example creates a dynamic distribution group with a custom recipient filter. The dynamic distribution group contains all mailbox users that have a value of "FullTimeEmployee" in the **CustomAttribute10** property.

```
New-DynamicDistributionGroup -Name "Full Time Employees" -
RecipientFilter {(RecipientTypeDetails -eq 'UserMailbox') -
and (CustomAttribute10 -eq 'FullTimeEmployee')}
```

For detailed syntax and parameter information, see New-DynamicDistributionGroup.

How do you know this worked?

To verify that you've successfully created a dynamic distribution group, do one of the following:

- In the EAC, navigate to **Recipients** > **Groups**. The new dynamic distribution group is displayed in the group list. Under **Group Type**, the type is **Dynamic distribution group**.
- In the Shell, run the following command to display information about the new dynamic distribution group.

```
Get-DynamicDistributionGroup | FL
Name,RecipientTypeDetails,RecipientFilter,PrimarySmtpAddres
s
```

Change dynamic distribution group properties

Use the EAC to change dynamic distribution group properties

1. In the EAC, navigate to **Recipients** > **Groups**.
2. In the list of groups, click the dynamic distribution group that you want to view or change, and then click **Edit** ✎.
3. On the group's properties page, click one of the following sections to view or change properties.
   - General
   - Ownership

- Membership
- Delivery management
- Message approval
- Email options
- MailTip
- Group delegation

General

Use this section to view or change basic information about the group.

- *** Display name** This name appears in the address book, on the To: line when email is sent to this group, and in the Groups list. The display name is required and should be user-friendly so people recognize what it is. It also has to be unique in your domain.
- *** Alias** This is the portion of the email address that appears to the left of the at (@) symbol. If you change the alias, the primary SMTP address for the group will also be changed, and contain the new alias. Also, the email address with the previous alias will be kept as a proxy address for the group.
- **Description** Use this box to describe the group so people know what the purpose of the group is. This description appears in the address book and in the Details pane in the EAC.
- **Hide this group from address lists** Select this check box if you don't want users to see this group in the address book. To send email to this group, a sender has to type the group's alias or email address on the To: or Cc: lines.
- **Organizational unit** This read-only box displays the organizational unit (OU) that contains the dynamic distribution group. You have to use Active Directory Users and Computers to move the group to a different OU.

Ownership

Use this section to assign a group owner. A dynamic distribution group can have only one owner. The group owner appears on the **Managed by** tab of the object in Active Directory Users and Computers.

You can add owners by clicking **Browse** and selecting the owner from the list. To remove the owner, click **Clear** and then click **Save**.**—**.

Membership

Use this section to change the criteria used to determine membership of the group. You can delete or change existing membership rules and add new rules. For procedures that tell you how to do this, see step 3 in the procedures for configuring membership when you use the EAC to create a new dynamic distribution group.

Delivery management

Use this section to manage who can send email to this group.

- **Only senders inside my organization** Select this option to allow only senders in your

organization to send messages to the group. This means that if someone outside your organization sends an email message to this group, it is rejected. This is the default setting.

- **Senders inside and outside of my organization** Select this option to allow anyone to send messages to the group.

You can further limit who can send messages to the group by allowing only specific senders to send messages to this group. Click **Add +** and then select one or more recipients. If you add senders to this list, they are the only ones who can send mail to the group. Mail sent by anyone not in the list will be rejected.

To remove a person or a group from the list, select them in the list and then click **Remove −**.

Important:
If you've configured the group to allow only senders inside your organization to send messages to the group, email sent from a mail contact is rejected, even if they're added to this list.

Message approval

Use this section to set options for moderating the group. Moderators approve or reject messages sent to the group before they reach the group members.

- **Messages sent to this group have to be approved by a moderator** This check box isn't selected by default. If you select this check box, incoming messages are reviewed by the group moderators before delivery. Group moderators can approve or reject incoming messages.
- **Group moderators** To add group moderators, click **Add +**. To remove a moderator, select the moderator, and then click **Remove −**. If you've selected "Messages sent to this group have to be approved by a moderator" and you don't select a moderator, messages to the group are sent to the group owners for approval.
- **Senders who don't require message approval** To add people or groups that can bypass moderation for this group, click **Add +**. To remove a person or a group, select the item, and then click **Remove −**.
- **Select moderation notifications** Use this section to set how users are notified about message approval.
  - **Notify all senders when their messages aren't approved** This is the default setting. Notify all senders, inside and outside your organization, when their message isn't approved.
  - **Notify senders in your organization only when their messages aren't approved** When you select this option, only people or groups in your organization are notified when a message that they sent to the group isn't approved by a moderator.
  - **Don't notify anyone when a message isn't approved** When you select this option, notifications aren't sent to message senders whose messages aren't approved by the group moderators.

Email options

Use this section to view or change the email addresses associated with the group. This includes the group's primary SMTP addresses and any associated proxy addresses. The primary SMTP address (also known as the *reply address*) is displayed in bold text in the address list, with the uppercase

SMTP value in the **Type** column.

- **Add** Click **Add +** to add a new email address for this mailbox. Select one of following address types:
  - **SMTP** This is the default address type. Click this button and then type the new SMTP address in the *** Email address** box.

Note:
To make the new address the primary SMTP address for the group, select the **Make this the reply address** check box.

  - **Custom address type** Click this button and type one of the supported non-SMTP email address types in the *** Email address** box.

Note:
With the exception of X.400 addresses, Exchange doesn't validate custom addresses for proper formatting. You must make sure that the custom address you specify complies with the format requirements for that address type.

- **Edit** To change an email address associated with the group, select it from the list, and then click **Edit** .

Note:
To make an existing address the primary SMTP address for the group, select the **Make this the reply address** check box.

- **Remove** To delete an email address associated with the group, select it from the list, and then click **Remove −**.
- **Automatically update email addresses based on the email address policy applied to this recipient** Select this check box to have the recipient's email addresses automatically updated based on changes made to email address policies in your organization. This box is selected by default.

MailTip

Use this section to add a MailTip to alert users of potential issues before they send a message to this group. A MailTip is text that's displayed in the InfoBar when this group is added to the To, Cc, or Bcc lines of a new email message. For example, you could add a MailTip to large groups to warn potential senders that their message will be sent to lots of people.

Note:
MailTips can include HTML tags, but scripts aren't allowed. The length of a custom MailTip can't exceed 175 displayed characters. HTML tags aren't counted in the limit.

Group delegation

Use this section to assign permissions to a user (called a *delegate*) to allow them to send messages as the group or send messages on behalf of the group. You can assign the following permissions:

- **Send As** This permission allows the delegate to send messages as the group. After this permission is assigned, the delegate has the option to add the group to the **From** line to indicate

that the message was sent by the group.

- **Send on Behalf Of** This permission also allows a delegate to send messages on behalf of the group. After this permission is assigned, the delegate has the option to add the group on the **From** line. The message will appear to be sent by the group and will say that it was sent by the delegate on behalf of the group.

To assign permissions to delegates, click **Add** under the appropriate permission to display the **Select Recipient** page, which displays a list of all recipients in your Exchange organization that can be assigned the permission. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search**.

Use the Shell to change dynamic distribution group properties

Use the **Get-DynamicDistributionGroup** and **Set-DynamicDistributionGroup** cmdlets to view and change properties for dynamic distribution groups. Advantages of using the Shell are the ability to change the properties that aren't available in the EAC and change properties for multiple groups. For information about what parameters correspond to distribution group properties, see the following topics:

- Get-DynamicDistributionGroup
- Set-DynamicDistributionGroup

Here are some examples of using the Shell to change dynamic distribution group properties.

This example changes the following parameters for all dynamic distribution groups in the organization:

- Hide all dynamic distribution groups from the address book
- Set the maximum message size that can be sent to the group to 5MB
- Enable moderation
- Assign the administrator as the group moderator

```
Get-DynamicDistributionGroup -ResultSize unlimited | Set-
DynamicDistributionGroup -HiddenFromAddressListsEnabled
$true -MaxReceiveSize 5MB -ModerationEnabled $true -
ModeratedBy administrator
```

This example adds the proxy SMTP email address, Seattle.Employees@contoso.com, to the All Employees group.

```
Set-DynamicDistributionGroup -Identity "All Employees" -
EmailAddresses SMTP:All.Employees@contoso.com,
smtp:Seattle.Employees@contoso.com
```

How do you know this worked?

To verify that you've successfully changed properties for a dynamic distribution group, do the

following:

- In the EAC, select the group and then click **Edit** to view the property or feature that you changed. Depending on the property that you changed, it might be displayed in the Details pane for the selected group.
- In the Shell, use the **Get-DynamicDistributionGroup** cmdlet to verify the changes. One advantage of using the Shell is that you can view multiple properties for multiple groups. In the first example, you would run the following command to verify the new values.

```
Get-DynamicDistributionGroup -ResultSize unlimited | fl
Name,HiddenFromAddressListsEnabled,MaxReceiveSize,Moderatio
nEnabled,ModeratedBy
```

For the example above where the message limits were changed, run this command.

```
Get-Mailbox -OrganizationalUnit "Marketing" | fl
Name,IssueWarningQuota,ProhibitSendQuota,ProhibitSendReceiv
eQuota,UseDatabaseQuotaDefaults
```

View members of a dynamic distribution group

Exchange Server 2013 > Recipients > Manage dynamic distribution groups >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-13*

Dynamic distribution groups are distribution groups whose membership is based on specific recipient filters rather than a defined set of recipients. Microsoft Exchange provides precanned filters to make it easier to create recipient filters for dynamic distribution groups. A *precanned filter* is a commonly used filter that you can use to meet a variety of recipient-filtering criteria. You can specify the recipient types you want to include in a dynamic distribution group. Additionally, you can also specify a list of conditions that the recipients must meet. You can use the Shell to preview the list of recipients for a dynamic distribution group that uses precanned filters.

You can also use the Exchange Administration Center (EAC) to see how many recipients received the most recent message that was sent to a dynamic distribution group.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "Dynamic distribution groups" entry in the Recipients Permissions topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to preview the list of members of a dynamic distribution group

This example returns the list of members for the dynamic distribution group Full Time Employees. The first command stores the dynamic distribution group object in the variable $FTE. The second command uses the **Get-Recipient** cmdlet to list the recipients that match the criteria defined for the dynamic distribution group.

```
$FTE = Get-DynamicDistributionGroup "Full Time Employees"

Get-Recipient -RecipientPreviewFilter $FTE.RecipientFilter
```

For detailed syntax and parameter information, see Get-DynamicDistributionGroup and Get-Recipient.

Use the EAC to see how many recipients received a message sent to a dynamic distribution group

1. In the EAC, navigate to **Recipients** > **Groups**.
2. Select a dynamic distribution group.

In the details pane under **Membership**, the number of people who received the last message sent to the dynamic distribution group is displayed.

How do you know this worked?

To verify that you've successfully viewed the members of a dynamic distribution group, do one of the following:

- In the Shell, a list of members is returned after you run the previous command to preview a list of dynamic distribution group members. For example, if you created a new user mailbox with properties that match the recipient filter for the dynamic distribution group, this new user should

be displayed in the list of group members.

- The number of people who received the last message sent to the dynamic distribution group is displayed in the details pane in the EAC. If no one has received a message sent to this group, it's possible that a message hasn't been sent to the group or that the group doesn't have any members that meet the conditions of the recipient filter.

Manage mail contacts

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-19*

Mail contacts are mail-enabled directory service objects that contain information about people or organizations that exist outside your Exchange or Exchange Online organization. Each mail contact has an external email address. For more information about mail contacts, see Recipients.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Create a mail contact

Use the EAC to create a mail contact

1. In the EAC, navigate to **Recipients** > **Contacts**.
2. Click **New +** > **Mail contact**.
3. Complete the following boxes on the **New mail contact** page:
   - **First name** Use this box to type the contact's first name.
   - **Initials** Use this box to type the contact's initials.
   - **Last name** Use this box to type the contact's last name.

- *** Display name** Use this box to type a display name for the contact. This is the name that's listed in the contacts list in the EAC and in your organization's address book. By default, this box is populated with the names you enter in the **First name**, **Initials**, and **Last name** boxes. If you didn't use those boxes, you must still type a name in this box because it's required. The name can't exceed 64 characters.
- *** Name** Use this box to type a name for the contact. This is the name that's listed in the directory service. Like the display name, this box is populated by default with the names you enter in the **First name**, **Initials**, and **Last name** boxes. If you didn't use those boxes, you must still type a name in this box because it's required. The name can't exceed 64 characters.
- *** Alias** Use this box to type a unique alias (64 characters or less) for the contact. This box is required.
- *** External email address** Use this box to type the outside email account of the contact. This box is required. Email sent to this contact is forwarded to this email address.
- **Organizational unit** You can select an organizational unit (OU) other than the default, which is the recipient scope. If the recipient scope is set to the forest, the default value is set to the Users container in the domain that contains the computer on which the EAC is running. If the recipient scope is set to a specific domain, the Users container in that domain is selected by default. If the recipient scope is set to a specific OU, that OU is selected by default.

To select a different OU, click **Browse**. The dialog box displays all OUs in the forest that are within the specified scope. Select the OU you want, and then click **OK**.

Note:
The **Organizational unit** box is only available in Exchange Server 2013. It isn't available in Exchange Online.

4. When you've finished, click **Save**.

Use the Shell to create a mail contact

This example creates a mail contact for Debra Garcia in Exchange Server 2013.

```
New-MailContact -Name "Debra Garcia" -ExternalEmailAddress
dgarcia@tailspintoys.com -OrganizationalUnit Users
```

This example creates a mail contact for Alan Shen in Exchange Online.

```
New-MailContact -Name "Alan Shen" -ExternalEmailAddress
alans@fourthcoffee.com
```

This example mail-enables an existing contact named Karen Toh in Exchange Server 2013.

```
Enable-MailContact -Identity "Karen Toh" -
ExternalEmailAddress ktoh@tailspintoys.com
```

How do you know this worked?

To verify that you've successfully created a mail contact, do one of the following:

- In the EAC, navigate to **Recipients** > **Contacts**. The new mail contact is displayed in the contact list. Under **Contact Type**, the type is **Mail contact**.
- In the Shell, run the following command to display information about the new mail contact.

```
Get-MailContact <Name> | FL
Name,RecipientTypeDetails,ExternalEmailAddress
```

Change mail contact properties

Use the EAC to change mail contact properties

1. In the EAC, navigate to **Recipients** > **Contacts**.
2. In the list of mail contacts and mail users, click the mail contact that you want to change the properties for, and then click **Edit** ✎.
3. On the mail contact properties page, click one of the following sections to view or change properties.
   - General
   - Contact Information
   - Organization
   - Email Options (not available in Exchange Online)
   - MailTip

General

Use the **General** section to view or change basic information about the mail contact.

- **First name**, **Initials**, **Last name**
- *** Name** This is the name that's listed in Active Directory. If you change this name, it can't exceed 64 characters.
- *** Display name** This name appears in your organization's address book, on the To and From lines in email, and in the Mailbox list. This name can't contain empty spaces before or after the display name.
- *** Alias** This is the mail contact's alias. If you change it, it must be unique in the organization and must be 64 characters or less.
- *** External email address** This is mail contact's primary SMTP address and their outside email account. Email sent to this contact is forwarded to this email address.
- Click **More options** to display the OU that contains the mail contact account. You have to use Active Directory Users and Computers to move the contact to a different OU.

Contact Information

Use the **Contact Information** section to view or change the recipient's contact information, such as mailing address and telephone numbers. This information is displayed in the address book.

Organization

Use the **Organization** section to record detailed information about the mail contact's role in the organization. This information is displayed in the address book. Also, you can create a virtual organization chart that's accessible from email clients such as Outlook.

- **Title** Use this box to view or change the contact's title.
- **Department** Use this box to view or change the department in which the contact works. You can use this box to create recipient conditions for dynamic distribution groups and address lists.
- **Company** Use this box to view or change the company for which the contact works. You can also use this box to create recipient conditions for dynamic distribution groups.
- **Manager** To add a manager, click **Browse**. In **Select Manager**, select a person, and then click **OK**.
- **Direct reports** You can't modify this box. A *direct report* is a recipient who reports to a specific manager. If you've specified a manager for the recipient, that recipient appears as a direct report in the details of the manager's mailbox. For example, Toby manages Ann and Spencer, who are mail contacts, so Toby is specified in the **Manager** box in the organization properties for Ann and Spencer, and Ann and Spencer appear in the **Direct reports** box in the properties of Toby's mailbox.

Email Options

Use the **Email Options** section to add or remove proxy addresses for the mail contact or edit existing proxy addresses. The mail contact's primary SMTP address is also displayed in this section, but you can't change it. To change it, you have to change the contact's external email address in the **General** section.

Note:

The **Email Options** section is only available in Exchange Server 2013. It's not available in Exchange Online.

MailTip

Use the **MailTip** section to add a MailTip to alert users of potential issues before they send a message to this recipient. A MailTip is text that's displayed in the InfoBar when this recipient is added to the To, Cc, or Bcc lines of a new email message.

Note:

MailTips can include HTML tags, but scripts aren't allowed. The length of a custom MailTip can't exceed 175 displayed characters. HTML tags aren't counted in the limit.

Use the Shell to change mail contact properties

Properties for a mail contact are stored in both Active Directory and Exchange. In general, use the **Get-Contact** and **Set-Contact** cmdlets to view and change organization and contact information properties. Use the **Get-MailContact** and **Set-MailContact** cmdlets to view or change mail-related properties, such as email addresses, the MailTip, custom attributes, and whether the contact is hidden from address lists.

For more information, see the following topics:

- Get-Contact
- Set-Contact
- Get-MailContact
- Set-MailContact

Here are some examples of using the Shell to change mail contact properties.

This example configures the Title, Department, Company, and Manager properties for the mail contact Kai Axford.

```
Set-Contact "Kai Axford" _-Title Consultant -Department
"Public Relations" -Company Fabrikam -Manager "Karen Toh"
```

This example sets the CustomAttribute1 property to a value of PartTime for all mail contacts and hides them from the organization's address book.

```
Get-MailContact | Set-MailContact -CustomAttribute1
PartTime -HiddenFromAddressListsEnabled $true
```

This example sets the CustomAttribute15 property to a value of TemporaryEmployee for all mail contacts in the Public Relations department.

```
Get-Contact -Filter "Department -eq 'Public Relations'" |
Set-MailContact -CustomAttribute15 TemporaryEmployee
```

How do you know this worked?

To verify that you've successfully changed properties for a mail contact, do the following:

- In the EAC, select the mail contact, and then click **Edit** to view the property that you changed.
- In the Shell, use the **Get-Contact** and **Get-MailContact** cmdlets to verify the changes. One advantage of using the Shell is that you can view multiple properties for multiple mail contacts. In the example above where all mail contacts had the CustomAttribute1 property set to PartTime and were hidden from the address book, run the following command to verify the changes.

```
Get-MailContact | Fl
Name,CustomAttribute1,HiddenFromAddressListsEnabled
```

In the example above where the CustomAttribute15 was set for all mail contacts in the Public Relations department, run the following command to verify the changes.

```
Get-Contact -Filter "Department -eq 'Public Relations'" |
Get-MailContact | FL Name,CustomAttribute15
```

Bulk edit mail contacts

You can use the EAC to change selected properties for multiple mail contacts. When you select two

or more mail contacts from the contacts list in the EAC, the properties that can be bulk edited are displayed in the Details pane. When you change one of these properties, the change is applied to all selected recipients.

When you bulk edit mail contacts, you can change the following property areas:

- **Contact Information** Change shared properties such as street, postal code, and city name.
- **Organization** Change shared properties such as department name, company name, and the manager that the selected mail contacts or mail users report to.

Use the EAC to bulk edit mail contacts

1. In the EAC, navigate to **Recipients** > **Contacts**.
2. In the list of contacts, select two or more mail contacts. You can't bulk edit a combination of mail contacts and mail users.

> **Tip:**
> You can select multiple adjacent mail contacts by holding down the Shift key and clicking the first mail contact, and then clicking the last mail contact you want to edit. You can also select multiple mail contacts by holding down the Ctrl key and clicking each one that you want to edit.

3. In the Details pane, under **Bulk Edit**, click **Update** under **Contact Information** or **Organization**.
4. Make the changes on the properties page and then save your changes.

How do you know this worked?

To verify that you've successfully bulk edited mail contacts, do one of the following:

- In the EAC, select each of the mail contacts that you bulk edited, and then click **Edit** to view the properties that you changed.
- In the Shell, use the **Get-Contact** cmdlet to verify the changes. For example, say you used the bulk edit feature in the EAC to change the manager and the office for all mail contacts from a vendor company named A. Datum Corporation. To verify these changes, you could run the following command in the Shell.

```
Get-Contact -ResultSize unlimited -Filter {(Company -eq
'Adatum')} | fl Name,Office,Manager
```

Enable or disable email for a mail contact

Exchange Server 2013 > Recipients > Manage mail contacts >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-05*

You can disable email for an existing mail contact in your Exchange organization. When you disable email for a mail contact, it's removed from Exchange and from your organization's address book. If the mail contact is a member of a distribution group, the contact no longer receives mail sent to the group. Also, the Exchange attributes are removed from the mail-enabled contact object in Active Directory, but the contact and its non-Exchange attributes (such as contact and organization information) are retained in Active Directory.

After you disable email for a mail contact, you can mail-enable the contact again by using the **Enable-MailContact** cmdlet in the Shell. You can also use this cmdlet to mail-enable any Active Directory contact.

For additional management tasks related to mail contacts, see Manage mail contacts.

What do you need to know before you begin?

- Estimated time to complete each procedure: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mail contacts" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Disable email for a mail contact

As previously stated, when you disable email for a mail contact, the Exchange attributes are removed from the corresponding Active Directory contact object, but the contact is retained. The mail contact is removed from the list of mail contacts in the EAC, but you can view and manage the corresponding Active Directory contact object by using Active Directory Users and Computers or by using the **Get-Contact** and **Set-Contact** cmdlets in the Shell.

Use the EAC to disable email for a mail contact

1. In the EAC, navigate to **Recipients** > **Contacts**.
2. In the list of contacts, click the mail contact for which you want to disable email.
3. Click **More** ··· and then click **Disable**.
4. A warning will appear asking if you're sure you want to disable the selected mail contact. Click **Yes** to disable it.

The mail contact will be removed from the contacts list.

Use the Shell to disable email for a mail contact

This example disables email for the mail contact Neil Black.

```
Disable-MailContact -Identity "Neil Black"
```

For detailed syntax and parameter information, see Disable-MailContact.

How do you know this worked?

To verify that you've successfully disabled email for a mail contact, do one of the following:

1. In the EAC, navigate to **Recipients** > **Contacts** and verify that the mail contact is no longer listed.
2. In Active Directory Users and Computers, right-click the contact, and then click **Properties**. On the **General** tab, notice that the **E-mail** box is blank. This verifies that the contact isn't mail-enabled.
3. In the Shell, run the following command.

```
Get-MailContact
```

The contact that you disabled email for won't be returned in the results because this cmdlet only returns mail-enabled contacts.

4. In the Shell, run the following command.

```
Get-Contact
```

The contact that you disabled email for is returned in the results because this cmdlet returns all Active Directory contact objects.

Use the Shell to mail-enable contacts

You can use the **Enable-MailContact** cmdlet to mail-enable existing Active Directory contacts. You can mail-enable a single contact or use a CSV file to mail-enable multiple contacts.

Use the Shell to mail-enable a single contact

This example mail-enables the contact Rene Valdes. You must provide an external email address.

```
Enable-MailContact -Identity "Rene Valdes" -
ExternalEmailAddress renev@tailspintoys.com
```

Use the Shell and a CSV file to mail-enable multiple contacts

When you're mail-enabling contacts in bulk, you first export the list of contacts that aren't mail-enabled to a CSV (comma-separated values) file, and then add the external email addresses to the CSV file by using a text editor such as Notepad, or a spreadsheet application such as Microsoft Excel. Then you use the updated CSV file in the Shell command to mail-enable the contacts listed in the CSV file.

1. Run the following command to export a list of existing contacts that aren't mail-enabled to a file

on the administrator's desktop named Contacts.csv.

```
Get-Contact | Where { $_.RecipientType -eq "Contact" } |
Out-File "C:\Users\Administrator\Desktop\Contacts.csv"
```

The resulting file will be similar to the following file.

```
Name
Walter Harp
James Alvord
Rainer Witt
Susan Burk
Ian Tien
...
```

2. Add a column heading named **EmailAddress** and then add an email address for each contact in the file. The name and external email address for each contact must be separated by a comma. The updated CSV file should look similar to the following file.

```
Name,EmailAddress
James Alvord,james@contoso.com
Susan Burk,sburk@tailspintoys.com
Walter Harp,wharp@tailspintoys.com
Ian Tien,iant@tailspintoys.com
Rainer Witt,rainerw@fourthcoffee.com
...
```

3. Run the following command to use the data in the CSV file to mail-enable the contacts listed in the file.

```
Import-CSV C:\Users\Administrator\Desktop\Contacts.csv |
ForEach-Object {Enable-MailContact -Identity $_.Name -
ExternalEmailAddress $_.EmailAddress}
```

The command results display information about the new mail-enabled contacts.

How do you know this worked?

To verify that you've successfully mail-enabled Active Directory contacts, do one of the following:

- In the EAC, navigate to **Recipients** > **Contacts**. New mail contacts are displayed in the contact list. Under **Contact Type**, the type is **Mail contact**.

Note:
You may have to click **Refresh** to display new mail contacts.

- In the Shell, run the following command to display information about new mail contacts.

```
Get-MailContact | Format-Table
Name,RecipientTypeDetails,ExternalEmailAddress
```

Manage mail users

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-04*

Mail users are similar to mail contacts. Both have external email addresses and both contain information about people outside your Exchange or Exchange Online organization that can be displayed in the shared address book and other address lists. However, unlike a mail contact, a mail user has logon credentials in your Exchange or Office 365 organization and can access resources. For more information, see Recipients.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Create a mail user

Use the EAC to create a mail user

1. In the EAC, navigate to **Recipients** > **Contacts** > **New** > **Mail user**.
2. On the **New mail user** page, in the *** Alias** box, type the alias for the mail user. The alias can't exceed 64 characters and must be unique in the forest. This box is required.
3. Do one of the following to specify the email address type for the mail user:
   - To specify an SMTP email address for the mail user's external email address, click **SMTP**.

Note:

Exchange validates SMTP addresses for correct formatting. If your entry is inconsistent with the SMTP format, an error message will be displayed when you click **Save** to create the mail user.

- To specify a custom address type, click the option button and then type the custom address type. For example, you can specify an X.500, GroupWise, or Lotus Notes address.

4. In the *** External email address** box, type the mail user's external email address. Email sent to this mail user is forwarded to this email address. This box is required.
5. Select one of the following options:
   - **Existing user** Select to mail-enable an existing user.

Click **Browse** to open the **Select User – Entire Forest** dialog box. This dialog box displays a list of user accounts in the organization that aren't mail-enabled or don't have mailboxes. Select the user account you want to mail-enable, and then click **OK**. If you select this option, you don't have to provide user account information because this information already exists in Active Directory.

   - **New user** Select to create a new user account in Active Directory and mail-enable the user. If you select this option, you'll have to provide the required user account information.

6. If you selected **New User** in Step 5, complete the following boxes on the **New mail user** page. Otherwise skip to Step 7.
   - **First name** Use this box to type the first name of the mail user.
   - **Initials** Use this box to type the initials of the mail user.
   - **Last name** Use this box to type the last name of the mail user.
   - *** Display name** Use this box to type a display name for the user. This is the name that's listed in the contacts list in the EAC and in your organization's address book. By default, this box is populated with the names you enter in the **First name**, **Initials**, and **Last name** boxes. If you didn't use those boxes, you must still type a name in this box because it's required. The name can't exceed 64 characters.
   - *** Name** Use this box to type a name for the mail user. This is the name that's listed in the directory service. This box is also populated with the names you enter in the **First name**, **Initials**, and **Last name** boxes. If you didn't use those boxes, you must still type a name because this box is required. This name also can't exceed 64 characters.

Note:

The **Name** box is only available in Exchange Server 2013. It isn't available in Exchange Online.

   - **Organizational unit** You can select an organizational unit (OU) other than the default (which is the recipient scope). If the recipient scope is set to the forest, the default value is set to the Users container in the domain that contains the computer on which the EAC is running. If the recipient scope is set to a specific domain, the Users container in that domain is selected by default. If the recipient scope is set to a specific OU, that OU is selected by default.

To select a different OU, click **Browse**. The dialog box displays all OUs in the forest that are within the specified scope. Select the OU you want, and then click **OK**.

Note:

The **Organizational unit** box is only available in Exchange Server 2013. It isn't available in Exchange Online.

- *** User logon name** Use this box to type the name that the mail user will use to log on to the domain. The user logon name consists of a user name on the left side of the at (@) symbol and a suffix on the right side. Typically, the suffix is the domain name the user account resides in.

Note:
In Exchange Online, this box is labeled as **User ID**.

- *** New Password** Use this box to type the password that the mail user must use to log on to the domain.

Note:
Make sure that the password you supply complies with the password length, complexity, and history requirements of the domain you're creating the user account in.

- *** Confirm password** Use this box to confirm the password that you typed in the **Password** box.
- **Require password change on next logon** Select this check box if you want mail users to reset the password when they first log on to the domain.

If you select this check box, at first logon, the new mail user will be prompted with a dialog box in which to change the password. The mail user won't be allowed to perform any tasks until the password is changed successfully.

7. When you've finished, click **Save** to create the mail user.

Use the Shell to create a mail user

This example creates a mail-enabled user account for Jeffrey Zeng in Exchange Server 2013 with the following details:

- The name and display name is Jeffrey Zeng.
- The alias is jeffreyz.
- The external email address is jzeng@tailspintoys.com.
- The first name is Jeffrey and the last name is Zeng.
- The logon name is jeffreyz@contoso.com.
- The password is Pa$$word1.
- The mail user will be created in the default OU. To specify a different OU, you can use the *OrganizationalUnit* parameter.

```
New-MailUser -Name "Jeffrey Zeng" -Alias jeffreyz -
ExternalEmailAddress jzeng@tailspintoys.com -FirstName
Jeffrey -LastName Zeng -UserPrincipalName
jeffreyz@contoso.com -Password (ConvertTo-SecureString -
String 'Pa$$word1' -AsPlainText -Force)
```

This example creates a mail-enabled user account for Rene Valdes in Exchange Online.

```
New-MailUser -Name "Rene Valdes" -Alias renev -
ExternalEmailAddress renevaldes@fineartschool.edu -
```

```
FirstName Rene -LastName Valdes -MicrosoftOnlineServicesID
renev@contoso.com -Password (ConvertTo-SecureString -String
'P@sswOrd' -AsPlainText -Force)
```

How do you know this worked?

To verify that you've successfully created a mail user, do one of the following:

- In the EAC, navigate to **Recipients** > **Contacts**. The new mail user is displayed in the list of contacts. Under **Contact Type**, the type is **Mail user**.
- In the Shell, run the following command to display information about the new mail user.

```
Get-MailUser <Name> | FL
Name,RecipientTypeDetails,ExternalEmailAddress
```

Change mail user properties

After you create a mail user, you can make changes and set additional properties by using the EAC or the Shell.

You can also change properties for multiple user mailboxes at the same time. For more information, see Bulk edit mail users.

The estimated time to complete this task will vary based on the number of properties you want to view or change.

Use the EAC to change user mailbox properties

1. In the EAC, navigate to **Recipients** > **Contacts**.
2. In the list of contacts, click the mail user that you want to change the properties for, and then click **Edit** ✎.
3. On the mail user properties page, click one of the following sections to view or change properties.
   - General
   - Contact Information
   - Organization
   - Email Addresses
   - Mail Flow Settings
   - Member Of
   - MailTip

General

Use the **General** section to view or change basic information about the mail user.

- **First name**, **Initials**, **Last name**
- *** Name** This is the name that's listed in Active Directory. If you change this name, it can't exceed 64 characters.

- *** Display name** This name appears in your organization's address book, on the To: and From: lines in email, and in the list of contacts in the EAC. This name can't contain empty spaces before or after the display name.
- *** User logon name** This is the name that the user uses to log on to the domain. In Exchange Online, this is the User ID that the user uses to sign in to Office 365.
- **Hide from address lists** Select this check box to prevent the mail user from appearing in the address book and other address lists that are defined in your Exchange organization. After you select this check box, users can still send messages to the recipient by using the email address.
- **Require password change on next logon** Select this check box if you want the user to reset their password the next time they log on to the domain.

Note:
This box isn't available in Exchange Online.

Click **More options** to view or change these additional properties:

- **Organizational unit** This read-only box displays the organizational unit (OU) that contains the mail user account. You have to use Active Directory Users and Computers to move the account to a different OU.

Note:
This box isn't available in Exchange Online.

- **Custom attributes** This section displays the custom attributes defined for the mail user. To specify custom attribute values, click **Edit** . You can specify up to 15 custom attributes for the recipient.

Contact Information

Use the **Contact Information** section to view or change the user's contact information. The information on this page is displayed in the address book. Click **More options** to display additional boxes.

Tip:
You can use the **State/Province** box to create recipient conditions for dynamic distribution groups, email address policies, or address lists.

Organization

Use the **Organization** section to record detailed information about the user's role in the organization. This information is displayed in the address book. Also, you can create a virtual organization chart that's accessible from email clients such as Outlook.

- **Title** Use this box to view or change the recipient's title.
- **Department** Use this box to view or change the department in which the user works. You can use this box to create recipient conditions for dynamic distribution groups, email address policies, or address lists.
- **Company** Use this box to view or change the company for which the user works. You can use this box to create recipient conditions for dynamic distribution groups, email address policies, or

address lists.

- **Manager** To add a manager, click **Browse**. In **Select Manager**, select a person, and then click **OK**.
- **Direct reports** You can't modify this box. A *direct report* is a user who reports to a specific manager. If you've specified a manager for the user, that user appears as a direct report in the details of the manager's mailbox. For example, Kari manages Chris and Kate, so Kari is specified in the **Manager** box for Chris and Kate, and Chris and Kate appear in the **Direct reports** box in the properties of Kari's account.

Email Addresses

Use the **Email Addresses** section to view or change the email addresses associated with the mail user. This includes the mail user's primary SMTP address, their external email address, and any associated proxy addresses. The primary SMTP address (also known as the *default reply address*) is displayed in bold text in the address list, with the uppercase **SMTP** value in the **Type** column. By default, after the mail user is created, the primary SMTP address and the external email address are the same.

- **Add** Click **Add +** to add a new email address for this mailbox. Select one of following address types:
  - **SMTP** This is the default address type. Click this button and then type the new SMTP address in the *** Email address** box.
  - **Custom address type** Click this button and type one of the supported non-SMTP email address types in the *** Email address** box.

Note:

With the exception of X.400 addresses, Exchange doesn't validate custom addresses for correct formatting. You must make sure that the custom address you specify complies with the format requirements for that address type.

- **Set the external email address** Use this box to change the mail user's external address. Email sent to this mail user is forwarded to this email address.
- **Automatically update email addresses based on the email address policy applied to this recipient** Select this check box to have the recipient's email addresses automatically updated based on changes made to email address policies in your organization. This box is selected by default.

Note:

This check box isn't available in Exchange Online.

Mail Flow Settings

Use the **Mail Flow Settings** section to view or change the following settings:

- **Message Size Restrictions** These settings control the size of messages that the mail user can send and receive. Click **View details** to view and change maximum size for sent and received messages.
  - **Sent messages** To specify a maximum size for messages sent by this user, select the

Maximum message size (KB) check box and type a value in the box. The message size must be between 0 and 2,097,151 KB. If the user sends a message larger than the specified size, the message will be returned to the user with a descriptive error message.

- **Received messages** To specify a maximum size for messages received by this user, select the **Maximum message size (KB)** check box and type a value in the box. The message size must be between 0 and 2,097,151 KB. If the user receives a message larger than the specified size, the message will be returned to the sender with a descriptive error message.

- **Message Delivery Restrictions** These settings control who can send email messages to this mail user. Click **View details** to view and change these restrictions.
  - **Accept messages from** Use this section to specify who can send messages to this user.
    - **All senders** Select this option to specify that the user can accept messages from all senders. This includes both senders in your Exchange organization and external senders. This option is selected by default. This option includes external users only if you clear the **Require that all senders are authenticated** check box. If you select this check box, messages from external users will be rejected.
    - **Only senders in the following list** Select this option to specify that the user can accept messages only from a specified set of senders in your Exchange organization. Click **Add +** to display the **Select Recipients** page, which displays a list of all recipients in your Exchange organization. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search**.
    - **Require that all senders are authenticated** Select this option to prevent anonymous users from sending messages to the user.
  - **Reject messages from** Use this section to block people from sending messages to this user.
    - **No senders** Select this option to specify that the mailbox won't reject messages from any senders in the Exchange organization. This option is selected by default.
    - **Senders in the following list** Select this option to specify that the mailbox will reject messages from a specified set of senders in your Exchange organization. Click **Add +** to display the **Select Recipients** page, which displays a list of all recipients in your Exchange organization. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search**.

Member Of

Use the **Member Of** section to view a list of the distribution groups or security groups to which this user belongs. You can't change membership information on this page. Note that the user may match the criteria for one or more dynamic distribution groups in your organization. However, dynamic distribution groups aren't displayed on this page because their membership is calculated each time they're used.

MailTip

Use the **MailTip** section to add a MailTip to alert users of potential issues before they send a

message to this recipient. A MailTip is text that's displayed in the InfoBar when this recipient is added to the To, Cc, or Bcc lines of a new email message.

Note:
MailTips can include HTML tags, but scripts aren't allowed. The length of a custom MailTip can't exceed 175 displayed characters. HTML tags aren't counted in the limit.

Use the Shell to change mail user properties

Properties for a mail user are stored in both Active Directory and Exchange. In general, use the **Get-User** and **Set-User** cmdlets to view and change organization and contact information properties. Use the **Get-MailUser** and **Set-MailUser** cmdlets to view or change mail-related properties, such email addresses, the MailTip, custom attributes, and whether the mail user is hidden from address lists.

Use the **Get-MailUser** and **Set-MailUser** cmdlets to view and change properties for mail users. For information, see the following topics:

- Get-User
- Set-User
- Get-MailUser
- Set-MailUser

Here are some examples of using the Shell to change mail user properties.

This example sets the external email address for Pilar Pinilla.

```
Set-MailUser "Pilar Pinilla" -ExternalEmailAddress
pilarp@tailspintoys.com
```

This example hides all mail users from the organization's address book.

```
Get-MailUser | Set-MailUser -HiddenFromAddressListsEnabled
$true
```

This example sets the Company property for all mail users to Contoso.

```
Get-User -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'mailuser')} | Set-User -Company
Contoso
```

This example sets the CustomAttribute1 property to a value of ContosoEmployee for all mail users that have a value of Contoso in the Company property.

```
Get-User -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'mailuser') -and (Company -eq
'Contoso')}| Set-MailUser -CustomAttribute1 ContosoEmployee
```

How do you know this worked?

To verify that you've successfully changed properties for mail users, do the following:

- In the EAC, select the mail user and then click **Edit** to view the property that you changed.
- In the Shell, use the **Get-User** and **Get-MailUser** cmdlets to verify the changes. One advantage of using the Shell is that you can view multiple properties for multiple mail contacts.

```
Get-MailUser | Fl Name,CustomAttribute1
```

In the example above where the Company property was set to Contoso for all mail contacts, run the following command to verify the changes:

```
Get-User -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'mailuser')} | FL Name,Company
```

In the example above where all mail users had the CustomAttribute1 property set to ContosoEmployee, run the following command to verify the changes.

```
Get-MailUser | Fl Name,CustomAttribute1
```

Bulk edit mail users

You can also use the EAC to change selected properties for multiple mail users. When you select two or more mail users from the contacts list in the EAC, the properties that can be bulk edited are displayed in the Details pane. When you change one of these properties, the change is applied to all selected recipients.

When you bulk edit mail users, you can change the following property areas:

- **Contact Information** Change shared properties such as street, postal code, and city name.
- **Organization** Change shared properties such as department name, company name, and the manager that the selected mail contacts or mail users report to.

Use the EAC to bulk edit mail users

1. In the EAC, navigate to **Recipients** > **Contacts**.
2. In the list of contacts, select two or more mail users. You can't bulk edit a combination of mail contacts and mail users.

Tip:
You can select multiple adjacent mail users by holding down the Shift key and clicking the first mail user, and then clicking the last mail user you want to edit. You can also select multiple mail users by holding down the Ctrl key and clicking each one that you want to edit.

3. In the Details pane, under **Bulk Edit**, click **Update** under **Contact Information** or **Organization**.
4. Make the changes on the properties page and then save your changes.

How do you know this worked?

To verify that you've successfully bulk edited mail users, do one of the following:

- In the EAC, select each of the mail users that you bulk edited and then click **Edit** to view the properties that you changed.
- In the Shell, use the **Get-User** cmdlet to verify the changes. For example, say you used the bulk edit feature in the EAC to change the manager and the office for all mail users from a vendor company named A. Datum Corporation. To verify these changes, you could run the following command in the Shell:

```
Get-User -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'mailuser') -and (Company -eq
'Adatum')} | fl Name,Office,Manager
```

Use directory synchronization to manage mail users in Exchange Online

This section provides information about managing email users by using directory synchronization in Exchange Online. Directory synchronization is available for hybrid customers with on-premises and cloud-hosted mailboxes, and for fully hosted Exchange Online customers whose Active Directory is on-premises.

Note:

If you use directory synchronization to manage your recipients, you can still add and manage users in the Office 365 admin center, but they will not be synchronized with your on-premises Active Directory. This is because directory synchronization only syncs recipients from your on-premises Active Directory to the cloud.

Note:

Using directory synchronization is recommended for use with the following features:

- **Outlook safe sender and blocked sender lists** When synchronized to the service, these lists will take precedence over spam filtering in the service. This lets users manage their own safe sender and blocked sender lists on a per-user or per-domain basis.
- **Directory Based Edge Blocking (DBEB)** For more information about DBEB, see **Use Directory Based Edge Blocking to Reject Messages Sent to Invalid Recipients**.
- **End user spam quarantine** In order to access the end user spam quarantine, end users must have a valid Office 365 user ID and password. Customers with on-premises mailboxes must be valid email users.
- **Transport rules** When you use directory synchronization, your existing Active Directory users and groups are automatically uploaded to the cloud, and you can then create Transport rules that target specific users and/or groups without having to manually add them via the EAC or remote Windows PowerShell. Note that dynamic distribution groups can't be synchronized via directory synchronization.

Before you begin

Get the necessary permissions and prepare for directory synchronization, as described in Prepare for directory synchronization.

To synchronize user directories

1. Activate directory synchronization, as described in Activate directory synchronization.
2. Set up your directory synchronization computer, as described in Set up your directory sync computer.
3. Synchronize your directories, as described in Use the Configuration Wizard to sync your directories.

Important:

When you finish the Windows Azure AD Sync Tool Configuration Wizard, the **MSOL_AD_SYNC** account is created in your Active Directory forest. This account is used to read and synchronize your on-premises Active Directory information. In order for directory synchronization to work correctly, make sure that TCP 443 on your local directory synchronization server is open.

4. Activate synced users, as described in Activate synced users.
5. Manage directory synchronization, as described in Manage directory synchronization.
6. Verify that Exchange Online is synchronizing correctly. In the EAC, go to **Recipients** > **Contacts** and view that the list of users was correctly synchronized from your on-premises environment.

Enable or disable email for a mail user

Exchange Server 2013 > Recipients > Manage mail users >

Applies to: *Exchange Online*

Topic Last Modified: *2012-11-14*

You can disable email for an existing mail user in your Exchange organization. When you disable email for a mail user, it's removed from Exchange and your organization's address book. If the mail user is a member of a distribution group, the user no longer receives mail sent to the group. Also, the Exchange attributes are removed from the user object in Active Directory, but the user object and its non-Exchange attributes (such as contact and organization information) are retained in Active Directory.

After you disable email for a mail user, you can mail-enable the user again by using the **Enable-MailUser** cmdlet in the Shell. You can also use this cmdlet to mail-enable any Active Directory user.

Note:

Mail users (also called *mail-enabled users*) are different than users in your organization that have a mailbox. The primary difference is that mail users represent users outside your Exchange organization that have an external email address. They don't have a mailbox in your organization. For more information about the differences between users who have mailboxes

in your organization and mail users, see Recipients.

For additional management tasks related to mail users, see Manage mail users.

What do you need to know before you begin?

- Estimated time to complete each procedure: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mail users" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Disable email for a mail user

As previously stated, when you disable email for a mail user, the Exchange attributes are removed from the corresponding Active Directory mail user object, but the user is retained. The mail user is removed from the list of mail users in the EAC, but you can view and manage the corresponding Active Directory contact object by using Active Directory Users and Computers or by using the **Get-MailUser** and **Set-Set-MailUser** cmdlets in the Shell.

Use the EAC to disable email for a mail user

1. In the EAC, navigate to **Recipients** > **Contacts**.
2. In the list of contacts, click the mail user you want to disable email for.
3. Click **More** ··· and then click **Disable**.
4. A warning will appear asking if you're sure you want to disable the selected mail user. Click **Yes** to disable it.

The mail user will be removed from the contacts list.

Use the Shell to disable email for a mail user

This example disables email for the mail user Yan Li.

```
Disable-MailUser -Identity "Yan Li"
```

For detailed syntax and parameter information, see Disable-MailUser.

How do you know this worked?

To verify that you've successfully disabled email for a mail user, do one of the following:

1. In the EAC, navigate to **Recipients** > **Contacts** and verify that the mail user is no longer listed.
2. In Active Directory Users and Computers, right-click the user, and then click **Properties**. On the **General** tab, notice that the **E-mail** box is blank. This verifies that the mail user isn't mail-enabled.
3. In the Shell, run the following command.

```
Get-MailUser
```

The mail user that you disabled email for won't be returned in the results because this cmdlet only returns mail-enabled users.

4. In the Shell, run the following command.

```
Get-User
```

The mail user that you disabled email for is returned in the results because this cmdlet returns all Active Directory user objects.

Use the Shell to mail-enable users

You can use the **Enable-MailUser** cmdlet to mail-enable existing Active Directory users. You can mail-enable a single user or use a CSV file to mail-enable multiple users.

Use the Shell to mail-enable a single user

This example mail-enables the user Sanjay Shah. You must provide an external email address.

```
Enable-MailUser -Identity "Sanjay Shah" -
ExternalEmailAddress renev@tailspintoys.com
```

Use the Shell and a CSV file to mail-enable multiple users

When you're mail-enabling users in bulk, you first export the list of users that aren't mail-enabled to a CSV (comma-separated values) file, and then add the external email addresses to the CSV file by using a text editor such as Notepad, or a spreadsheet application such as Microsoft Excel. Then you use the updated CSV file in the Shell command to mail-enable the users listed in the CSV file.

1. Run the following command to export a list of existing users that aren't mail-enabled or don't have a mailbox in your organization to a file on the administrator's desktop named UsersToMailEnable.csv.

```
Get-User | Where { $_.RecipientType -eq "User" } | Out-File
"C:\Users\Administrator\Desktop\UsersToMailEnable.csv"
```

The resulting file will be similar to the following file.

```
Name              RecipientType
----              -------------
```

```
Guest           User
krbtgt          User
RMS_SERVICE     User
David Pelton    User
Kim Akers       User
Janet Schorr    User
Jeffrey Zang    User
Spencer Low     User
Toni Poe        User
...
```

2. Make the following changes to the CSV file:
   - Delete any users that you don't want to mail-enable from the CSV file. For example, you would delete the first three entries in the previous example because they're default system accounts.
   - Delete the **RecipientType** column and all the instances of `User`.
   - Add a column heading named **EmailAddress** and then add an email address for each user in the file. The name and external email address for each user must be separated by a comma.

The updated CSV file should look similar to the following file.

```
Name,EmailAddress
David Pelton,davidp@contoso.com
Kim Akers,kakers@tailspintoys.com
Janet Schorr,janet.schorr@adatum.com
Jeffrey Zang,jzang@tailspintoys.com
Spencer Low,spencerl@fouthcoffee.com
Toni Poe,tonip@contoso.com
...
```

3. Run the following command to use the data in the CSV file to mail-enable the users listed in the file.

```
Import-CSV "C:\Users\Administrator\Desktop
\UsersToMailEnable.csv" | ForEach-Object {Enable-MailUser -
Identity $_.Name -ExternalEmailAddress $_.EmailAddress}
```

The command results display information about the new mail-enabled users.

How do you know this worked?

To verify that you've successfully mail-enabled Active Directory users, do one of the following:

- In the EAC, navigate to **Recipients** > **Contacts**. New mail users are displayed in the contact list. Under **Contact Type**, the type is **Mail user**.

Note:

You may have to click **Refresh** to display new mail users.

- In the Shell, run the following command to display information about new mail users.

```
Get-MailUser | Format-Table
Name,RecipientTypeDetails,ExternalEmailAddress
```

Create and Manage Room Mailboxes

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-29*

Estimated time to complete: 5 minutes.

A *room mailbox* is a resource mailbox that's assigned to a physical location, such as a conference room, an auditorium, or a training room. After an administrator creates room mailboxes, users can easily reserve rooms by including room mailboxes in meeting requests. You can use the EAC or the Shell to create a room mailbox. For more details, check out Recipients.

For information about another type of resource mailbox, an equipment mailbox, see Manage equipment mailboxes.

What do you need to know before you begin?

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Important:

If you're running Exchange 2013 in a hybrid scenario, make sure you create the room mailboxes in the appropriate place. Create your room mailboxes for your on-premises organization on-premises, and room mailboxes for Exchange Online side should be created in the cloud.

What do you want to do?

Create a room mailbox

Use the EAC to create a room mailbox

1. In the EAC, navigate to **Recipients** > **Resources**.
2. To create a room mailbox, click **New +** > **Room mailbox**. To create an equipment mailbox, click **New +** > **Equipment mailbox**.
3. Use the options on the page to specify the settings for the new resource mailbox.
   - *** Room name** Use this box to type a name for the room mailbox. This is the name that's listed in the resource mailbox list in the EAC and in your organization's address book. This name is required and it can't exceed 64 characters.

Tip:
Although there are other fields that describe the details of the room, for example, Location and Capacity, consider summarizing the most important details in the room name using a consistent naming convention. Why? So users can easily see the details when they select the room from the address book in the meeting request.

   - *** Email address** A room mailbox has an email address so it can receive booking requests. The email address consists of an alias on the left side of the @ symbol, which must be unique in the forest, and your domain name on the right. The email address is required.
   - **Organizational unit** You can select an organizational unit (OU) other than the default (which is the recipient scope). If the recipient scope is set to the forest, the default value is set to the Users container in the Active Directory domain that contains the computer on which the EAC is running. If the recipient scope is set to a specific domain, the Users container in that domain is selected by default. If the recipient scope is set to a specific OU, that OU is selected by default.

To select a different OU, click **Browse**. The dialog box displays all OUs in the forest that are within the specified scope. Select the desired OU, and then click **OK**.

   - **Location**, **Phone**, **Capacity** You can use these fields to enter details about the room. However, as explained earlier, you can include some or all of this information in the room name so users can see it.
   - **Booking requests** Use this section to configure how the room mailbox handles reservation requests.
     - **Automatically accept or decline booking requests** A valid meeting request automatically reserves the resource. If there's a scheduling conflict with an existing reservation, or if the booking request violates the scheduling limits of the resource, for example, the reservation duration is too long, the meeting request is automatically declined.
     - **Select delegates to accept or decline booking requests** The delegates are responsible for accepting or declining meeting requests that are sent to the room mailbox. If you assign more than one resource delegate, only one of them has to act on a specific meeting request.
   - **Delegates** If you selected the option requiring that booking requests are sent to delegates, use this section to select delegates. To add a delegate, click **Add**. On the **Select Delegates** page, select a user, click **Add**, and then click **OK** to return to the **New room mailbox** page.

To remove a delegate, select the user and then click **Remove**.

4. Click **More options** to configure the following fields. Otherwise, skip to Step 5 to save the new room mailbox.
   - **Alias** Use this field to create an alias that is different than the one that you used to create the email address for the room mailbox. For example, if you create a room mailbox with an email

address of ConfRoom1@contoso.com, the portion of the email address on the left side of the **@** symbol is used as the alias by default. If you want the alias to be different, type it in this field.

- **Specify the mailbox database** Use this option to specify a mailbox database instead of allowing Exchange to select a database for you. Click **Browse** to open the **Select Mailbox Database** dialog box. This dialog box lists all the mailbox databases in your Exchange organization. By default, the mailbox databases are sorted by name. You can also click the title of the corresponding column to sort the databases by server name or version. Select the mailbox database you want to use, and then click **OK**.
- **Address book policy** Use this option to specify an address book policy (ABP) for the room mailbox. ABPs contain a global address list (GAL), an offline address book (OAB), a room list, and a set of address lists. To learn more, see Address book policies.

In the drop-down list, select the policy that you want associated with this mailbox.

5. When you're finished, click **Save** to create the room mailbox.

Use the Shell to create a room mailbox

This example creates a room mailbox with the following configuration:

- The room mailbox resides on Mailbox Database 1.
- The mailbox's name is ConfRoom1. This name will also be used to create the room's email address.
- The display name in the EAC and the address book will be Conference Room 1.
- The mailbox is in the Conference Rooms organizational unit.
- The *Room* switch specifies that this mailbox will be created as a room mailbox.

```
New-Mailbox -Database "Mailbox Database 1" -Name ConfRoom1
-OrganizationalUnit "Conference Rooms" -DisplayName
"Conference Room 1" -Room
```

For detailed syntax and parameter information, see New-Mailbox.

How do you know this worked?

To verify that you've successfully created a room mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Resources**. The new room mailbox is displayed in the mailbox list. Under **Mailbox Type**, the type is **Room**.
- In the Shell, run the following command to display information about the new room mailbox.

```
Get-Mailbox <Name> | FL
Name,RecipientTypeDetails,PrimarySmtpAddress
```

Create a room list

If you're planning to have more than a hundred rooms, or already have more than a hundred rooms

created, use a room list to help you organize your rooms. If your company has several buildings with rooms that can be booked for meetings, it might help to create room lists for each building. Room lists are specially marked distribution groups that you can use the same way you use distribution groups. However, you can only create room lists using the Exchange Management Shell.

Use the Shell to create a room list

This example creates a room list for building 32.

```
New-DistributionGroup -Name "Building 32 Conference Rooms"
-OrganizationalUnit "contoso.com/rooms" -RoomList
```

Use the Shell to add a room to a room list

This example adds confroom3223 to the building 32 room list.

```
Add-DistributionGroupMember -Identity "Building 32
Conference Rooms" -Member confroom3223@contoso.com
```

Use the Shell to convert a distribution group to a room list

You may already have created distribution groups in the past that contain your conference rooms. You don't need to recreate them; we can convert them quickly into a room list.

This example converts the distribution group, building 34 conference rooms, to a room list.

```
Set-DistributionGroup -Identity "Building 34 Conference
Rooms" -RoomList
```

Change room mailbox properties

After you create a room mailbox, you can make changes and set additional properties by using the EAC or the Shell.

Use the EAC to change room mailbox properties

1. In the EAC, navigate to **Recipients** > **Resources**.
2. In the list of resource mailboxes, click the room mailbox that you want to change the properties for, and then click **Edit** ✎.
3. On the room mailbox properties page, click one of the following sections to view or change properties.
   - General
   - Delegates
   - Booking Options
   - Contact Information
   - Email Address

- MailTip

General

Use the **General** section to view or change basic information about the resource.

- *** Room name** This name appears in the resource mailbox list in the EAC and in your organization's address book. It can't exceed 64 characters if you change it.
- *** Email address** This read-only box displays the email address for the room mailbox. You can change it in the Email Address section.
- **Capacity** Use this box to enter the maximum number of people who can safely occupy the room.

Click **More options** to view or change these additional properties:

- **Organizational unit** This read-only box displays the organizational unit (OU) that contains the account for the room mailbox. You have to use Active Directory Users and Computers to move the account to a different OU.
- **Mailbox database** This read-only box displays the name of the mailbox database that hosts the room mailbox. Use the **Migration** page in the EAC to move the mailbox to a different database.
- *** Alias** Use this box to change the alias for the room mailbox.
- **Hide from address lists** Select this check box to prevent the room mailbox from appearing in the address book and other address lists that are defined in your Exchange organization. After you select this check box, users can still send booking messages to the room mailbox by using the email address.
- **Department** Use this box to specify a department name that the room is associated with. You can use this property to create recipient conditions for dynamic distribution groups and address lists.
- **Company** Use this box to specify a company that the room is associated with, if applicable. Like the Department property, you can use this property to create recipient conditions for dynamic distribution groups and address lists.
- **Address book policy** Use this option to specify an address book policy (ABP) for the room mailbox. ABPs contain a global address list (GAL), an offline address book (OAB), a room list, and a set of address lists. To learn more, see Address book policies.

In the drop-down list, select the policy that you want associated with this mailbox.

- **Custom attributes** This section displays the custom attributes defined for the room mailbox. To specify custom attribute values, click **Edit** ✎. You can specify up to 15 custom attributes for the recipient.

Delegates

Use this section to view or change how the room mailbox handles reservation requests and to define who can accept or decline booking requests if it isn't done automatically.

- **Booking requests** Select one of the following options to handle booking requests.
  - **Accept or decline booking requests automatically** A valid meeting request automatically reserves the room. If there's a scheduling conflict with an existing reservation, or if the booking request violates the scheduling limits of the resource, for example, the reservation duration is

too long, the meeting request is automatically declined.
    - **Select delegates who can accept or decline booking requests** Resource delegates are responsible for accepting or declining meeting requests that are sent to the room mailbox. If you assign more than one resource delegate, only one of them has to act on a specific meeting request.
- **Delegates** If you selected the option requiring that booking requests be sent to delegates, the specified delegates are listed. Click **Add +** or **Remove −** to add or remove delegates from this list.

Booking Options

Use the **Booking Options** section to view or change the settings for the booking policy that defines when the room can be scheduled, how long it can be reserved, and how far in advance it can be reserved.

- **Allow repeating meetings** This setting allows or prevents repeating meetings for the room. By default, this setting is enabled, so repeating meetings are allowed.
- **Allow scheduling only during working hours** This setting accepts or declines meeting requests that aren't during the working hours defined for the room. By default, this setting is disabled, so meeting requests are allowed outside the working hours.By default, working hours are 8:00 A.M. to 5:00 P.M. Monday through Friday. You can configure the working hours of the room mailbox in the Appearance section on the Calendar page.
- **Always decline if the end date is beyond this limit** This setting controls the behavior of repeating meetings that extend beyond the date specified by the maximum booking lead time setting.
    - If you enable this setting, a repeating booking request is automatically declined if the bookings start on or before the date specified by the value in the **Maximum booking lead time** box, and they extend beyond the specified date. This is the default setting.
    - If you disable this setting, a repeating booking request is automatically accepted if booking requests start on or before the date specified by the value in the **Maximum booking lead time** box, and they extend beyond the specified date. However, the number of bookings is reduced so bookings won’t occur after the specified date.
- **Maximum booking lead time (days)** This setting specifies the maximum number of days in advance that the room can be booked. Valid input is an integer between 0 and 1080. The default value is 180 days.
- **Maximum duration (hours)** This setting specifies the maximum duration that the room can be reserved in a booking request. The default value is 24 hours.

For repeating booking requests, the maximum booking duration applies to the length of each instance of the repeating booking request.

There’s also a box on this page that you can use to write a message that will be sent to users who send booking requests to reserve the room.

Contact Information

Use the **Contact Information** section to view or change the contact information for the room. The information on this page is displayed in the address book.

Tip:
You can use the **State/Province** box to create recipient conditions for dynamic distribution groups, email address policies, or address lists.

Email Address

Use the **Email Address** section to view or change the email addresses associated with the room mailbox. This includes the mailbox's primary SMTP address and any associated proxy addresses. The primary SMTP address (also known as the *reply address*) is displayed in bold text in the address list, with the uppercase **SMTP** value in the **Type** column.

- **Add** Click **Add +** to add a new email address for this mailbox. Select one of following address types:
  - **SMTP** This is the default address type. Click this button and then type the new SMTP address in the *** Email address** box.
  - **EUM** An EUM (Exchange Unified Messaging) address is used by the Microsoft Exchange Unified Messaging service to locate UM-enabled recipients within an Exchange organization. EUM addresses consist of the extension number and the UM dial plan for the UM-enabled user. Click this button and type the extension number in the **Address/Extension** box. Then click **Browse** and select a dial plan for the mailbox.
  - **Custom address type** Click this button and type one of the supported non-SMTP email address types in the *** Email address** box.

Note:
With the exception of X.400 addresses, Exchange doesn't validate custom addresses for correct formatting. You must make sure that the custom address you specify complies with the format requirements for that address type.

Note:
When you add a new email address, you have the option to make it the primary SMTP address.

- **Automatically update email addresses based on the email address policy applied to this recipient** Select this check box to have the recipient's email addresses automatically updated based on changes made to email address policies in your organization.

MailTip

Use the **MailTip** section to add a MailTip to alert users of potential issues before they send a booking request to the room mailbox. A MailTip is text that's displayed in the InfoBar when this recipient is added to the To, Cc, or Bcc lines of a new email message.

Note:
MailTips can include HTML tags, but scripts aren't allowed. The length of a custom MailTip can't exceed 175 displayed characters. HTML tags aren't counted in the limit.

Use the Shell to change room mailbox properties

Use the following sets of cmdlets to view and change room mailbox properties:**Get-Mailbox** and **Set-Mailbox** cmdlets to view and change general properties and email addresses for room mailboxes. Use the **Get-CalendarProcessing** and **Set-CalendarProcessing** cmdlets to view and change delegates and booking options.

- **Get-User** and **Set-User** Use these cmdlets to view and set general properties such as location, department, and company names.
- **Get-Mailbox** and **Set-Mailbox** Use these cmdlets to view and set mailbox properties, such as email addresses and the mailbox database.
- **Get-CalendarProcessing** and **Set-CalendarProcessing** Use these cmdlets to view and set booking options and delegates.

For information about these cmdlets, see the following topics:

- Get-User
- Set-User
- Get-Mailbox
- Set-Mailbox
- Get-CalendarProcessing
- Set-CalendarProcessing

Here are some examples of using the Shell to change room mailbox properties.

This example changes the display name, the primary SMTP address (called the default reply address), and the room capacity. Also, the previous reply address is kept as a proxy address.

```
Set-Mailbox "Conf Room 123" -DisplayName "Conf Room 31/123
(12)" -EmailAddresses
SMTP:Rm33.123@contoso.com,smtp:rm123@contoso.com -
ResourceCapacity 12
```

This example configures room mailboxes to allow booking requests to be scheduled only during working hours and sets a maximum duration of 9 hours.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'RoomMailbox')} | Set-
CalendarProcessing -ScheduleOnlyDuringWorkHours $true -
MaximumDurationInMinutes 540
```

This example uses the **Get-User** cmdlet to find all room mailboxes that correspond to private conference rooms, and then uses the **Set-CalendarProcessing** cmdlet to send booking requests to a delegate named Robin Wood to accept or decline.

```
Get-User -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'RoomMailbox') -and (DisplayName
-like 'Private*')} | Set-CalendarProcessing -
```

```
AllBookInPolicy $false -AllRequestInPolicy $true -
ResourceDelegates "Robin Wood"
```

How do you know this worked?

To verify that you've successfully changed properties for a room mailbox, do the following:

- In the EAC, select the mailbox and then click **Edit** to view the property or feature that you changed. Depending on the property that you changed, it might be displayed in the Details pane for the selected mailbox.
- In the Shell, use the **Get-Mailbox** cmdlet to verify the changes. One advantage of using the Shell is that you can view multiple properties for multiple mailboxes. In the example above where booking requests could be scheduled only during working hours and have a maximum duration of 9 hours, run the following command to verify the new values.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'RoomMailbox')} | Get-
CalendarProcessing | fl
Identity,ScheduleOnlyDuringWorkHours,MaximumDurationInMinut
es
```

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Manage equipment mailboxes

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-10-09*

An *equipment mailbox* is a resource mailbox assigned to a resource that's not location specific, such as a portable computer, projector, microphone, or a company car. After an administrator creates an equipment mailbox, users can easily reserve the piece of equipment by including the corresponding equipment mailbox in a meeting request. You can use the EAC and the Shell to create an equipment mailbox or change equipment mailbox properties. For more information, see Recipients.

For information about another type of resource mailbox, a room mailbox, see Create and Manage Room Mailboxes.

What do you need to know before you begin?

- Estimated time to complete: 2 to 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Create an equipment mailbox

Use the EAC to create an equipment mailbox

1. In the EAC, navigate to **Recipients** > **Resources**.
2. To create an equipment mailbox, click **New** > **Equipment mailbox**. To create a room mailbox, click **New** > **Room mailbox**.
3. Use the options on the page to specify the settings for the new resource mailbox.
   - *** Equipment name** Use this box to type a name for the equipment mailbox. This is the name that's listed in the resource mailbox list in the EAC and in your organization's address book. This name is required and it can't exceed 64 characters.

Tip:

Although there are other fields that describe the details of the room, for example, Capacity, consider summarizing the most important details in the equipment name using a consistent naming convention. Why? So users can easily see the details when they select the equipment from the address book in a meeting request.

   - *** Email address** An equipment mailbox has an email address so it can receive booking requests. The email address consists of an alias on the left side of the @ symbol, which must be unique in the forest, and your domain name on the right. The email address is required.
   - **Organizational unit** You can select an organizational unit (OU) other than the default (which is the recipient scope). If the recipient scope is set to the forest, the default value is set to the Users container in the Active Directory domain that contains the computer on which the EAC is running. If the recipient scope is set to a specific domain, the Users container in that domain is selected by default. If the recipient scope is set to a specific OU, that OU is selected by default.

To select a different OU, click **Browse**. The dialog box displays all OUs in the forest that are within the specified scope. Select the desired OU, and then click **OK**.

   - **Booking requests** Use this section to configure how the equipment mailbox handles

reservation requests.

- **Accept or decline booking requests automatically** A valid meeting request automatically reserves the resource. If there's a scheduling conflict with an existing reservation, or if the booking request violates the scheduling limits of the resource, for example, the reservation duration is too long, the meeting request is automatically declined.
- **Select delegates who can accept or decline booking requests** The delegates are responsible for accepting or declining meeting requests that are sent to the equipment mailbox. If you assign more than one resource delegate, only one of them has to act on a specific meeting request.

- **Delegates** If you selected the option requiring that booking requests are sent to delegates, use this section to select delegates. To add a delegate, click **Add +**. On the **Select Delegates** page, select a user, click **Add**, and then click **OK** to return to the **New equipment mailbox** page.

4. Click **More options** to configure the following fields. Otherwise, skip to Step 5 to save the new equipment mailbox.
   - **Alias** Use this field to create an alias that is different than the one that you used to create the email address for the equipment mailbox. For example, if you create an equipment mailbox with an email address of motorpool1@contoso.com, the portion of the email address on the left side of the **@** symbol is used as the alias by default. If you want the alias to be different, type it in this field.
   - **Specify the mailbox database** Use this option to specify a mailbox database instead of allowing Exchange to select a database for you. Click **Browse** to open the **Select Mailbox Database** dialog box. This dialog box lists all the mailbox databases in your Exchange organization. By default, the mailbox databases are sorted by name. You can also click the title of the corresponding column to sort the databases by server name or version. Select the mailbox database you want to use, and then click **OK**.
   - **Address book policy** Use this option to specify an address book policy (ABP) for the equipment mailbox. ABPs contain a global address list (GAL), an offline address book (OAB), a room list, and a set of address lists. To learn more, see Address book policies.

In the drop-down list, select the policy that you want associated with this mailbox.

5. When you're finished, click **Save** to create the equipment mailbox.

Use the Shell to create an equipment mailbox

This example creates an equipment mailbox with the following configuration:

- The equipment mailbox resides on Mailbox Database 1.
- The equipment's name is MotorVehicle2 and the name will display in the GAL as Motor Vehicle 2.
- The email address is MotorVehicle2@contoso.com.
- The mailbox is in the Equipment organizational unit.
- The *Equipment* parameter specifies that this mailbox will be created as an equipment mailbox.

```
New-Mailbox -Database "Mailbox Database 1" -Name
MotorVehicle2 -OrganizationalUnit Equipment -DisplayName
```

```
"Motor Vehicle 2" -Equipment
```

For detailed syntax and parameter information, see New-Mailbox.

How do you know this worked?

To verify that you've successfully created a user mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Resources**. The new user mailbox is displayed in the mailbox list. Under **Mailbox Type**, the type is **Equipment**.
- In the Shell, run the following command to display information about the new equipment mailbox.

```
Get-Mailbox <Name> | FL
Name,RecipientTypeDetails,PrimarySmtpAddress
```

Change equipment mailbox properties

After you create an equipment mailbox, you can make changes and set additional properties by using the EAC or the Shell.

Use the EAC to change equipment mailbox properties

1. In the EAC, navigate to **Recipients** > **Resources**.
2. In the list of resource mailboxes, click the equipment mailbox that you want to change the properties for, and then click **Edit** ✎.
3. On the equipment mailbox properties page, click one of the following sections to view or change properties.
   - General
   - Delegates
   - Booking Options
   - Contact Information
   - Email Address
   - MailTip

General

Use the **General** section to view or change basic information about the resource.

- *** Equipment name** This name appears in the resource mailbox list in the EAC and in your organization's address book. It can't exceed 64 characters if you change it.
- *** Email address** This read-only box displays the email address for the equipment mailbox. You can change it in the Email Address section.
- **Capacity** Use this box to enter the maximum number of people who can use this resource, if applicable, For example, if the equipment mailbox corresponds to a compact car, you could enter **4**.

Click **More options** to view or change these additional properties:

- **Organizational unit** This read-only box displays the organizational unit (OU) that contains the account for the equipment mailbox. You have to use Active Directory Users and Computers to move the account to a different OU.
- **Mailbox database** This read-only box displays the name of the mailbox database that hosts the equipment mailbox. Use the **Migration** page in the EAC to move the mailbox to a different database.
- *** Alias** Use this box to change the alias for the equipment mailbox.
- **Hide from address lists** Select this check box to prevent equipment mailbox from appearing in the address book and other address lists that are defined in your Exchange organization. After you select this check box, users can still send booking messages to the equipment mailbox by using the email address.
- **Department** Use this box to specify a department name that the resource is associated with. You can use this property to create recipient conditions for dynamic distribution groups and address lists.
- **Company** Use this box to specify a company that the resource is associated with. Like the Department property, you can use this property to create recipient conditions for dynamic distribution groups and address lists.
- **Address book policy** Use this option to specify an address book policy (ABP) for the resource. ABPs contain a global address list (GAL), an offline address book (OAB), a room list, and a set of address lists. To learn more, see Address book policies.

In the drop-down list, select the policy that you want associated with this mailbox.

- **Custom attributes** This section displays the custom attributes defined for the equipment mailbox. To specify custom attribute values, click **Edit** ✎. You can specify up to 15 custom attributes for the recipient.

Delegates

Use this section to view or change how the equipment mailbox handles reservation requests and to define who can accept or decline booking requests if it isn't done automatically.

- **Booking requests** Select one of the following options to handle booking requests.
  - **Accept or decline booking requests automatically** A valid meeting request automatically reserves the resource. If there's a scheduling conflict with an existing reservation, or if the booking request violates the scheduling limits of the resource, for example, the reservation duration is too long, the meeting request is automatically declined.
  - **Select delegates who can accept or decline booking requests** Resource delegates are responsible for accepting or declining meeting requests that are sent to the equipment mailbox. If you assign more than one resource delegate, only one of them has to act on a specific meeting request.
- **Delegates** If you selected the option requiring that booking requests be sent to delegates, the specified delegates are listed. Click **Add +** or **Remove −** to add or remove delegates from this list.

Booking Options

Use the **Booking Options** section to view or change the settings for the booking policy that defines when the resource can be scheduled, how long it can be reserved, and how far in advance it can be reserved.

- **Allow repeating meetings** This setting allows or prevents repeating meetings for the resource. By default, this setting is enabled, so repeating meetings are allowed.
- **Allow scheduling only during working hours** This setting accepts or declines meeting requests that aren't during the working hours defined for the resource. By default, this setting is disabled, so meeting requests are allowed outside the working hours.By default, working hours are 8:00 A.M. to 5:00 P.M. Monday through Friday. You can configure the working hours of the equipment mailbox in the Appearance section on the Calendar page.
- **Always decline if the end date is beyond this limit** This setting controls the behavior of repeating meetings that extend beyond the date specified by the maximum booking lead time setting.
  - If you enable this setting, a repeating booking request is automatically declined if the bookings start on or before the date specified by the value in the **Maximum booking lead time** box, and they extend beyond the specified date. This is the default setting.
  - If you disable this setting, a repeating booking request is automatically accepted if the booking requests start on or before the date specified by the value in the **Maximum booking lead time** box, and they extend beyond the specified date. However, the number of bookings is reduced so bookings won't occur after the specified date.
- **Maximum booking lead time (days)** This setting specifies the maximum number of days in advance that the resource can be booked. Valid input is an integer between 0 and 1080. The default value is 180 days.
- **Maximum duration (hours)** This setting specifies the maximum duration that the resource can be reserved in a booking request. The default value is 24 hours.

For repeating booking requests, the maximum booking duration applies to the length of each instance of the repeating booking request.

There is also a box on this page that you can use to write a message that will be sent to users who send meeting requests to reserve the resource.

Contact Information

Use the **Contact Information** section to view or change the contact information for the resource. The information on this page is displayed in the address book.

Tip:
You can use the **State/Province** box to create recipient conditions for dynamic distribution groups, email address policies, or address lists.

Email Address

Use the **Email Address** section to view or change the email addresses associated with the equipment mailbox. This includes the mailbox's primary SMTP address and any associated proxy addresses. The primary SMTP address (also known as the *reply address*) is displayed in bold text in

the address list, with the uppercase **SMTP** value in the **Type** column.

- **Add** Click **Add +** to add a new email address for this mailbox. Select one of following address types:
  - **SMTP** This is the default address type. Click this button and then type the new SMTP address in the *** Email address** box.
  - **EUM** An EUM (Exchange Unified Messaging) address is used by the Microsoft Exchange Unified Messaging service to locate UM-enabled recipients within an Exchange organization. EUM addresses consist of the extension number and the UM dial plan for the UM-enabled user. Click this button and type the extension number in the **Address/Extension** box. Then click **Browse** and select a dial plan for the mailbox.
  - **Custom address type** Click this button and type one of the supported non-SMTP email address types in the *** Email address** box.

Note:
With the exception of X.400 addresses, Exchange doesn't validate custom addresses for correct formatting. You must make sure that the custom address you specify complies with the format requirements for that address type.

Note:
When you add a new email address, you have the option to make it the primary SMTP address.

- **Automatically update email addresses based on the email address policy applied to this recipient** Select this check box to have the recipient's email addresses automatically updated based on changes made to email address policies in your organization.

MailTip

Use the **MailTip** section to add a MailTip to alert users of potential issues before they send a booking request to the equipment mailbox. A MailTip is text that's displayed in the InfoBar when this recipient is added to the To, Cc, or Bcc lines of a new email message.

Note:
MailTips can include HTML tags, but scripts aren't allowed. The length of a custom MailTip can't exceed 175 displayed characters. HTML tags aren't counted in the limit.

Use the Shell to change equipment mailbox properties

Use the following sets of cmdlets to view and change equipment mailbox properties:**Get-Mailbox** and **Set-Mailbox** cmdlets to view and change general properties and email addresses for equipment mailboxes. Use the **Get-CalendarProcessing** and **Set-CalendarProcessing** cmdlets to view and change delegates and booking options.

- **Get-User** and **Set-User** Use these cmdlets to view and set general properties such as department and company names.
- **Get-Mailbox** and **Set-Mailbox** Use these cmdlets to view and set mailbox properties, such as email addresses and the mailbox database.
- **Get-CalendarProcessing** and **Set-CalendarProcessing** Use these cmdlets to view and set booking options and delegates.

For information about these cmdlets, see the following topics:

- Get-User
- Set-User
- Get-Mailbox
- Set-Mailbox
- Get-CalendarProcessing
- Set-CalendarProcessing

Here are some examples of using the Shell to change equipment mailbox properties.

This example changes the display name and primary SMTP address (called the default reply address) for the MotorPool 1 equipment mailbox. The previous reply address is kept as a proxy address.

```
Set-Mailbox "MotorPool 1" -DisplayName "Motor Pool 1 -
Compact" -EmailAddresses
SMTP:MP1.compact@contoso.com,smtp:MP.1@contoso.com
```

This example configures equipment mailboxes to allow booking requests to be scheduled only during working hours.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'EquipmentMailbox')} | Set-
CalendarProcessing -ScheduleOnlyDuringWorkHours $true
```

This example uses the **Get-User** cmdlet to find all equipment mailboxes in the Audio Visual department, and then uses the **Set-CalendarProcessing** cmdlet to send booking requests to a delegate named Ann Beebe to accept or decline.

```
Get-User -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'EquipmentMailbox') -and
(Department -eq 'Audio Visual')} | Set-CalendarProcessing -
AllBookInPolicy $false -AllRequestInPolicy $true -
ResourceDelegates "Ann Beebe"
```

How do you know this worked?

To verify that you've successfully changed properties for an equipment mailbox, do the following:

- In the EAC, select the mailbox and then click **Edit** ✎ to view the property or feature that you changed. Depending on the property that you changed, it might be displayed in the Details pane for the selected mailbox.
- In the Shell, use the **Get-Mailbox** cmdlet to verify the changes. One advantage of using the Shell is that you can view multiple properties for multiple mailboxes. In the example above where booking requests could be scheduled only during working hours, run the following command to

verify the new value.

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'EquipmentMailbox')} | Get-
CalendarProcessing | fl
Identity,ScheduleOnlyDuringWorkHours
```

Disconnected mailboxes

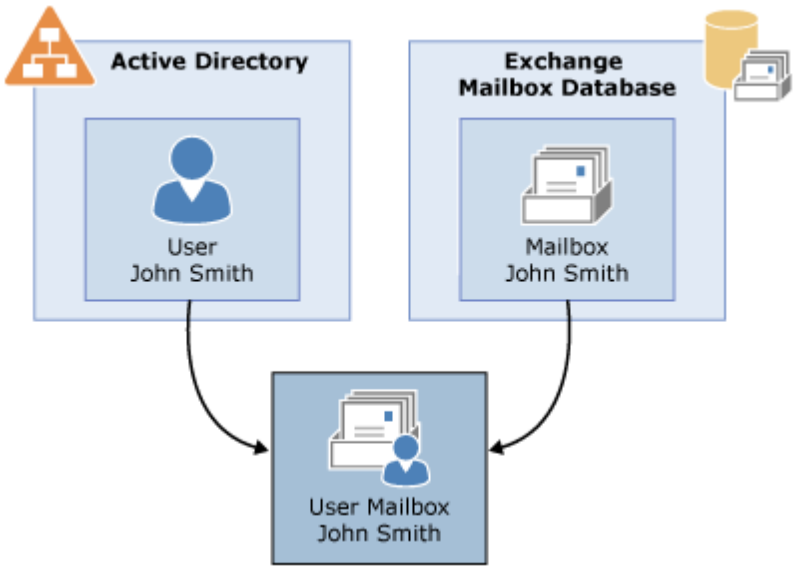
Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-25*

Each Microsoft Exchange mailbox consists of an Active Directory user account and the mailbox data stored in the Exchange mailbox database. All configuration data for a mailbox is stored in the Exchange attributes of the Active Directory user object. The mailbox database contains the mail data that's in the mailbox associated with the user account. The following figure shows the components of a mailbox.

Mailbox components



A *disconnected mailbox* is a mailbox object in the mailbox database that isn't associated with an Active Directory user account. There are two types of disconnected mailboxes:

- **Disabled mailboxes** When a mailbox is disabled or deleted in the Exchange Administration Center (EAC) or using the **Disable-Mailbox** or **Remove-Mailbox** cmdlet in the Exchange Management Shell, Exchange retains the deleted mailbox in the mailbox database, and switches the mailbox to a disabled state. This is why mailboxes that are either disabled or deleted are referred to as *disabled mailboxes*. The difference is that when you disable a mailbox, the Exchange attributes are removed from the corresponding Active Directory user account, but the user

account is retained. When you delete a mailbox, both the Exchange attributes and the Active Directory user account are deleted.

Disabled and deleted mailboxes are retained in the mailbox database until the deleted mailbox retention period expires, which is 30 days by default. After the retention period expires, the mailbox is permanently deleted (also called *purged*). If a mailbox is deleted using the **Remove-Mailbox** cmdlet, it's also retained for the duration of the retention period.

Important:
If a mailbox is deleted using the **Remove-Mailbox** cmdlet and either the *Permanent* or *StoreMailboxIdentity* parameter, it will be immediately deleted from the mailbox database.

To identify the disabled mailboxes in your organization, run the following command in the Shell.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisconnectReason -eq "Disabled" } | ft
DisplayName,Database,DisconnectDate
```

- **Soft-deleted mailboxes** When a mailbox is moved to a different mailbox database, Exchange doesn't fully delete the mailbox from the source mailbox database when the move is complete. Instead, the mailbox in the source mailbox database is switched to a *soft-deleted* state. Like disabled mailboxes, soft-deleted mailboxes are retained in the source database either until the deleted mailbox retention period expires or until the **Remove-StoreMailbox** cmdlet is used to purge the mailbox.

Run the following command to identify soft-deleted mailboxes in your organization.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisconnectReason -eq "SoftDeleted" } | ft
DisplayName,Database,DisconnectDate
```

Contents

Working with disabled mailboxes

You can perform several operations on a disabled mailbox before it's purged from the mailbox database:

- Reconnect it to the same user account.
- Connect it to a different user account that isn't mail-enabled, which means the user account doesn't have a mailbox.

- Restore it to a user account that has an existing mailbox. For example, if a user whose mailbox was deleted has a new mailbox, you can restore the user's disabled mailbox to their new mailbox.
- Permanently delete it from the Exchange mailbox database.

Connecting or restoring a disabled mailbox

Here are scenarios in which you may want to connect or restore a disabled mailbox before the mailbox retention period expires or before it's permanently deleted:

- You disabled a mailbox and now want to reconnect the mailbox to the same Active Directory user account.
- You deleted a mailbox by using the EAC or the Remove-Mailbox cmdlet and now want to reconnect the mailbox to a different Active Directory user account.
- You deleted a mailbox and now want to restore the mailbox to an existing mailbox. For example, if a user whose mailbox was deleted has a new mailbox, you can restore the user's disabled mailbox to their new mailbox.
- You want to convert a user mailbox to a linked mailbox associated with a user account that's external to the forest in which your Exchange organization exists. The resource forest scenario is an example of when you would want to associate a mailbox with an external account. In this scenario, user objects in the Exchange forest have mailboxes, but the user objects are disabled for logon. You must associate a mailbox in the Exchange forest with a user account in the external account forest.

There are two ways you can reconnect or restore a disabled mailbox. The first method is to use the EAC or the **Connect-Mailbox** cmdlet to connect a disabled mailbox to a user account. For procedures to reconnect disabled mailboxes, see Connect a disabled mailbox.

The second method uses the **New-MailboxRestoreRequest** cmdlet to merge the contents of the disabled mailbox with an existing mailbox. This cmdlet uses the Mailbox Replication Service (MRS) to restore the mailbox. For procedures to restore disabled mailboxes, see Connect or restore a deleted mailbox.

Permanently deleting a disabled mailbox

As stated previously, Exchange retains disabled mailboxes in the mailbox database based on the deleted mailbox retention settings configured for that mailbox database. After the specified retention period, a disabled mailbox is purged from the Exchange mailbox database. You can also permanently delete a disabled mailbox and all its message content from the mailbox database by using the **Remove-StoreMailbox** cmdlet. After a disabled mailbox is automatically purged or permanently deleted by an administrator, the data loss is permanent and the mailbox can't be recovered.

For more information, see Permanently delete a mailbox.

Return to top

Working with disabled archive mailboxes

Archive mailboxes become disconnected when they're disabled. Similar to a disabled primary mailbox, a disconnected archive mailbox can be connected by using the **Connect-Mailbox** cmdlet with the *Archive* parameter.

The primary mailbox and the archive mailbox share the same legacy distinguished name (DN), so you must connect the archive mailbox to the same user mailbox that it was previously connected to. You can't connect the archive mailbox to a different user mailbox.

You can perform two operations on a disconnected archive mailbox:

- **Connect it to an existing primary mailbox** Like a disconnected primary mailbox, a disconnected archive mailbox is retained in the mailbox database until the deleted mailbox retention period expires, which is 30 days by default. During this time, you can recover the archive mailbox by reconnecting it to the same user account that it was connected to before it was disabled.

Note:

If you disable an archive mailbox for a user mailbox and then enable an archive mailbox for that same user, that user mailbox will get a new archive mailbox. While you can use the **Connect-Mailbox** cmdlet to connect a primary mailbox to a user, you must use the **Enable-Mailbox** cmdlet to connect a disabled archive mailbox to an existing mailbox.

For more information, see Manage In-Place Archives.

- **Permanently delete it from the Exchange mailbox database** Exchange retains disconnected archive mailboxes based on the deleted mailbox retention settings configured for the mailbox database. The default retention period is 30 days. After the specified mailbox retention period, a disconnected archive mailbox is purged from the Exchange mailbox database.

Like a disabled primary mailbox, you can permanently delete a disabled archive mailbox at any time by using the **Remove-StoreMailbox** cmdlet. For more information, see Permanently delete a mailbox.

Return to top

Working with soft-deleted mailboxes

A soft-deleted mailbox is created when a mailbox is moved from one Exchange mailbox database to any other mailbox database. Exchange doesn't fully delete the mailbox from the source database after a move in case an error occurs during the move that causes the mailbox on the destination database to fail. You can always restore the source mailbox and try again. Exchange will retain the soft-deleted mailbox for the duration of the mailbox retention period.

You can perform two operations on a soft-deleted mailbox:

- Restore it to an existing mailbox.
- Permanently delete it from the Exchange mailbox database.

The procedures for restoring and permanently deleting a soft-deleted mailbox are similar to those for a disabled mailbox. For more information, see the following topics:

- Restore a soft-deleted mailbox
- Permanently delete a mailbox

Return to top

Summary of working with disconnected mailboxes

The following table summarizes the information about disconnected mailboxes, including how the mailbox was disconnected, what happens to the corresponding Active Directory user account when a mailbox is disconnected, and the options and tools you have to connect or restore disconnected mailboxes.

How mailbox was disabled	Value of *DisconnectReason* property	Is Active Directory user account retained?	Connect or restore options	Tools
• The EAC: **Recipients** > **Mailboxes** > **Disable** • The Shell: **Disable-Mailbox** cmdlet	Disabled	Yes	Connect to same user account	• The EAC: **Recipients** > **Mailboxes** > **Connect a Mailbox** • The Shell: **Connect-Mailbox** cmdlet
• The EAC: **Recipients** > **Mailboxes** > **Delete** • The Shell: **Remove-Mailbox** cmdlet	Disabled	No	• Connect to a different user account • Restore to a different mailbox	• The EAC: **Recipients** > **Mailboxes** > **Connect a Mailbox** • The Shell: **Connect-Mailbox** cmdlet • **Enable-Mailbox** • The Shell: **New-MailboxRestore** cmdlet

Moved to a different mailbox database	SoftDeleted	Yes	• Connect to a different user account • Restore to a different mailbox	• The EAC: **Recipients** > **Mailboxes** > **Connect a Mailbox** • The Shell: **Connect-Mailbox** cmdlet • **Enable-Mailbox** • The Shell: **New-MailboxRestore** cmdlet

Return to top

Disconnected mailbox documentation

The following table contains links to topics that will help you manage disconnected mailboxes. This includes managing disconnected user mailboxes, linked mailboxes, resource mailboxes, and shared mailboxes.

Topic	Description
Disable or delete a mailbox	Learn how to disable or delete mailboxes.
Connect a disabled mailbox	Learn how to connect a disabled mailbox to an existing user account.
Connect or restore a deleted mailbox	Learn how to connect a deleted mailbox to a user account or restore the contents of a deleted mailbox to an existing mailbox.
Restore a soft-deleted mailbox	Learn how to connect a soft-deleted mailbox to a user account or restore a soft-deleted mailbox to an existing mailbox.
Manage mailbox restore requests	Learn how to manage mailbox restore requests using the Shell.

Permanently delete a mailbox	Learn how to permanently delete a mailbox.

Return to top

Disable or delete a mailbox

Topic Last Modified: *2014-08-11*

You can use the EAC or the Shell to disable or delete a mailbox. When a mailbox is disabled or deleted, Exchange retains the mailbox in the mailbox database and switches the mailbox to a disabled state. Disabled and deleted mailboxes are retained in the mailbox database until the deleted mailbox retention period expires, which is 30 days by default. After the retention period expires, the mailbox is permanently deleted or *purged*.

Note:
Disabled or deleted mailboxes are referred to as *disconnected mailboxes*.

The primary difference between deleting and disabling a mailbox is that when you disable a mailbox, the Exchange attributes are removed from the corresponding Active Directory user account, but the user account is retained. When you delete a mailbox, both the Exchange attributes and the Active Directory user account are deleted. This difference also determines your options to reconnect or restore disabled and deleted mailboxes.

The following table shows which types of Exchange mailboxes you can disable and delete.

Mailbox type	Disable?	Delete?
Archive mailbox	Yes	No *
Linked mailbox	Yes	Yes
Resource mailbox (Room or Equipment)	No	Yes
Shared mailbox	Yes	Yes
User mailbox	Yes	Yes

* If an archive mailbox is enabled, it will be deleted when the primary mailbox is deleted. For information about disabling archive mailboxes, see Manage In-Place Archives.

If an administrator deletes a user account that has a mailbox, the Exchange Information store will eventually detect that the mailbox is no longer connected to a user account and mark that mailbox

for deletion, even if the mailbox is on hold. If you want to retain the mailbox you must do the following:

1. Instead of deleting the user account, disable the user account.
2. Change the properties of the mailbox to restrict its use and access to the mailbox. For example, set send and receive quotas equal to 1, block who can send messages to the mailbox, and restrict who can access the mailbox.
3. Retain the mailbox until all data has been expunged, or until hold is no longer required.

For additional management tasks related to disconnected mailboxes, see the following topics:

- Disconnected mailboxes
- Connect a disabled mailbox
- Connect or restore a deleted mailbox
- Permanently delete a mailbox

What do you need to know before you begin?

- Estimated time to complete each procedure: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Disable a mailbox

As previously stated, when you disable a mailbox, the Exchange attributes are removed from the corresponding Active Directory user account, but the user account is retained.

Use the EAC to disable a mailbox

The following procedure shows how to disable a user mailbox. Use the same procedure to disable other mailbox types after navigating to the appropriate page in the EAC.

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to disable.
3. Click **More** ··· and then click **Disable**.
4. A warning appears asking if you're sure you want to disable the mailbox. Click **Yes** to disable the mailbox.

The mailbox is removed from the mailbox list.

Use the Shell to disable a mailbox

Use the following command to disable user mailboxes, linked mailboxes, resource mailboxes, and shared mailboxes.

```
Disable-Mailbox <identity>
```

When you run this command, a message is displayed that asks you to confirm that you want to disable the mailbox.

Here are some examples of commands for disabling mailboxes.

```
Disable-Mailbox danj

Disable-Mailbox "Conf Room 31/1234 (12)"

Disable-Mailbox sharedmbx@contoso.com
```

How do you know this worked?

To verify that you've successfully disabled a mailbox, do one of the following:

- In the EAC, click **Recipients**, navigate to the appropriate page for the mailbox type that you disabled, and then verify that the mailbox is no longer listed.
- In Active Directory Users and Computers, right-click the user account whose mailbox you disabled, and then click **Properties**. On the **General** tab, notice that the **E-mail** field is blank. This verifies that the mailbox is disabled, but the user account still exists.
- In the Shell, run the following command.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisplayName -eq "<display name>" } | fl
DisconnectReason,DisconnectDate
```

The `Disabled` value in the *DisconnectReason* property indicates that the mailbox is disabled.

Note:
When you delete a mailbox, the value in the *DisconnectReason* property is also `Disabled`. However, the corresponding Active Directory user account is deleted.

- In the Shell, run the following command.

```
Get-User <identity>
```

Note that that value for the *RecipientType* property is `User`, instead of `UserMailbox`, which is the value for users with enabled mailboxes. This also verifies that the mailbox is disabled, but the user account is retained.

Delete a mailbox

As previously stated, when you delete a mailbox, both the Exchange attributes and the Active Directory user account are deleted. The mailbox (and the archive mailbox, if it's enabled) will be permanently deleted from the mailbox database after the mailbox retention period expires.

Use the EAC to delete a mailbox

The following procedure shows how to delete a user mailbox. Use the same procedure to delete other mailbox types after navigating to the appropriate page in the EAC.

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of user mailboxes, click the mailbox that you want to delete, and then click **Delete** 🗑.
3. A warning appears asking if you're sure you want to delete the mailbox. Click **Yes** to delete the mailbox.

The mailbox is removed from the mailbox list.

Use the Shell to delete a mailbox

Use the following command to delete user mailboxes, linked mailboxes, resource mailboxes, and shared mailboxes.

```
Remove-Mailbox <identity>
```

When you run this command, a message is displayed that asks you to confirm that you want to remove the mailbox and the corresponding Active Directory user account.

Here are some examples of commands for deleting mailboxes.

```
Remove-Mailbox pilarp@contoso.com

Remove-Mailbox "Fleet Van (16)"

Remove-Mailbox corpprint
```

How do you know this worked?

To verify that you've successfully deleted a mailbox, do one of the following sets of verification procedures.

1. In the EAC, click **Recipients** and then navigate to the appropriate page for the mailbox type that you deleted, and verify that the mailbox is no longer listed.
2. In Active Directory Users and Computers, verify that the corresponding user account is no longer listed.

Or

1. Run the following command to verify that the mailbox has been deleted.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisplayName -eq "<display name>" } | fl
DisconnectReason,DisconnectDate
```

The `Disabled` value in the *DisconnectReason* property indicates that the mailbox has been deleted.

Note:
When you disable a mailbox, the value in the *DisconnectReason* property is also `Disabled`. However, the corresponding Active Directory user account is retained.

2. Run the following command to verify that Active Directory user account has been deleted.

```
Get-User <identity>
```

The command will return an error stating that user couldn't be found, verifying that the account was deleted.

Connect a disabled mailbox

Exchange Server 2013 > Recipients > Disconnected mailboxes >

Applies to: *Exchange Online*

Topic Last Modified: *2012-11-13*

You can use the EAC or the Shell to connect a disabled mailbox to an Active Directory user account. When you disable a mailbox, Exchange retains the mailbox in the mailbox database and switches the mailbox to a disabled state. The Exchange attributes are also removed from the corresponding Active Directory user account, but the user account is retained. The mailbox is retained until the deleted mailbox retention period expires, which is 30 days by default, and then it's permanently deleted (or *purged*) from the mailbox database.

Until a disabled mailbox is permanently deleted from the Exchange mailbox database, you can use the EAC or the Shell to reconnect it to the original Active Directory user account.

To learn more about disconnected mailboxes and perform other related management tasks, see the following topics:

- Disconnected mailboxes
- Disable or delete a mailbox
- Connect or restore a deleted mailbox
- Permanently delete a mailbox

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

- Run the **Get-User** cmdlet in the Shell to verify that the Active Directory user account that you want to connect the disabled mailbox to exists and that it isn't already associated with another mailbox. To connect a disabled mailbox to a user account, the account must exist and the value for the *RecipientType* property has to be `User`, which indicates that the account isn't already mailbox-enabled.

For on-premises Exchange organizations, you can also verify this information in Active Directory Users and Computers.

- Run the following command to verify that the disabled mailbox that you want to connect a user account to exists in the mailbox database and isn't a soft-deleted mailbox.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisplayName -eq "<display name>" } | fl
DisplayName,Database,DisconnectReason
```

To be able to connect a disabled mailbox, the mailbox has to exist in the mailbox database and the value for the *DisconnectReason* property has to be `Disabled`. If the mailbox has been purged from the database, the command won't return any results.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to connect a disabled mailbox

The following procedure shows how to connect a disabled user mailbox. You can also reconnect disabled linked mailboxes and disabled shared mailboxes to the corresponding user account.

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. Click **More** ···, and then click **Connect a mailbox**.

A list of mailboxes that are disconnected on the selected Exchange server in your Exchange organization will be displayed.

Note:

This list of disconnected mailboxes includes disabled mailboxes, deleted mailboxes, and soft-deleted mailboxes.

3. Click the disabled mailbox that you want to reconnect, and then click **Connect**.
4. In the window that asks if you're sure that you want to reconnect the mailbox, click **Yes**.

Exchange will reconnect the disabled mailbox to the corresponding user account.

Use the Shell to connect a disabled mailbox

Use the **Connect-Mailbox** cmdlet in the Shell to connect a user account to a disabled mailbox. You have to specify the type of mailbox that you're connecting. The following examples show the syntax for reconnecting user, linked, and shared mailboxes.

This example connects a user mailbox. The *Identity* parameter specifies the disconnected mailbox in the Exchange database. The *User* parameter specifies the Active Directory user account to reconnect the mailbox to.

```
Connect-Mailbox -Identity "Jeffrey Zeng" -Database MBXDB01
-User "Jeffrey Zeng"
```

This example connects a linked mailbox. The *Identity* parameter specifies the disconnected mailbox in the Exchange database. The *LinkedMasterAccount* parameter specifies the Active Directory user account in the account forest that you want to reconnect the mailbox to. The *Alias* parameter specifies the alias, which is the portion of the email address on the left side of the at (@) symbol, for the reconnected mailbox.

```
Connect-Mailbox -Identity "Kai Axford" -Database MBXDB02 -
LinkedDomainController FabrikamDC01 -LinkedMasterAccount
kai.axford@fabrikam.com -Alias kaia
```

This example connects a shared mailbox.

```
Connect-Mailbox -Identity "Corporate Shared Mailbox" -
Database "Mailbox Database 03" -User "Corporate Shared
Mailbox" -Alias corpshared -Shared
```

Note:

If you don't include the *Alias* parameter when you run the **Connect-Mailbox** cmdlet, the value specified in the *User* or *LinkedMasterAccount* parameter is used to create the email address alias for the reconnected mailbox.

For detailed syntax and parameter information, see Connect-Mailbox.

How do you know this worked?

To verify that you've successfully connected a disabled mailbox to a user account, do one of the following:

- In the EAC, click **Recipients**, navigate to the appropriate page for the mailbox type that you reconnected, click **Refresh**, and verify that the mailbox is listed.
- In Active Directory Users and Computers, right-click the user account whose mailbox you disabled, and then click **Properties**. On the **General** tab, notice that the **E-mail** box is populated

with the email address for the reconnected mailbox.
- In the Shell, run the following command.

```
Get-User <identity>
```

The **UserMailbox** value for the *RecipientType* property indicates that the user account and the mailbox are connected. You can also run the **Get-Mailbox** cmdlet to verify that the mailbox exists.

Connect or restore a deleted mailbox

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-12*

You can use the EAC or the Shell to connect a deleted mailbox to an Active Directory user account. When you delete a mailbox, Exchange retains the mailbox in the mailbox database and switches the mailbox to a disabled state. The associated Active Directory user account is also deleted. The mailbox is retained until the deleted mailbox retention period expires, which is 30 days by default, and then it's permanently deleted (or *purged*) from the mailbox database.

Until a deleted mailbox is permanently deleted from the Exchange mailbox database, you can use the EAC or the Shell to connect it to an Active Directory user account. You can also use the Shell to restore the contents of the deleted mailbox to an existing mailbox.

To learn more about disconnected mailboxes and perform other related management tasks, see the following topics:
- Disconnected mailboxes
- Disable or delete a mailbox
- Connect a disabled mailbox
- Permanently delete a mailbox

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- Create a new user account in Active Directory to connect the deleted mailbox to. Or use the **Get-User** cmdlet in the Shell to verify that the Active Directory user account that you want to connect the deleted mailbox to exists and that it isn't already associated with another mailbox. To connect a deleted mailbox to a user account, the account must exist and the value for the *RecipientType* property has to be `User`, which indicates that the account isn't already mailbox-

enabled.

For on-premises Exchange organizations, you can also verify this information in Active Directory Users and Computers.

Important:

When you connect deleted linked mailboxes, resource mailboxes, or shared mailboxes, the Active Directory user account that you're connecting the mailbox to must be disabled.

- To verify that the deleted mailbox that you want to connect a user account to exists in the mailbox database and isn't a soft-deleted mailbox, run the following command.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisplayName -eq "<display name>" } | fl
DisplayName,Database,DisconnectReason
```

The deleted mailbox has to exist in the mailbox database and the value for the *DisconnectReason* property has to be `Disabled`. If the mailbox has been purged from the database, the command won't return any results.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.
- Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Connect a deleted mailbox

When you connect a deleted mailbox, you associate the mailbox with a user account that isn't mail-enabled, which means that it doesn't have an existing mailbox. To connect a deleted mailbox to a user account that has a mailbox, you have to restore the deleted mailbox. For more information, see Restore a deleted mailbox later in this topic.

Use the EAC to connect a deleted mailbox

The following procedure shows how to connect a deleted user mailbox to a user account. You can also use this procedure to connect linked mailboxes, resource mailboxes, and shared mailboxes that have been deleted to a user account.

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. Click **More** ···, and then click **Connect a mailbox**.

A list of mailboxes that are disconnected on the selected Exchange server in your Exchange organization will be displayed.

Note:

This list of disconnected mailboxes includes disabled mailboxes, deleted mailboxes, and soft-deleted mailboxes.

3. Click the deleted mailbox that you want to connect a user to, and then click **Connect**.
4. In the window that asks if you're sure that you want to connect the mailbox, click **Yes**.

A list of user accounts that aren't mail-enabled is displayed.

5. Click the user that you want to connect the deleted mailbox to, and then click **OK**.

Exchange will connect the deleted mailbox to the user account that you selected.

Use the Shell to connect a deleted mailbox

Use the **Connect-Mailbox** cmdlet in the Shell to connect a deleted mailbox to a user account that isn't mail enabled. You have to specify the type of mailbox that you're connecting. The following examples show the syntax for reconnecting user, linked, room, equipment, and shared mailboxes. In all examples, the optional *Alias* parameter is used to specify the email alias, which is the portion of the email address on the left side of the at (@) symbol. If you don't include the *Alias* parameter, the value specified in the *User* or *LinkedMasterAccount* parameter is used to create the alias for the email address for the reconnected mailbox.

Note:

As previously stated, when you connect linked, resource, or shared mailboxes, the Active Directory user account that you're linking the mailbox to must be disabled.

This example connects a user mailbox. The *Identity* parameter specifies the display name of the deleted mailbox retained in the mailbox database named MBXDB01. The *User* parameter specifies the Active Directory user account to connect the mailbox to.

```
Connect-Mailbox -Identity "Paul Cannon" -Database MBXDB01 -
User "Robin Wood" -Alias robinw
```

Note:

You can also use the values for the `LegacyDN` or `MailboxGuid` properties to identify the deleted mailbox.

This example connects a linked mailbox. The *Identity* parameter specifies the deleted mailbox on the mailbox database named MBXDB02. The *LinkedMasterAccount* parameter specifies the Active Directory user account in the account forest that you want to connect the mailbox to. The *LinkedDomainController* parameter specifies a domain controller in the account forest.

```
Connect-Mailbox -Identity "Temp User" -Database MBXDB02 -
LinkedDomainController FabrikamDC01 -LinkedMasterAccount
danpark@fabrikam.com -Alias dpark
```

This example connects a room mailbox.

```
Connect-Mailbox -Identity "rm2121" -Database
"MBXResourceDB" -User "Conference Room 2121" -Alias
ConfRm2121 -Room
```

This example connects an equipment mailbox.

```
Connect-Mailbox -Identity "MotorPool01" -Database
"MBXResourceDB" -User "Van01 (12 passengers)" -Alias van01
-Equipment
```

This example connects a shared mailbox.

```
Connect-Mailbox -Identity "Printer Support" -Database
MBXDB01 -User "Corp Printer Support" -Alias corpprint -
Shared
```

Note:

You can also use the `LegacyDN` or `MailboxGuid` values to identify the deleted mailbox.

For detailed syntax and parameter information, see Connect-Mailbox.

How do you know this worked?

To verify that you've successfully connected a deleted mailbox to a user account, do one of the following:

- In the EAC, click **Recipients**, navigate to the appropriate page for the mailbox type that you connected, click **Refresh**, and verify that the mailbox is listed.
- In Active Directory Users and Computers, right-click the user account that you connected to the mailbox, and then click **Properties**. On the **General** tab, notice that the **E-mail** box is populated with the email address for the connected mailbox.
- In the Shell, run the following command.

```
Get-User <identity>
```

The **UserMailbox** value for the *RecipientType* property indicates that the user account and the mailbox are connected. You can also run the **Get-Mailbox <identity>** command to verify that the mailbox was connected.

Restore a deleted mailbox

You can use the Shell to restore a deleted mailbox to an existing mailbox using the **New-MailboxRestoreRequest** cmdlet. When you restore a deleted mailbox, its contents are copied to an existing mailbox, which is referred to as the *target mailbox*. After a deleted mailbox is restored, it's still retained in the mailbox database until it's permanently deleted by an administrator or purged after the deleted mailbox retention period expires.

After a mailbox restore request is successfully completed, it's retained for 30 days, by default, before it's removed. You can remove it sooner by using the **Remove-StoreMailbox** cmdlet.

Note:

You can't use the EAC to restore a deleted mailbox.

Use the Shell to restore a deleted mailbox

To create a mailbox restore request, you have to use the display name, legacy distinguished name (DN), or mailbox GUID of the deleted mailbox. Use the **Get-MailboxStatistics** cmdlet to display the values of the `DisplayName`, `MailboxGuid`, and `LegacyDN` properties for the deleted mailbox that you want to restore. For example, run the following command to return this information for all disabled and deleted mailboxes in your organization.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where {$_.DisconnectReason -eq "Disabled"} | fl DisplayName,MailboxGuid,LegacyDN,Database
```

This example restores the deleted mailbox, which is identified by the *SourceStoreMailbox* parameter and is located on the MBXDB01 mailbox database, to the target mailbox Debra Garcia. The *AllowLegacyDNMismatch* parameter is used so the source mailbox can be restored to a different mailbox, one that doesn't have the same legacy DN value.

```
New-MailboxRestoreRequest -SourceStoreMailbox e4890ee7-79a2-4f94-9569-91e61eac372b -SourceDatabase MBXDB01 -TargetMailbox "Debra Garcia" -AllowLegacyDNMismatch
```

This example restores Pilar Pinilla's deleted archive mailbox to her current archive mailbox. The *AllowLegacyDNMismatch* parameter isn't necessary because a primary mailbox and its corresponding archive mailbox have the same legacy DN.

```
New-MailboxRestoreRequest -SourceStoreMailbox "Personal Archive - Pilar Pinilla" -SourceDatabase "MDB01" -TargetMailbox pilarp@contoso.com -TargetIsArchive
```

For detailed syntax and parameter information, see New-MailboxRestoreRequest.

How do you know this worked?

To verify that you've successfully restored a deleted mailbox to the target mailbox, run the **Get-MailboxRestoreRequest** cmdlet to display information about the restore request. If the restore request was successfully created, the *Status* property will have a value of `Queued`, `InProgress`, or `Completed`. After the restore request is completed, the contents from the deleted mailbox will appear in the target mailbox.

For more information, see:

- Manage mailbox restore requests
- Get-MailboxRestoreRequest
- Get-MailboxRestoreRequestStatistics

Restore a soft-deleted mailbox

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-29*

Use the Shell to connect a soft-deleted mailbox to an Active Directory user account. A mailbox becomes *soft-deleted* in the source mailbox database when it's moved to a different mailbox database. Exchange doesn't fully delete the mailbox from the source mailbox database when the move is complete. Instead, the mailbox in the source mailbox database is switched to a soft-deleted state. This lets you restore the source mailbox in case errors occur during the move that cause a failure or corruption of the mailbox on the destination database. If this happens, you can restore the source mailbox and try the move again.

A soft-deleted mailbox is retained in the source database until the deleted mailbox retention period expires or until the **Remove-StoreMailbox** cmdlet is used to purge the soft-deleted mailbox. Until a soft-deleted mailbox is permanently deleted from the Exchange mailbox database, you can use the Shell to restore the contents of the soft-deleted mailbox to an existing mailbox or an archive mailbox.

To learn more about soft-deleted mailboxes and perform other related management tasks, see the following topics:

- Disconnected mailboxes
- Connect or restore a deleted mailbox
- Manage mailbox restore requests
- Permanently delete a mailbox

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- The procedures in this topic can only be performed in the Shell. You can't use the EAC to restore soft-deleted mailboxes.
- Run the following command to verify that the soft-deleted mailbox that you want to connect a user account still exists in the mailbox database and is not a disabled mailbox.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisplayName -eq "<display name>" } | fl
DisplayName,DisconnectReason,DisconnectDate
```

The soft-deleted mailbox has to exist in the mailbox database and the value for the *DisconnectReason* property has to be `SoftDeleted`. If the mailbox has been purged from the database, the command won't return any results.

Alternatively, run the following command to display all soft-deleted mailboxes in your organization.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisconnectReason -eq "SoftDeleted" } | fl
DisplayName,DisconnectReason,DisconnectDate
```

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.
- Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to restore a soft-deleted mailbox

You can use the Shell to restore a soft-deleted mailbox to an existing mailbox by using the **New-MailboxRestoreRequest** cmdlet. When you restore a soft-deleted mailbox, its contents are copied to an existing mailbox, which is called the *target mailbox*. After a mailbox restore request is successfully completed, the request is retained for 30 days, by default, before it's removed. You can remove it sooner by using the **Remove-MailboxRestoreRequest** cmdlet.

After a soft-deleted mailbox is restored, the mailbox is retained in the mailbox database until it's permanently deleted by an administrator or purged when the deleted mailbox retention period expires.

To create a mailbox restore request, you have to use the display name, mailbox GUID, or legacy distinguished name (DN) of the soft-deleted mailbox. Use the **Get-MailboxStatistics** cmdlet to display the values of the **DisplayName**, **MailboxGuid**, and **LegacyDN** properties for the soft-deleted mailbox that you want to restore. For example, run the following command to return this information for all disabled and soft-deleted mailboxes in your organization.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{$_.DisconnectReason -eq "SoftDeleted"} | fl
DisplayName,MailboxGuid,LegacyDN,Database
```

This example restores a soft-deleted mailbox, which is identified by the display name in the *SourceStoreMailbox* parameter and is located on the MBXDB01 mailbox database, to the target mailbox named Debra Garcia. The *AllowLegacyDNMismatch* parameter is used so the source mailbox can be restored to a mailbox that doesn't have the same legacy DN value as the soft-deleted mailbox.

```
New-MailboxRestoreRequest -SourceStoreMailbox "Debra
```

```
Garcia" -SourceDatabase MBXDB01 -TargetMailbox "Debra
Garcia" -AllowLegacyDNMismatch
```

This example restores Pilar Pinilla's soft-deleted archive mailbox, which is identified by the mailbox GUID, to her current archive mailbox. The *AllowLegacyDNMismatch* parameter isn't necessary because a primary mailbox and its corresponding archive mailbox have the same legacy DN.

```
New-MailboxRestoreRequest -SourceStoreMailbox dc35895a-
a628-4bba-9aa9-650f5cdb9ae7 -SourceDatabase MBXDB02 -
TargetMailbox pilarp@contoso.com -TargetIsArchive
```

For detailed syntax and parameter information, see New-MailboxRestoreRequest.

How do you know this worked?

To verify that you've successfully restored a soft-deleted mailbox to the target mailbox, run the **Get-MailboxRestoreRequest** cmdlet or the **Get-MailboxRestoreRequestStatistics** cmdlet to display information about the restore request. If the restore request was successfully created, the *Status* property will have a value of **Queued**, **InProgress**, or **Completed**. After the restore request is completed, the contents from the soft-deleted mailbox will appear in the target mailbox.

For more information, see:

- Manage mailbox restore requests
- Get-MailboxRestoreRequest
- Get-MailboxRestoreRequestStatistics

Manage mailbox restore requests

Exchange Server 2013 > Recipients > Disconnected mailboxes >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-05-30*

Mailbox restore requests are used to restore disconnected mailboxes. A disconnected mailbox is a mailbox in an Exchange mailbox database that isn't associated with an Active Directory user account. Mailboxes become disconnected when they're disabled, deleted, or moved to another database. For more information, see Disconnected mailboxes.

Disconnected mailboxes remain in the mailbox database for the duration specified in the deleted mailbox retention settings for the mailbox database. By default, disconnected mailboxes are retained for 30 days. During this retention period, the contents of a deleted mailbox can be restored (copied) to an existing mailbox. This topic describes how to use the Shell to manage mailbox restore requests.

For additional management tasks related to disconnected mailboxes, see the following topics:

Disable or delete a mailbox

Connect a disabled mailbox

Connect or restore a deleted mailbox

Restore a soft-deleted mailbox

Permanently delete a mailbox

What do you need to know before you begin?

- Estimated time to complete each procedure: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox restore request" entry in the Recipients Permissions topic.
- The procedures in this topic can only be performed in the Shell. You can't use the EAC to manage mailbox restore requests.
- To display the value of the *Identity* property for all mailbox restore requests, run the following command.

```
Get-MailboxRestoreRequest | Format-Table Identity
```

You can use this identity value to specify a specific mailbox restore request when you're performing the procedures in this topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to view restore request properties

You can view the properties of a mailbox restore request, which provide you with basic information about the status of a mailbox restore request.

To display a list and the value of the *Identity* property for all mailbox restore requests, run the following command.

```
Get-MailboxRestoreRequest | Format-Table Identity
```

You can use the identity to get information about specific mailbox restore requests.

This example returns the status of the restore request "Pilar Pinilla \MailboxRestore" using the *Identity* parameter.

```
Get-MailboxRestoreRequest -Identity "Pilar Pinilla \MailboxRestore"
```

This example returns all information for the second restore request for the Pilar Pinilla target mailbox.

```
Get-MailboxRestoreRequest -Identity "Pilar Pinilla \MailboxRestore1" | Format-List
```

This example returns the status of restore requests being restored from the source database MBD01.

```
Get-MailboxRestoreRequest -SourceDatabase MBD01
```

This example returns all restore requests that are currently in progress.

```
Get-MailboxRestoreRequest -Status InProgress
```

Other useful status states include `Queued`, `Completed`, `Suspended`, and `Failed`.

This example returns all restore requests that have been suspended.

```
Get-MailboxRestoreRequest -Suspend $true
```

For detailed syntax and parameter information, see Get-MailboxRestoreRequest.

Get-MailboxRestoreRequest Output

By default, the **Get-MailboxRestoreRequest** cmdlet returns the name of the request, the target mailbox to which data is being restored, and the status of the request. The following table lists useful information returned if you pipe the cmdlet to the **Format-List** cmdlet.

Value	Description
`SourceDatabase`	Specifies the database that contains the disconnected mailbox that's being restored.
`TargetMailbox`	Specifies the mailbox into which data is being restored.
`Name`	Specifies the name of the request.
`RequestQueue`	Specifies the database on which the Microsoft

	Exchange Mailbox Replication service (MRS) stores the detailed status of the request.
`Status`	Specifies the status of the request.
`Suspend`	Specifies whether the request is suspended. A mailbox restore can be suspended when it's created using the **New-MailboxRestoreRequest** cmdlet with the *Suspend* parameter. It can also be suspended if the mailbox restore operation fails or by an administrator using the **Suspend-MailboxRestoreRequest** cmdlet.
`Identity`	Specifies the identity of the request. This identity is a combination of the target mailbox name and the request name.

How do you know this worked?

Run the **Get-MailboxRestoreRequest** cmdlet to verify that you can view properties for mailbox restore requests. If the cmdlet returns an error, verify that you're using the correct syntax and identity. In some cases, the cmdlet may be successful and not return any results. For example, if you've submitted a mailbox restore request and run the command `Get-MailboxRestoreRequest -Status InProgress` and no results are returned, then none of the restore requests are currently running.

Use the Shell to view restore request statistics

You can view the statistics of a mailbox restore request, which provide you with detailed information that can be used for troubleshooting purposes.

This example returns the default statistics for the restore request danp\MailboxRestore1. By default, the information returned includes name, mailbox, status, and percentage complete.

```
Get-MailboxRestoreRequestStatistics -Identity danp
\MailboxRestore1
```

This example returns the statistics for Dan Park's mailbox and exports the report to a .csv file.

```
Get-MailboxRestoreRequestStatistics -Identity "Dan Park
```

```
\MailboxRestore" | Export-CSV \\SERVER01
\RestoreRequest_Reports\DanPark_Restorestats.csv
```

This example returns additional information about the restore request for Pilar Pinilla's mailbox using the *IncludeReport* parameter and piping the results to the **Format-List** cmdlet.

```
Get-MailboxRestoreRequestStatistics -Identity "Pilar
Pinilla\MailboxRestore" -IncludeReport | Format-List
```

This example returns additional information for all restore requests that have a status of `Failed` using the *IncludeReport* parameter, and then saves the information to the file AllRestoreReports.txt in the location where the command is being run.

```
Get-MailboxRestoreRequest -Status Failed | Get-
MailboxRestoreRequestStatistics -IncludeReport | Format-
List > AllRestoreReports.txt
```

For detailed syntax and parameter information, see Get-MailboxRestoreRequestStatistics and Get-MailboxRestoreRequest.

Get-MailboxRestoreRequestStatistics Output

By default, the Get-MailboxRestoreRequestStatistics cmdlet returns the name of the request, the status of the request, the alias of the target mailbox, and the percentage completed. The following table lists other useful information returned if you pipeline the cmdlet to the **Format-List** cmdlet.

Value	Description
`Name`	Specifies the name of the request.
`Status`	Specifies the status of the request.
`StatusDetail`	Specifies more details about the request status. For example, if the `Status` value returns `InProgress`, the `StatusDetail` value would return the specific stages for the `InProgress` status, such as `CreatingFolderHierarchy` and `CopyingMessages`.
`SyncStage`	Specifies how far along the request is through the restore process.
`Suspend`	Specifies whether the restore request is suspended. This value is `true` in the following

	scenarios: • MRS stopped or is in the process of stopping the request due to a failure. • An administrator suspended the request.
`SourceExchangeGuid`	Specifies the GUID of the source mailbox from which data is being restored.
`SourceRootFolder`	Specifies the name of the root folder in the source mailbox hierarchy from which data is being restored. If this value is blank, data is restored from the folder Top of Information Store.
`SourceDatabase`	Specifies the name of the database on which the source mailbox is located.
`MailboxRestoreFlags`	Specifies that the mailbox being restored is either `Disabled` or `Soft-Deleted`.
`TargetAlias`	Specifies the alias of the target mailbox.
`TargetIsArchive`	Specifies whether the mailbox is being restored into an archive.
`TargetExchangeGuid`	Specifies the GUID of the target mailbox.
`TargetRootFolder`	Specifies the name of the root folder in the target mailbox hierarchy to which data is being restored. If this value is blank, data is restored to the folder Top of Information Store.
`TargetDatabase`	Specifies the name of the database on which the target mailbox is located.
`TargetMailboxIdentity`	Specifies the identity of the target mailbox.
`IncludeFolders`	Specifies the list of folders to include during the

	restore. If this value is blank, no folders were specified when the request was created, and all folders will be restored to the mailbox (unless the *ExcludeFolders* parameter is used to exclude specific folders).
`ExcludeFolders`	Specifies the list of folders to exclude during the restore. If this value is blank, no folders were specified when the request was created, and all folders will be restored to the mailbox (unless the *IncludeFolders* parameter is used to include specific folders).
`ExcludeDumpster`	Specifies whether the Recoverable Items folder was excluded when the request was created.
`ConflictResolutionOption`	Specifies the action for MRS to take if there are matching messages in the target and source folders.
`AssociatedMessagesCopyOption`	Specifies whether the associated messages are copied when the request is processed. Associated messages are special messages that contain hidden data with information about rules, views, and forms.
`BadItemLimit`	Specifies the number of bad items that MRS will skip if the request encounters corrupted messages.
`BadItemsEncountered`	Specifies the number of corrupted messages encountered by the command. If the *BadItemsEncountered* value is greater than the *BadItemLimit* value, the request fails.
`QueuedTimeStamp`	Specifies the date and time at which the request

	was initiated to MRS.
`StartTimeStamp`	Specifies the date and time at which MRS started processing the restore request.
`LastUpdateTimeStamp`	Specifies the date and time at which the last change was made to the request. The change may have been made by an administrator or by MRS.
`SuspendTimeStamp`	Specifies the date and time at which the request was suspended.
`OverallDuration`	Specifies the amount of time it took to complete the request. If the request is in a `Failed` state, this value specifies the amount of time between the request being initiated and the request failing. If the request isn't complete, this value specifies the amount of time between the request being initiated and the **Get-MailboxRestoreRequestStatistics** cmdlet being run.
`TotalSuspendedDuration`	Specifies the amount of time the request was in the `Suspended` state.
`TotalFailedDuration`	Specifies the amount of time the request was in the `Failed` state.
`TotalQueuedDuration`	Specifies the amount of time the request was in the `Queued` state.
`TotalInProgressDuration`	Specifies the amount of time the request was in the `In Progress` state.
`TotalStalledDueToHADuration`	Specifies the amount of time the request was stalled due to high availability.

`MRSServerName`	Specifies the name of the Client Access server that processed the request.
`EstimatedTransferSize`	Specifies the total file size that was restored or the file size that MRS expects to restore if the request is in the `In Progress` state.
`EstimatedTransferItemCount`	Specifies the number of items that were restored or the number of items that MRS expects to restore if the request is in the `In Progress` state.
`BytesTransferredPerMinute`	Specifies the average number of bytes that have been transferred per minute.
`ItemsTransferred`	Specifies the number of items that have been transferred.
`PercentComplete`	Specifies the percentage of the request that has been completed.
`CompletedRequestAgeLimit`	Specifies how long a completed restore request will be retained before it's deleted. The default is 30 days.
`PositionInQueue`	If the request hasn't started, this value specifies the request's position in the queue.
`FailureCode`	If there is a failure, this value specifies the failure code.
`FailureType`	If there is a failure, this value specifies the failure type.
`FailureSide`	If there is a failure, this value specifies whether the failure occurred on the target mailbox or the source mailbox.
`Message`	If there is a failure, this value specifies the failure

	message. This value can also specify the suspend comment.
`FailureTimestamp`	If the request failed, this value specifies the date and time at which the request failed.
`FailureContext`	If the request failed, this value specifies information about the action being performed at the time of failure.
`ValidationMessage`	If the request isn't valid, this value specifies the reason.
`RequestQueue`	Specifies the database on which MRS stores the detailed status of the request.
`Identity`	Specifies the identity of the request.
`Report`	If you used the *IncludeReport* parameter, this value specifies information that can be used to troubleshoot the request.

How do you know this worked?

Run the **Get-MailboxRestoreRequestStatistics** cmdlet to verify that you can view the statistics for mailbox restore requests. If the cmdlet returns an error, verify that you're using the correct identity for the restore request.

Use the Shell to change restore request properties

If a mailbox restore request fails, you can use the **Set-MailboxRestoreRequest** cmdlet to change the request's properties to try to recover from the failure.

This example specifies that the restore request MailboxRestore1 for Debra Garcia's mailbox skips 10 corrupted mailbox items.

```
Set-MailboxRestoreRequest -Identity "Debra Garcia
\MailboxRestore1" -BadItemLimit 10
```

This example specifies that the restore request MailboxRestore1 for Florence Flipo's mailbox skips 100 corrupted items. Because the *BadItemLimit* value is greater than 50, the *AcceptLargeDataLoss* parameter must be specified.

```
Set-MailboxRestoreRequest -Identity "Florence Flipo
\MailboxRestore1" -BadItemLimit 100 -AcceptLargeDataLoss
```

For detailed syntax and parameter information, see Set-MailboxRestoreRequest.

How do you know this worked?

To verify that you've successfully changed the properties of a restore request, run the **Get-MailboxRestoreRequestStatistics** cmdlet to display the revised properties for the restore request. If the restore request was successfully created, the *Status* property will have a value of `Queued`, `InProgress`, or `Completed`. After the restore request is completed, the contents of the soft-deleted mailbox will appear in the target mailbox.

For detailed syntax and parameter information, see Get-MailboxRestoreRequestStatistics.

Use the Shell to suspend a restore request

You can suspend a restore request any time after the request was created but before the request reaches the status of `Completed`. See Use the Shell to resume a restore request later in this topic for the command syntax to resume the restore request using the Resume-MailboxRestoreRequest cmdlet.

This example suspends the restore request MailboxRestore1 for Pilar Pinilla's mailbox.

```
Suspend-MailboxRestoreRequest -Identity "Pilar Pinilla
\MailboxRestore1"
```

This example suspends all restore requests in progress by first retrieving all requests that have a status of `InProgress`, and then piping the output to the **Suspend-MailboxRestoreRequest** cmdlet and including the suspend comment "Resume after FY13Q2 Maintenance."

```
Get-MailboxRestoreRequest -Status InProgress | Suspend-
MailboxRestoreRequest -SuspendComment "Resume after FY13Q2
Maintenance"
```

For detailed syntax and parameter information, see Suspend-MailboxRestoreRequest.

How do you know this worked?

To verify that you've successfully suspended a mailbox restore request, run the following command.

```
Get-MailboxRestoreRequest <identity> | Format-List
Suspend,Status
```

If the value of the *Suspend* property equals `True`, the restore request was successfully suspended. Also, a value of `Suspended` for the *Status* property indicates that the restore request was suspended.

Use the Shell to resume a restore request

Use the **Resume-MailboxRestoreRequest** cmdlet to resume a restore request that failed or was suspended.

This example resumes the restore request Pilar Pinilla\MailboxRestore1.

```
Resume-MailboxRestoreRequest -Identity "Pilar Pinilla
\MailboxRestore1"
```

This example resumes all restore requests that have a status of Failed.

```
Get-MailboxRestoreRequest -Status Failed | Resume-
MailboxRestoreRequest
```

For detailed syntax and parameter information, see Resume-MailboxRestoreRequest.

How do you know this worked?

To verify that a restore request has resumed, run the following command.

```
Get-MailboxRestoreRequest <identity> | Format-List
Suspend,Status
```

If the value of the *Suspend* property equals `False`, the restore request successfully resumed. Also, a value of `InProgress` for the *Status* property indicates that the restore request resumed.

Use the Shell to remove a restore request

You can use the **Remove-MailboxRestoreRequest** cmdlet to remove mailbox restore requests. If you remove a restore request after mailbox data begins being copied to the target mailbox, the mailbox data that's copied remains in the target mailbox.

Note:
As previously stated, completed restore requests are retained for 30 days by default before they're automatically deleted.

This example removes the restore request Pilar Pinilla\MailboxRestore1.

```
Remove-MailboxRestoreRequest -Identity "Pilar Pinilla
\MailboxRestore1"
```

This example removes all restore requests that have the status of Completed.

```
Get-MailboxRestoreRequest -Status Completed | Remove-
MailboxRestoreRequest
```

This example cancels the restore request by using the *RequestGuid* parameter for a request stored on MBXDB01. The parameter set that requires the *RequestGuid* and *RequestQueue* parameters is used only for Microsoft Replication Service debugging purposes. You should use this parameter set only if instructed by Microsoft Customer Service and Support.

```
Remove-MailboxRestoreRequest -RequestQueue MBXDB01 -
RequestGuid 25e0eaf2-6cc2-4353-b83e-5cb7b72d441f
```

For detailed syntax and parameter information, see Remove-MailboxRestoreRequest.

How do you know this worked?

To verify that you've successfully removed a mailbox restore request, run the following command.

```
Get-MailboxRestoreRequest -Identity <identity of removed
restore request>
```

The command will return an error stating that the restore request doesn't exist.

You can also run the **Get-MailboxRestoreRequest** cmdlet. If a restore request was successfully removed, it won't be included in the results.

Permanently delete a mailbox

Exchange Server 2013 > Recipients > Disconnected mailboxes >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-16*

When you permanently delete active mailboxes and disconnected mailboxes, all mailbox contents are purged from the Exchange mailbox database, and the data loss is permanent. When you permanently delete an active mailbox, the associated Active Directory user account is also deleted.

An alternative to permanently deleting a mailbox is to disconnect it. After you disconnect a mailbox, by default, it's retained in the mailbox database for 30 days. This gives you the opportunity to reconnect or restore a mailbox before it's purged from the database.

To learn more about disconnected mailboxes and perform other related management tasks, see the following topics:

- Disconnected mailboxes
- Disable or delete a mailbox
- Connect a disabled mailbox
- Connect or restore a deleted mailbox

Note:

You can't use the EAC to permanently delete an active mailbox or a disconnected mailbox.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Permanently delete an active mailbox

Use the Shell to permanently delete an active mailbox

Run the following command to permanently delete an active mailbox and the associated Active Directory user account.

```
Remove-Mailbox -Identity <identity> -Permanent $true
```

Note:

If you don't include the *Permanent* parameter, the deleted mailbox is retained in the mailbox database for 30 days, by default, before it's permanently deleted.

For detailed syntax and parameter information, see Remove-Mailbox.

How do you know this worked?

To verify that you've permanently deleted an active mailbox, do the following:

1. Verify that the mailbox is no longer listed in the EAC.
2. Verify that the associated user account is no longer listed in Active Directory Users and Computers.
3. Run the following command to verify that the mailbox was successfully purged from the Exchange mailbox database.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisplayName -eq "<display name>" }
```

If you successfully purged the mailbox, the command won't return any results. If the mailbox wasn't

purged, the command will return information about the mailbox.

Permanently delete a disconnected mailbox

Use the Shell to permanently delete a disconnected mailbox

There are two types of disconnected mailboxes: disabled and soft-deleted. You must specify one of these types when running the cmdlet to permanently delete the mailbox. If the type you specify doesn't match the actual type of the disconnected mailbox, the command fails.

Run the following command to determine whether a disconnected mailbox is disabled or soft-deleted.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisplayName -eq "<display name>" } | fl
DisplayName,MailboxGuid,Database,DisconnectReason
```

The value for the *DisconnectReason* property for disconnected mailboxes will be either `Disabled` or `SoftDeleted`.

You can run the following command to display the type for all disconnected mailboxes in your organization.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisconnectReason -ne $null } | fl
DisplayName,MailboxGuid,Database,DisconnectReason
```

⚠Warning:

When you use the **Remove-StoreMailbox** cmdlet to permanently delete a disconnected mailbox, all its contents are purged from the mailbox database and the data loss is permanent.

This example permanently deletes the disabled mailbox with the GUID 2ab32ce3-fae1-4402-9489-c67e3ae173d3 from mailbox database MBD01.

```
Remove-StoreMailbox -Database MBD01 -Identity "2ab32ce3-
fae1-4402-9489-c67e3ae173d3" -MailboxState Disabled
```

This example permanently deletes the soft-deleted mailbox for Dan Jump from mailbox database MBD01.

```
Remove-StoreMailbox -Database MBD01 -Identity "Dan Jump" -
MailboxState SoftDeleted
```

This example permanently deletes all soft-deleted mailboxes from mailbox database MBD01.

```
Get-MailboxStatistics -Database MBD01 | where
```

```
{$_.DisconnectReason -eq "SoftDeleted"} | ForEach {Remove-
StoreMailbox -Database $_.Database -Identity $_.MailboxGuid
-MailboxState SoftDeleted}
```

For detailed syntax and parameter information, see Remove-StoreMailbox and Get-MailboxStatistics.

How do you know this worked?

To verify that you've permanently deleted a disconnected mailbox and that it was successfully purged from the Exchange mailbox database, run the following command.

```
Get-MailboxDatabase | Get-MailboxStatistics | Where
{ $_.DisplayName -eq "<display name>" }
```

If you successfully purged the mailbox, the command won't return any results. If the mailbox wasn't purged, the command will return information about the mailbox.

Custom attributes

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

Microsoft Exchange Server 2013 includes 15 extension attributes. You can use these attributes to add information about a recipient, such as an employee ID, organizational unit (OU), or some other custom value for which there isn't an existing attribute. These custom attributes are labeled in Active Directory as **ms-Exch-Extension-Attribute1** through **ms-Exch-Extension-Attribute15**. In the Exchange Management Shell, the corresponding parameters are *CustomAttribute1* through *CustomAttribute15*. These attributes aren't used by any Exchange components. They can be used to store Active Directory data without having to extend the Active Directory schema.

In Exchange Server 2003 and earlier, if you wanted to store this information in Active Directory, you had to create an attribute by extending the Active Directory schema. Schema extension requires planning, procuring object identifiers (OIDs) for new attributes, and testing the extension process in a test environment before you implement it in a production environment. In Exchange 2013, schema extensions can't be used in recipient filters used by address lists, e-mail address policies, and dynamic distribution groups.

Contents

Advantages of custom attributes

Some of the advantages of using custom attributes include:

- You avoid extending the Active Directory schema.
- The attributes are created by Exchange Setup.
- You can use the Exchange Administration Center (EAC) or the Exchange Management Shell to manage the attributes. You don't need to build custom controls or write scripts to populate and display these attributes.
- The attributes are filterable properties that can be used in the *Filter* parameter with recipient cmdlets such as **Get-Mailbox**. They can also be used in the EAC and the Shell to create filters for e-mail address policies, address lists, and dynamic distribution groups.

Multivalued custom attributes

In Exchange 2010 Service Pack 2 (SP2), five multivalued custom attributes were added to Exchange to allow you to store additional information for mail recipients if the traditional custom attributes didn't meet your needs. The *ExtensionCustomAttribute1* to *ExtensionCustomAttribute5* parameters can hold up to 1,300 values each. You can specify multiple values as a comma-delimited list. The following cmdlets support these new parameters:

- Set-DistributionGroup
- Set-DynamicDistributionGroup
- Set-Mailbox
- Set-MailContact
- Set-MailPublicFolder
- Set-RemoteMailbox

For more information about multivalued properties, see Modifying multivalued properties.

Custom attribute examples

In many Exchange deployments, creating an e-mail address policy for all recipients in an OU is a common scenario. The OU isn't a filterable property that can be used in the *RecipientFilter* parameter of an e-mail address policy or an address list.

Note:
Dynamic distribution groups have an additional parameter that you can use to restrict it to recipients in a particular OU or container.

If the recipients in that OU don't share any common properties that you can filter by, such as

department or location, you can populate one of the custom attributes with a common value, as shown in this example.

```
Get-Mailbox -OrganizationalUnit Sales | Set-Mailbox
CustomAttribute1 "SalesOU"
```

Now you can create an e-mail address policy for all recipients that have the *CustomAttribute1* property that equals SalesOU, as shown in this example.

```
New-EmailAddressPolicy -Name "Sales" -RecipientFilter
{ CustomAttribute1 -eq "SalesOU"} -
EnabledEmailAddressTemplates "SMTP:%s%2g@sales.contoso.com"
```

Custom attribute example using the ConditionalCustomAttributes parameter

When creating dynamic distribution groups, email address policies, or address lists, you don't need to use the *RecipeintFilter* parameter to specify custom attributes. You can use the *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters instead.

This example creates a dynamic distribution group based on the recipients whose *CustomAttribute1* is set to SalesOU.

```
New-DynamicDistributionGroup -Name "Sales Users and
Contacts" -IncludedRecipients "MailboxUsers,MailContacts" -
ConditionalCustomAttribute1 "SalesOU"
```

Note:

You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. In addition, you can't use *Conditional* parameters if you use the *RecipientFilter* parameter. If you want to include additional filters to create your dynamic distribution group, email address policies, or address lists, you should use the *RecipientFilter* parameter.

Custom attribute example using ExtensionCustomAttributes parameter

In this example, the mailbox for Kweku will have *ExtensionCustomAttribute1* updated to reflect that he's enrolled in the following educational classes: MATH307, ECON202, and ENGL300.

```
Set-Mailbox -Identity Kweku -ExtensionCustomAttribute1
MATH307,ECON202,ENGL300
```

Next, a dynamic distribution group for all students enrolled MATH307 is created by using the *RecipientFilter* parameter where *ExtensionCustomAttribute1* is equal to MATH307. When using the *ExtentionCustomAttributes* parameters, you can use the `-eq` operator instead of the `-like` operator.

```
New-DynamicDistributionGroup -Name Students_MATH307 -
RecipientFilter {ExtensionCustomAttribute1 -eq "MATH307"}
```

In this example, Kweku's *ExtensionCustomAttribute1* values are updated to reflect that he's added the class ENGL210 and removed the class ECON202.

```
Set-Mailbox -Identity Kweku -ExtensionCustomAttribute1
@{Add="ENGL210"; Remove="ECON202"}
```

Filters in recipient Shell commands

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-05*

You can use several Exchange Management Shell commands to filter a set of recipients. You can create the following types of filters in an Exchange command:

- Precanned filters
- Custom filters using the *RecipientFilter* parameter
- Custom filters using the *Filter* parameter
- Custom filters using the *ContentFilter* parameter

In Microsoft Exchange Server 2003 and earlier versions, LDAP filtering syntax is used to create custom address lists, global address lists (GALs), email address policies, and distribution groups. In later versions of Exchange, the OPATH filtering syntax replaces the LDAP filtering syntax.

Contents

Precanned filters

A *precanned filter* is a commonly used Exchange filter that you can use to meet a variety of recipient-filtering criteria for creating dynamic distribution groups, email address policies, address lists, or GALs. With precanned filters, you can use either the Exchange Management Shell or the Exchange Administration Center (EAC). Using precanned filters, you can do the following:

- Determine the scope of recipients.
- Add conditional filtering based on properties such as company, department, and state or region.
- Add custom attributes for recipients. For more information, see Custom attributes.

The following parameters are considered precanned filters:

- *IncludedRecipients*
- *ConditionalCompany*
- *ConditionalDepartment*
- *ConditionalStateOrProvince*
- *ConditionalCustomAttribute1*–15.

Precanned filters are available for the following cmdlets:

- New-DynamicDistributionGroup
- Set-DynamicDistributionGroup
- New-EmailAddressPolicy
- Set-EmailAddressPolicy
- New-AddressList
- Set-AddressList
- New-GlobalAddressList
- Set-GlobalAddressList

Example

This example describes using precanned filters in the Shell to create a dynamic distribution group. The syntax in this example is similar but not identical to the syntax you would use to create an email address policy, address list, or GAL. When creating a precanned filter, you should ask the following questions:

- From which organizational unit (OU) do you want to include recipients? (This question corresponds to the *RecipientContainer* parameter.)

Note:

Selecting the OU for this purpose applies only when creating dynamic distribution groups, and not when creating email address policies, address lists, or GALs.

- What type of recipients do you want to include? (This question corresponds to the *IncludedRecipients* parameter.)
- What additional conditions do you want to include in the filter? (This question corresponds to the *ConditionalCompany*, *ConditionalDepartment*, *ConditionalStateOrProvince*, and

ConditionalCustomAttribute parameters.)

This example creates the dynamic distribution group Contoso Finance for user mailboxes in the OU Contoso.com/Users and specifies the condition to include only recipients who have the **Department** attribute defined as Finance and the **Company** attribute defined as Contoso.

```
New-DynamicDistributionGroup -Name "Contoso Finance" -
OrganizationalUnit Contoso.com/Users -RecipientContainer
Contoso.com/Users -IncludedRecipients MailboxUsers -
ConditionalDepartment "Finance" -ConditionalCompany
"Contoso"
```

This example displays the properties of this new dynamic distribution group.

```
Get-DynamicDistributionGroup -Identity "Contoso Finance" |
Format-List Recipient*,Included*
```

Return to top

Custom filters using the RecipientFilter parameter

If precanned filters don't meet your needs for creating or modifying dynamic distribution groups, email address policies, and address lists, you can create a custom filter by using the *RecipientFilter* parameter.

The recipient filter parameter is available for the following cmdlets:

- New-DynamicDistributionGroup
- Set-DynamicDistributionGroup
- New-EmailAddressPolicy
- Set-EmailAddressPolicy
- New-AddressList
- Set-AddressList
- New-GlobalAddressList
- Set-GlobalAddressList

For more information about the filterable properties you can use with the *RecipientFilter* parameter, see Filterable properties for the -RecipientFilter parameter.

Example

The following example uses the *RecipientFilter* parameter to create a dynamic distribution group. The syntax in this example is similar but not identical to the syntax you use to create an email address policy, address list, or GAL.

This example uses custom filters to create a dynamic distribution group for user mailboxes that

have the **Company** attribute defined as Contoso and the **Office** attribute defined as North Building.

```
New-DynamicDistributionGroup -Name AllContosoNorth -
OrganizationalUnit contoso.com/Users -RecipientFilter
{ ((RecipientType -eq 'UserMailbox') -and (Company -eq
'Contoso') -and (Office -eq 'North Building')) }
```

Return to top

Custom filters using the Filter parameter

You can use the *Filter* parameter to filter the results of a command to specify which objects to retrieve. For example, instead of retrieving all users or groups, you can specify a set of users or groups by using a filter string. This type of filter doesn't modify any configuration or attributes of objects. It only modifies the set of objects that the command returns.

Using the *Filter* parameter to modify command results is known as *server-side filtering*. Server-side filtering submits the command and the filter to the server for processing. The Shell also supports client-side filtering, in which the command retrieves all objects from the server and then applies the filter in the local console window. To perform client-side filtering, use the **Where-Object** cmdlet. For more information about server-side and client-side filtering, see "How to Filter Data" in Working with command output.

To find the filterable properties for cmdlets that have the *Filter* parameter, you can run the **Get** command against an object and format the output by pipelining the **Format-List** parameter. Most of the returned values will be available for use in the *Filter* parameter. The following example returns a detailed list for the mailbox Ayla.

```
Get-Mailbox -Identity Ayla | Format-List
```

The *Filter* parameter is available for the following cmdlets:

- Get-ActiveSyncDevice
- Get-ActiveSyncDeviceClass
- Get-CASMailbox
- Get-Contact
- Get-DistributionGroup
- Get-DynamicDistributionGroup
- Get-Group
- Get-Mailbox
- Get-MailboxStatistics
- Get-MailContact
- Get-MailPublicFolder
- Get-MailUser
- Get-Message

- Get-MobileDevice
- Get-Queue
- Get-QueueDigest
- Get-Recipient
- Get-RemoteMailbox
- Get-RoleGroup
- Get-SecurityPrincipal
- Get-StoreUsageStatistics
- Get-ThrottlingPolicyAssociation
- Get-UMMailbox
- Get-User
- Get-UserPhoto
- Remove-Message
- Resume-Message
- Resume-Queue
- Retry-Queue
- Suspend-Message
- Suspend-Queue

For more information about the filterable properties you can use with the *Filter* parameter, see Filterable properties for the -Filter parameter.

Example

This example uses the *Filter* parameter to return information about users whose title contains the word "manager".

```
Get-User -Filter {Title -like 'Manager*'}
```

Return to top

Custom filters using the ContentFilter parameter

You can use the *ContentFilter* parameter to select specific message content to export when using the New-MailboxExportRequest cmdlet. If the command finds a message that contains the match to the content filter, it exports the message to a .pst file.

Example

This example creates an export request that searches Ayla's mailbox for messages where the body contains the phrase "company prospectus". If that phrase is found, the command exports all messages with that phrase to a .pst file.

```
New-MailboxExportRequest -Mailbox Ayla -ContentFilter {Body
-like "company prospectus*"}
```

For more information about the filterable properties you can use with the *ContentFilter* parameter, see Filterable properties for the -ContentFilter parameter.

Return to top

Additional OPATH syntax information

When creating your own custom filters, be aware of the following:

- Use braces { } around the entire OPATH syntax string with the *Filter* or *RecipientFilter* parameter.
- Include the hyphen before all operators. The most common operators include:
  - **-and**
  - **-or**
  - **-not**
  - **-eq** (equals)
  - **-ne** (not equal)
  - **-lt** (less than)
  - **-gt** (greater than)
  - **-like** (string comparison)
  - **-notlike** (string comparison)
- Many of the properties for the *RecipientFilter* and *Filter* parameters accept wildcard characters. If you use a wildcard character, use the **like** operator instead of the **eq** operator. The **like** operator is used to find pattern matches in rich types, such as strings, whereas the **eq** operator is used to find an exact match.
- Run the following commands to get information about operators you can use:
  - `Help about_logical_operator`
  - `Help about_comparison_operator`
- You can use most properties of recipient types to create filter strings. For information about filterable properties you can use with a specific cmdlet, see the cmdlet reference topics in Exchange Management Shell.

Recipient filter documentation

The following table contains links to topics that will help you learn more about the filterable properties that you can use with Exchange recipient commands.

Topic	Description
Filterable properties for the -RecipientFilter parameter	Learn more about the filterable properties for the *RecipientFilter* parameter.

Filterable properties for the -Filter parameter	Learn more about the filterable properties for the *Filter* parameter.
Filterable properties for the -ContentFilter parameter	Learn more about using the *ContentFilter* parameter when using the **New-MailboxExportRequest** cmdlet.

Return to top

Filterable properties for the -RecipientFilter parameter

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-05*

This topic lists the filterable properties for the *-RecipientFilter* parameter in Microsoft Exchange 2013. The *-RecipientFilter* parameter allows you to define the criteria to create custom dynamic distribution groups, email address policies, address lists, or global address lists (GALs).

The following two tables in this topic outline the common filterable properties and the advanced filterable properties. Other properties, including any customer schema extensions, cannot be used in the *-RecipientFilter* parameter.

The *-RecipientFilter* parameter is used in the following cmdlets:

- New-DynamicDistributionGroup
- Set-DynamicDistributionGroup
- New-EmailAddressPolicy
- Set-EmailAddressPolicy
- New-AddressList
- Set-AddressList
- New-GlobalAddressList
- Set-GlobalAddressList

In many instances, a filterable property accepts the distinguished name (DN) or the globally unique identifier (GUID) of an object instead of the object's name. To locate the DN or the GUID of the object that you want to use in a recipient filter, use the corresponding **Get-** cmdlet.

For example, to locate the DN or GUID for a Microsoft Exchange ActiveSync mailbox policy named NewActiveSyncPolicy, run the following command:

```
Get-ActiveSyncMailboxPolicy -Identity NewActiveSyncPolicy |
format-list DistinguishedName,GUID
```

You can then copy the DN or GUID and paste it into your recipient filter. For example, to use the DN to create a dynamic distribution group based on the Exchange ActiveSync mailbox policy, run the following command:

```
New-DynamicDistributionGroup -Name NewDDG -RecipientFilter
{(ActiveSyncMailboxPolicy -eq 'CN=Default,CN=Mobile Mailbox
Policies,CN=First Organization,CN=Microsoft
Exchange,CN=Services,CN=Configuration,DC=IELRVS-
dom,DC=extest,DC=Contoso,DC=com')}
```

Common filterable properties

The following table contains the common filterable properties for the *-RecipientFilter* parameter. This table lists the name of the property, the Lightweight Directory Access Protocol (LDAP) display name, a description of the property, and the possible values that the property can take. You can use the LDAP display name to convert your Microsoft Exchange Server 2003 or earlier LDAP filters into Exchange Server 2007 OPath filters. For more information about converting LDAP filters to Exchange OPath filters, see the Microsoft Exchange Team Blog article, Need help converting your LDAP filters to OPATH.

Note:
The content of each blog and its URL are subject to change without notice. The content within each blog is provided "AS IS" with no warranties, and confers no rights. Use of included script samples or code is subject to the terms specified in the Microsoft Terms of Use.

Note:
Not all filterable properties have an LDAP display name. The properties that do not have an LDAP display name are not Active Directory directory service properties. They are properties that are calculated by Exchange.

Note:
Many of the properties for the *-RecipientFilter* parameter accept wildcard characters. If you use a wildcard character, use the *-like* operator instead of the *-eq* operator. The *-like* operator is used to find pattern matches in rich types, such as strings, whereas the *-eq* operator is used to find an exact match.

Property name	LDAP display name	Description	Value
ActiveSyncMailboxPolicy	*msExchMobileMailboxPolicyLink*	This property contains the name of the Exchange ActiveSync	DN

		mailbox policy for the mailbox.	
Alias	*mailNickname*	This property contains the alias (the generic name used to identify a mail account) of the recipient.	• String • Wildcard character accepted
AssistantName	*msExchAssistantName*	This property contains the assistant name of the recipient.	• String • Wildcard character accepted
C	*C*	This property contains the two-letter country/region designation from International Organization for Standardization (ISO) 3166.	• ISO 3166 **Note:** For a complete list of the ISO 3166 standard values, see International Organization for Standardization. **Note:** The third-party Web site information in this topic is provided to help you find the technical information you need. The URLs are subject to change without notice.
City	*l*	This property contains the city name of the recipient.	• String • Wildcard character accepted
Co	*Co*	This property contains the country/region name in which the recipient resides.	• Country/Region • You can locate valid *Co* values on the **Address and Phone** tab in the recipient's properties.
Company	*company*	This property contains	• String

		the company of the recipient.	• Wildcard character accepted
CountryCode	*CountryCode*	This property contains the numeric country/ region designation from ISO 3166.	• ISO 3166 **Note:** For a complete list of the ISO 3166 standard values, see International Organization for Standardization. **Note:** The third-party Web site information in this topic is provided to help you find the technical information you need. The URLs are subject to change without notice.
CountryOrRegion	Not applicable	This property contains the country or region in which the recipient resides.	• Country/Region • You can locate valid *CountryOrRegion* values on the **Address and Phone** tab in the recipient's properties.
CustomAttribute1	*extensionAttribute1*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute2	*extensionAttribute2*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute3	*extensionAttribute3*	This property contains a custom attribute that you can add to a	• String • Wildcard character accepted

		recipient.	
CustomAttribute4	*extensionAttribute4*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute5	*extensionAttribute5*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute6	*extensionAttribute6*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute7	*extensionAttribute7*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute8	*extensionAttribute8*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute9	*extensionAttribute9*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute10	*extensionAttribute10*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted

CustomAttribute11	*extensionAttribute11*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute12	*extensionAttribute12*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute13	*extensionAttribute13*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute14	*extensionAttribute14*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
CustomAttribute15	*extensionAttribute15*	This property contains a custom attribute that you can add to a recipient.	• String • Wildcard character accepted
Database	*homeMDB*	This property contains the mailbox database.	• Mailbox database • Identity • DN
Department	*department*	This property contains the department of the recipient.	• String • Wildcard character accepted
DisplayName	*displayName*	This property contains the ambiguous name resolution (ANR) search for the display	• String • Wildcard character accepted

		name of the recipient.	
EmailAddresses	*proxyAddresses*	This property contains the email addresses of this recipient. All Exchange 2007 email address types are valid. Separate multiple values by using commas.	• Email address • Wildcard character accepted
EmailAddressPolicyEnabled	Not applicable	This property contains the control for applying email address policies to this recipient.	• Boolean • $true or $false
ExternalEmailAddress	*targetAddress*	This property contains the external email address. Email messages sent to the mail-enabled user are sent to this external address.	• Email address • Wildcard character accepted
ExternalOofOptions	*msExchExternalOOFOptions*	This property contains the option for sending an out-of-office message to external senders.	• InternalOnly • External
FirstName	*givenName*	This property contains the ANR search for the first name of the recipient.	• String • Wildcard character accepted

HiddenFromAddressListsEnabled	*msExchHideFromAddressLists*	This property specifies whether the recipient is visible in the address list.	• Boolean • $true or $false
IsLinked	Not applicable	This property specifies whether a folder is linked to a master folder.	• Boolean • $true or $false
IsMailboxEnabled	Not applicable	This property specifies whether a mailbox is mailbox-enabled.	• Boolean • $true or $false
IsResource	Not applicable	This property specifies whether a mailbox is a resource mailbox.	• Boolean • $true or $false
IsShared	Not applicable	This property specifies whether a resource mailbox is shared.	• Boolean • $true or $false
IssueWarningQuota	*mDBStorageQuota*	This property contains the mailbox size at which a warning message is sent to the user.	Unlimited or integer
LanguagesRaw	*msExchUserCulture*	This property contains the language preference for this mailbox.	• ISO 639-ISO 3166 **Note:** An acceptable value for this parameter is a combination of an ISO 639 two-letter lowercase culture code that is associated with a language and an ISO 3166 two-letter uppercase subculture

			code that is associated with a country or region. For example, United States English is represented as `en-us`. To learn more about culture codes and for a full list of acceptable values, see CultureInfo Class in the MSDN Library.
LastName	*sn*	This property contains the ANR search for the last name of the recipient.	• String • Wildcard character accepted
ManagedFolderMailboxPolicy	*msExchMailboxTemplateLink*	This property contains the managed folder mailbox policy that controls messaging records management (MRM).	• Name • GUID • DN
Manager	*manager*	This property contains the manager of the recipient.	• String • Wildcard character accepted
MaxReceiveSize	*delivContLength*	This property contains the maximum size of messages that this recipient can receive.	Unlimited or integer
MaxSendSize	*submissionContLength*	This property contains the maximum size of messages that this recipient can send.	Unlimited or integer

MobileFeaturesEnabled	*msExchOmaAdminWirelessEnable*	This property specifies whether a mailbox has mobile features enabled.	• Boolean • $true or $false
Name	*name*	This property contains the name of the recipient.	String
Office	*physicalDeliveryOfficeName*	This property contains the office of the recipient.	String
OfflineAddressBook	*msExchUseOAB*	This property contains the offline address book (OAB) that is associated with this recipient.	• Name • GUID • DN
PostalCode	*postalCode*	This property contains the postal code of the recipient.	String
ProhibitSendQuota	*mDBOverQuotaLimit*	This property contains the mailbox size at which the user associated with this mailbox can no longer send messages.	Unlimited or integer
ProhibitSendReceiveQuota	*mDBOverHardQuotaLimit*	This property contains the mailbox size at which the user associated with this mailbox can no longer send or receive	Unlimited or integer

		messages	
RecipientLimits	*msExchRecipLimit*	This property contains the maximum number of recipients per message to which this mailbox can send.	Unlimited or integer
RecipientType	Not applicable	This property specifies the recipient type.	• UserMailbox • MailUser • MailContact • MailUniversalDistributionGroup • MailUniversalSecurityGroup • MailNonUniversalGroup • DynamicDistributionGroup • PublicFolder
RecipientTypeDetails	Not applicable	This property specifies the recipient subtype.	• ConferenceRoomMailbox • EquipmentMailbox • LegacyMailbox • LinkedMailbox • UserMailbox • MailContact • DynamicDistributionGroup • MailForestContact • MailNonUniversalGroup • MailUniversalDistributionGroup • MailUniversalSecurityGroup

			• MailUser • PublicFolder • SharedMailbox
SamAccountName	*SamAccountName*	This property contains the logon name that is used to support client computers and servers running older versions of Microsoft Windows operating systems.	• String • Wildcard character accepted
ServerName	Not applicable	This property contains the name of the server where the recipient resides.	Server name
StateOrProvince	*st*	This property contains the state or province information that is defined for this recipient.	• String • Wildcard character accepted
StreetAddress	*streetAddress*	This property contains the street address that is defined for this recipient.	String
Title	*title*	This property contains the title of the recipient.	String
UMEnabled	Not applicable	This property specifies whether Unified Messaging (UM) is	• Boolean • $true or $false

		enabled for this mailbox.	
UseDatabaseQuotaDefaults	*mDBUseDefaults*	This property specifies whether the mailbox uses the quota attributes for the mailbox database in which this mailbox resides. The quota attributes are: ProhibitSendQuota, ProhibitSendReceiveQuota, IssueWarningQuota, and RulesQuota.	• ProhibitSendQuota • ProhibitSendReceiveQuota • IssueWarningQuota • RulesQuota
UserPrincipalName	*userPrincipalName*	This property contains the user principal name (UPN) for this recipient. The UPN is the logon name for the user and consists of a user name and a suffix. Typically, the suffix is the domain name where the user account resides. For example, `kim@contoso.com`.	• User logon name • User principal name • Wildcard character accepted

Advanced filterable properties

The following table lists filterable properties that are not commonly used. They are included in this topic for reference purposes.

Name	LDAP name
AcceptMessagesOnlyFrom	*authOrig*
AcceptMessagesOnlyFromDLMembers	*dLMemSubmitPerms*
ActiveSyncAllowedDeviceIDs	*msExchMobileAllowedDeviceIds*
ActiveSyncDebugLogging	*msExchMobileDebugLogging*
ActiveSyncEnabled	Not applicable
AddressListMembership	*showInAddressBook*
AllowUMCallsFromNonUsers	*msExchUMListInDirectorySearch*
CallAnsweringAudioCodec	*msExchUMAudioCodec*
Certificate	*userCertificate*
CommonName	*cn*
DeletedItemFlags	*deletedItemFlags*
DeliverToMailboxAndForward	*deliverAndRedirect*
Description	*description*
DistinguishedName	*distinguishedName*
ElcExpirationSuspensionEndDate	*msExchELCExpirySuspensionEnd*
ElcExpirationSuspensionStartDate	*msExchELCExpirySuspensionStart*
ElcMailboxFlags	*msExchELCMailboxFlags*
ExchangeGuid	*msExchMailboxGuid*
ExchangeSecurityDescriptor	*msExchMailboxSecurityDescriptor*
ExchangeVersion	*msExchVersion*
ExpansionServer	*msExchExpansionServerName*

ExternalOofOptions	*msExchExternalOOFOptions*
Fax	*facsimileTelephoneNumber*
ForwardingAddress	*altRecipient*
GrantSendOnBehalfTo	*publicDelegates*
GroupType	*groupType*
Guid	*objectGuid*
HasActiveSyncDevicePartnership	Not applicable
HiddenGroupMembershipEnabled	*hideDLMembership*
HomeMTA	*homeMTA*
HomePhone	*homePhone*
Id	*distinguishedName*
ImapEnabled	Not applicable
IncludedRecipients	Not applicable
Initials	*initials*
InternetEncoding	*internetEncoding*
LdapRecipientFilter	*msExchDynamicDLFilter*
LegacyExchangeDN	*legacyExchangeDN*
LinkedMasterAccount	Not applicable
LocaleID	*localeID*
MailboxFolderSet	*msExchMailboxFolderSet*
ManagedBy	*managedBy*
MAPIEnabled	Not applicable

MapiRecipient	*mAPIRecipient*
MasterAccountSid	*msExchMasterAccountSid*
MaxBlockedSenders	*msExchMaxBlockedSenders*
MaxSafeSenders	*msExchMaxSafeSenders*
MemberOfGroup	*memberOf*
Members	*member*
MessageHygieneFlags	*msExchMessageHygieneFlags*
MobileAdminExtendedSettings	*msExchOmaAdminExtendedSettings*
MobileMailboxFlags	*msExchMobileMailboxFlags*
MobilePhone	*mobile*
msRTCSIP-Line	*msRTCSIP-Line*
Name	*LDAP Display Name*
Notes	*info*
NTSecurityDescriptor	*ntSecurityDescriptor*
ObjectCategory	*objectCategory*
ObjectClass	*objectClass*
ObjectState	Not applicable
OperatorNumber	*msExchUMOperatorNumber*
OriginalId	Not applicable
OriginalPrimarySmtpAddress	Not applicable
OriginalWindowsEmailAddress	Not applicable
OtherFax	*otherFacsimileTelephoneNumber*

OtherHomePhone	*otherHomePhone*
OtherTelephone	*otherTelephone*
OWAActiveSyncIntegrationEnabled	Not applicable
OWAAllAddressListsEnabled	Not applicable
OWACalendarEnabled	Not applicable
OWAChangePasswordEnabled	Not applicable
OWAContactsEnabled	Not applicable
OWAEnabled	Not applicable
OWAJournalEnabled	Not applicable
OWANotesEnabled	Not applicable
OWAPremiumClientEnabled	Not applicable
OWAPublicFoldersEnabled	Not applicable
OWARecoverDeletedItemsEnabled	Not applicable
OWARemindersAndNotificationsEnabled	Not applicable
OWARulesEnabled	Not applicable
OWASearchFoldersEnabled	Not applicable
OWASignaturesEnabled	Not applicable
OWASMimeEnabled	Not applicable
OWASpellCheckerEnabled	Not applicable
OWATasksEnabled	Not applicable
OWAThemeSelectionEnabled	Not applicable
OWAUMIntegrationEnabled	Not applicable

OWAUNCAccessonPrivateComputersEnabled	Not applicable
OWAUNCAccessonPublicComputersEnabled	Not applicable
OWAWSSAccessOnPrivateComputersEnabled	Not applicable
OWAWSSAccessOnPublicComputersEnabled	Not applicable
Pager	*pager*
Phone	*telephoneNumber*
PhoneticCompany	*msDS-PhoneticCompanyName*
PhoneticDepartment	*msDS-PhoneticDepartment*
PhoneticDisplayName	*msDS-PhoneticDisplayName*
PhoneticFirstName	*msDS-PhoneticFirstName*
PhoneticLastName	*msDS-PhoneticLastName*
PoliciesExcluded	*msExchPoliciesExcluded*
PoliciesIncluded	*msExchPoliciesIncluded*
PopEnabled	Not applicable
PostOfficeBox	*postOfficeBox*
PrimaryGroupId	*primaryGroupId*
PrimarySmtpAddress	Not applicable
ProtocolSettings	*protocolSettings*
PublicFolderContacts	*pFContacts*
PublicFolderRootUrl	*msExchPfRootUrl*
PublicFolderType	*msExchPFTreeType*
PurportedSearchUI	*msExchPurportedSearchUI*

QueryBaseDN	*msExchQueryBaseDN*
RawCanonicalName	*canonicalName*
RawExternalEmailAddress	*targetAddress*
RawName	*name*
RecipientContainer	*msExchDynamicDLBaseDN*
RecipientDisplayType	*msExchRecipientDisplayType*
RecipientFilter	*msExchQueryFilter*
RejectMessagesFrom	*unauthOrig*
RejectMessagesFromDLMembers	*dLMemRejectPerms*
ReportToManagerEnabled	*reportToOwner*
ReportToOriginatorEnabled	*reportToOriginator*
RequireAllSendersAreAuthenticated	*msExchRequireAuthToSendTo*
ResourceCapacity	*msExchResourceCapacity*
ResourceCustom	Not applicable
ResourceMetaData	*msExchResourceMetaData*
ResourcePropertiesDisplay	*msExchResourceDisplay*
ResourceSearchProperties	*msExchResourceSearchProperties*
ResourceType	Not applicable
RetainDeletedItemsFor	*garbageCollPeriod*
RTCSIPPrimaryUserAddress	*msRTCSIP-PrimaryUserAddress*
RulesQuota	*msExchMDBRulesQuota*
SafeRecipientsHash	*msExchSafeRecipientsHash*

SafeSendersHash	*msExchSafeSendersHash*
SCLDeleteThresholdInt	*msExchMessageHygieneSCLDeleteThreshold*
SCLJunkThresholdInt	*msExchMessageHygieneSCLJunkThreshold*
SCLQuarantineThresholdInt	*msExchMessageHygieneSCLQuarantineThreshold*
SCLRejectThresholdInt	*msExchMessageHygieneSCLRejectThreshold*
SecurityProtocol	*securityProtocol*
SendDeliveryReportsTo	Not applicable
SendOofMessageToOriginatorEnabled	*oOFReplyToOriginator*
ServerLegacyDN	*msExchHomeServerName*
Sid	*objectSid*
SidHistory	*sIDHistory*
SimpleDisplayName	*displayNamePrintable*
SMimeCertificate	*userSMIMECertificate*
TelephoneAssistant	*telephoneAssistant*
TextEncodedORAddress	*textEncodedORAddress*
UMDtmfMap	*msExchUMDtmfMap*
UMEnabledFlags	*msExchUMEnabledFlags*
UMMailboxPolicy	*msExchUMTemplateLink*
UMPinChecksum	*msExchUMPinChecksum*
UMRecipientDialPlanId	*msExchUMRecipientDialPlanLink*
UMServerWritableFlags	*msExchUMServerWritableFlags*

UMSpokenName	*msExchUMSpokenName*
UnicodePassword	*unicodePwd*
UserAccountControl	*userAccountControl*
ViewDepth	Not applicable
WebPage	*wWWHomePage*

For more information

For more information about the syntax that can be used within Opath filters, see Syntax.

For more information about creating filters in recipient commands, see Filters in recipient Shell commands.

For detailed syntax and parameter information, see the following reference topics:

- New-DynamicDistributionGroup
- Set-DynamicDistributionGroup
- New-EmailAddressPolicy
- Set-EmailAddressPolicy
- New-AddressList
- Set-AddressList
- New-GlobalAddressList
- Set-GlobalAddressList

Filterable properties for the -Filter parameter

Exchange Server 2013 > Recipients > Filters in recipient Shell commands >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-07*

This topic lists the filterable properties for the *-Filter* parameter in Microsoft Exchange 2013.

Use the *-Filter* parameter to return the objects from a **Get-** command in a filtered list. For example, if you want the **Get-Contact** cmdlet to return a subset of all of the contacts in your Microsoft Exchange organization, you would use the *-Filter* parameter. The subset of objects that is returned is based on specific properties.

The *-Filter* parameter is used in the following cmdlets:

- Get-CASMailbox
- Get-Contact
- Get-DistributionGroup
- Get-DynamicDistributionGroup
- Get-Group
- Get-Mailbox
- Get-MailContact
- Get-MailboxStatistics
- Get-MailPublicFolder
- Get-MailUser
- Get-Recipient
- Get-RemoteMailbox
- Get-UMMailbox
- Get-User

Common filterable properties

The following table contains the common filterable properties for the *-Filter* parameter. This table lists the name of the property, the Lightweight Directory Access Protocol (LDAP) display name, a description of the property, the possible values that the property can take, and each of the cmdlets that accept this property for the *-Filter* parameter. You can use the LDAP display name to convert your Exchange Server 2003 or earlier LDAP filters into Exchange Opath filters. For more information about converting LDAP filters to Opath filters, see the Microsoft Exchange Team Blog article, Need help converting your LDAP filters to OPATH.

Note:

The content of each blog and its URL are subject to change without notice. The content within each blog is provided "AS IS" with no warranties, and confers no rights. Use of included script samples or code is subject to the terms specified in the Microsoft Terms of Use.

Note:

Not all filterable properties have an LDAP display name. The properties that do not have an LDAP display name are not Active Directory directory service properties. They are properties that are calculated by Exchange.

Note:

Many of the properties for the *-Filter* parameter accept wildcard characters. If you use a wildcard character, use the *-like* operator instead of the *-eq* operator. The *-like* operator is used to find pattern matches in rich types, such as strings, whereas the *-eq* operator is used to find an exact match.

Property name	LDAP display name	Description	Value	Cmdlets that accept this property

AcceptMessagesOnlyFrom	*authOrig*	This property contains the mailbox users and mail-enabled contacts that can send email messages to this distribution group.	• Distinguished name (DN) • Canonical name • Globally unique identifier (GUID) • Name • Display name • Alias • Exchange DN • Primary Simple Mail Transfer Protocol (SMTP) address	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox
AcceptMessagesOnlyFromDLMembers	*dLMemSubmitPerms*	This property contains the distribution groups that are allowed to send email messages to this distribution group.	• DN • Canonical name • GUID • Name • Display name • Alias • Exchange DN • Primary SMTP address	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox
ActiveSyncAllowedDeviceIDs	*msExchMobileAllowedDeviceIds*	This property contains a list of device IDs that are allowed to synchronize with	Device IDs	Get-CASMailbox

		the mailbox.		
ActiveSyncDebugLogging	*msExchMobileDebugLogging*	This property specifies whether error logging is enabled for mobile devices.	• String • Wildcard character accepted	Get-CASMailbox
ActiveSyncEnabled	Not applicable	This property specifies whether Exchange ActiveSync is enabled for the mailbox	• Boolean • $true or $false	Get-CASMailbox
ActiveSyncMailboxPolicy	*msExchMobileMailboxPolicyLink*	This property contains the name of the Exchange ActiveSync mailbox policy for the mailbox.	• String • Wildcard character accepted	Get-CASMailbox Get-Recipient
AddressListMembership	*showInAddressBook*	This property contains the address lists and global address list (GAL) to which the recipient is a member.	DN	Get-Recipient
Alias	*mailNickname*	This property contains the alias (the generic name used to identify a	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributi

		mail account) of the recipient. The alias can be a combination of characters separated by a period with no intervening spaces.		onGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
AllowUMCallsFromNonUsers	*msExchUMListInDirectorySearch*	This property specifies whether to exclude the mailbox from directory searches.	• 0 or None • 1 or SearchEnabled	Get-Contact Get-UMMailbox Get-User
AssistantName	*msExchAssistantName*	This property contains the assistant name of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-User
CallAnsweringAudioCodec	*msExchUMAudioCodec*	This property contains the audio codec to use for call answering messages. The default for the user's mailbox is the audio codec that is configured	• 0 or G711 • 1 or Wma • 2 or Gsm	Get-UMMailbox

		on the Unified Messaging (UM) dial plan.		
City	*L*	This property contains the city name of the recipient.	• String • Wildcard character accepted	Get-Contact Get-Recipient Get-User
Company	*Company*	This property contains the company name of the recipient.	• String • Wildcard character accepted	Get-Contact Get-Recipient Get-User
CountryOrRegion	Not applicable	This property contains the country or region in which the recipient resides.	• Country/Region • You can locate valid *CountryOrRegion* values on the **Address and Phone** tab in the recipient's properties.	Get-Contact Get-Recipient Get-User
CustomAttribute1	*extensionAttribute1*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser

		distribution group.		Get-Recipient Get-RemoteMailbox
CustomAttribute2	*extensionAttribute2*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute3	*extensionAttribute3*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-

				RemoteMailbox
CustomAttribute4	*extensionAttribute4*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute5	*extensionAttribute5*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute6	*extensionAttribut*	This property	• String	Get-

	e6	contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.	• Wildcard character accepted	DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute7	*extensionAttribute7*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute8	*extensionAttribute8*	This property contains a custom attribute that you	• String • Wildcard character accepted	Get-DistributionGroup Get-

		can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.		DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute9	*extensionAttribute9*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute10	*extensionAttribute10*	This property contains a custom attribute that you can add to a mailbox, mail	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup

		contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.		Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute11	*extensionAttribute11*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute12	*extensionAttribute12*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder,	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact

		dynamic distribution group, or distribution group.		Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute13	*extensionAttribute13*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute14	*extensionAttribute14*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder

		group, or distribution group.		Get-MailUser Get-Recipient Get-RemoteMailbox
CustomAttribute15	*extensionAttribute15*	This property contains a custom attribute that you can add to a mailbox, mail contact, mail user, mail-enabled public folder, dynamic distribution group, or distribution group.	• String • Wildcard character accepted	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
Database	*homeMDB*	This property contains the mailbox database.	DN	Get-Mailbox Get-Recipient
Department	*department*	This property contains the department of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-Recipient Get-User
DisplayName	*displayName*	This property contains the ambiguous name resolution (ANR)	• String • Wildcard character accepted	Get-CASMailbox Get-Contact Get-DistributionGroup

		search for the display name of the recipient.		Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailboxStatistics Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User
DistinguishedName	*distinguishedName*	This property contains the DN of the recipients for which you are filtering.	DN	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder

				Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User
EmailAddresses	*proxyAddresses*	This property contains the email addresses of this recipient. All Exchange 2007 email address types are valid. Separate multiple values with commas.	• Email address • Wildcard character accepted	Get-CASMailbox Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox
EmailAddressPolicyEnabled	Not applicable	This property contains the control for applying email address policies to this recipient.	• Boolean • $true or $false	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact

				Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
ExchangeGuid	*msExchMailboxGuid*	This property contains the GUID for the identity of the mailbox.	GUID	Get-Mailbox
ExchangeVersion	*msExchVersion*	This property contains the earliest version of Exchange that you can use to manage the returned object.	String	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User

ExpansionServer	*msExchExpansionServerName*	This property contains the name of the Exchange server on which to expand the distribution group or dynamic distribution group.	Server name	Get-DistributionGroup Get-DynamicDistributionGroup Get-Recipient
ExternalEmailAddress	*targetAddress*	This property contains the external email address. Email messages sent to the mail-enabled user are sent to this external address.	• Email address • Wildcard character accepted	Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox
ExternalOofOptions	*msExchExternalOOFOptions*	This property contains the option for sending an out-of-office message to external senders.	• InternalOnly • External	Get-Mailbox
Fax	*facsimileTelephoneNumber*	This property contains the fax number of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-User
FirstName	*givenName*	This property	• String	Get-Contact

		contains the ANR search for the first name of the recipient.	• Wildcard character accepted	Get-Recipient Get-User
ForwardingAddress	*altRecipient*	This property contains the forwarding address to which messages are sent.	Existing SMTP address	Get-Mailbox Get-MailPublicFolder
GrantSendOnBehalfTo	*publicDelegates*	This property contains the DN of other mailboxes that can send messages on behalf of this recipient.	DN	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox
GroupType	*groupType*	This property contains the group type in Active Directory.	• DomainLocal • SecurityEnabled • Global • Universal • BuiltinLocal	Get-DistributionGroup Get-Group
Guid	*objectGuid*	This property contains the GUID for the identity of	GUID	Get-CASMailbox Get-Contact Get-

		the Active Directory object.		DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User
HiddenFromAddressListsEnabled	*msExchHideFromAddressLists*	This property specifies whether the user is visible in the address list.	• Boolean • $true or $false	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox

HomePhone	*homePhone*	This property contains the home phone number of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-User
IncludedRecipients	Not applicable	This property contains the recipient types that are used to build the dynamic distribution group.	• AllRecipients • MailboxUsers • Resources • MailContacts • MailGroups • Mail Users • None	Get-DynamicDistributionGroup
Initials	*initials*	This property contains the initials for the name of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-User
IsLinked	Not applicable	This property specifies whether a folder is linked to a master folder.	• Boolean • $true or $false	Get-Mailbox
IsMailboxEnabled	Not applicable	This property specifies whether a mailbox is mailbox-enabled.	• Boolean • $true or $false	Get-Mailbox
IsResource	Not applicable	This property specifies whether a mailbox is a resource mailbox.	• Boolean • $true or $false	Get-Mailbox

IsShared	Not applicable	This property specifies whether a resource mailbox is shared.	• Boolean • $true or $false	Get-Mailbox
IssueWarningQuota	*mDBStorageQuota*	This property contains the mailbox size at which a warning message is sent to the user.	Unlimited or integer	Get-Mailbox
LanguagesRaw	*msExchUserCulture*	This property contains the language preference for this mailbox.	An acceptable value for this parameter is a combination of an International Organization for Standardization (ISO) 639 two-letter lowercase culture code that is associated with a language and an ISO 3166 two-letter uppercase subculture code that is associated with a country or region. For example, United States English is represented as `en-us`. To learn more about culture codes and for	Get-Mailbox

			a full list of acceptable values, see CultureInfo Class in the MSDN Library.	
LastName	*sn*	This property contains the ANR search for the last name of the recipient.	• String • Wildcard character accepted	Get-Contact Get-Recipient Get-User
LinkedMasterAccount	Not applicable	If this mailbox is a linked mailbox, this property contains the master account in the forest in which the user account resides.	• GUID • DN • Domain\Account • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias	Get-CASMailbox Get-Mailbox Get-UMMailbox
ManagedBy	*managedBy*	This property contains the DN of the user or contact that manages this group.	DN	Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Recipient
ManagedFolderMailboxPolicy	*msExchMailboxTemplateLink*	This property contains the managed folder mailbox policy that controls messaging	• Name • GUID • DN	Get-Mailbox Get-Recipient

		records management (MRM) for the mailbox.		
Manager	*manager*	This property contains the manager of the recipient.	• String • Wildcard character accepted	Get-Contact Get-Recipient Get-User
MaxBlockedSenders	*msExchMaxBlockedSenders*	This property contains the maximum number of senders that can be included in the Blocked Senders list.	Integer	Get-Mailbox
MaxReceiveSize	*delivContLength*	This property contains the maximum size of messages that this recipient can receive.	Unlimited or integer	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox
MaxSafeSenders	*msExchMaxSafeS*	This property	Integer	Get-Mailbox

	enders	contains the maximum number of senders that can be included in the Safe Senders list.		
MaxSendSize	*submissionContLength*	This property contains the maximum size of messages that this recipient can send.	Unlimited or integer	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox
MemberOfGroup	*memberOf*	This property contains the groups of which the recipient is a member.	String	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailUser

				Get-MailPublicFolder Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User
MobilePhone	*mobile*	This property contains the mobile phone number of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-User
Name	*name*	This property contains the name of the recipient.	String	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox

				Get-UMMailbox Get-User
Notes	*info*	This property contains specific comments about the recipient.	• String • Wildcard character accepted	Get-Contact Get-DynamicDistributionGroup Get-Group Get-Recipient Get-User
Office	*physicalDeliveryOfficeName*	This property contains the office of the recipient.	String	Get-Contact Get-Mailbox Get-Recipient Get-User
OfflineAddressBook	*msExchUseOAB*	This property contains the offline address book (OAB) that is associated with this mailbox.	• Name • GUID • DN	Get-Mailbox
OperatorNumber	*msExchUMOperatorNumber*	This property contains the string of digits for the personal operator.	• String • Wildcard character accepted	Get-UMMailbox
OtherFax	*otherFacsimileTelephoneNumber*	This property contains the additional fax number of the	• String • Wildcard character accepted	Get-Contact Get-User

		user or contact.		
OtherHomePhone	*otherHomePhone*	This property contains the additional home phone number of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-User
OtherTelephone	*otherTelephone*	This property contains the additional telephone number of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-User
Pager	*pager*	This property contains the pager number of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-User
Phone	*telephoneNumber*	This property contains the phone number of the recipient.	• String • Wildcard character accepted	Get-Contact Get-Recipient Get-User
PhoneticDisplayName	*msDS-PhoneticDisplayName*	This property contains a phonetic pronunciation of the *DisplayName* property. UM uses this property for automatic	• String • Wildcard character accepted	Get-Contact Get-DynamicDistributionGroup Get-Group Get-MailPublicFolder

		speech recognition.		Get-User
PoliciesExcluded	*msExchPoliciesExcluded*	This property contains the GUIDs of any policies that are excluded.	GUID	Get-Recipient
PostalCode	*postalCode*	This property contains the postal code of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-Recipient Get-User
PostOfficeBox	*postOfficeBox*	This property contains the post office box number of the user or contact.	• String • Wildcard character accepted	Get-Contact Get-User
PrimarySmtpAddress	Not applicable	This property contains the primary SMTP address, which is the email address that external users will see when they receive a message from this recipient.	SMTP address	Get-CASMailbox Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-

				RemoteMailbox Get-UMMailbox
ProhibitSendQuota	*mDBOverQuotaLimit*	This property contains the mailbox size at which the user associated with this mailbox can no longer send messages.	Unlimited or integer	Get-Mailbox
ProhibitSendReceiveQuota	*mDBOverHardQuotaLimit*	This property contains the mailbox size at which the user associated with this mailbox can no longer send or receive messages.	Unlimited or integer	Get-Mailbox
PublicFolderContacts	*pFContacts*	This property contains the contacts for the public folder.	Multiple DNs	Get-MailPublicFolder
PublicFolderType	*msExchPFTreeType*	This property specifies the public folder type.	• GeneralPurpose • MAPI • Network News Transfer Protocol (NNTP) • NotSpecified	Get-MailPublicFolder
RecipientFilter	*msExchQueryFilter*	This property contains the recipient filter	String	Get-DynamicDistributionGroup

		that has been applied.		
RecipientLimits	*msExchRecipLimit*	This property contains the maximum number of recipients per message to which this mailbox can send.	Unlimited or integer	Get-Mailbox Get-MailContact Get-MailUser Get-RemoteMailbox
RecipientType	Not applicable	This property specifies the recipient type.	• UserMailbox • MailUser • MailContact • MailUniversalDistributionGroup • MailUniversalSecurityGroup • MailNonUniversalGroup • DynamicDistributionGroup • PublicFolder	Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-User
RecipientTypeDetails	Not applicable	This property specifies the recipient subtype.	• ConferenceRoomMailbox • EquipmentMailbox	Get-Contact Get-DistributionGroup

			• LegacyMailbox • LinkedMailbox • UserMailbox • MailContact • DynamicDistributionGroup • MailForestContact • MailNonUniversalGroup • MailUniversalDistributionGroup • MailUniversalSecurityGroup • MailUser • PublicFolder • SharedMailbox	Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-User
RejectMessagesFrom	*unauthOrig*	This property contains the recipients from whom messages will be rejected by this recipient.	• DN • Canonical name • GUID • Name • Display name • Alias • Exchange DN • Primary SMTP address	Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox
RejectMessagesFromDLMembers	*dLMemRejectPerms*	This property specifies the distribution	• DN • Canonical name • GUID	Get-DistributionGroup Get-

		groups from which this recipient will reject messages.	• Name • Display name • Alias • Exchange DN • Primary SMTP address	DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox
ResourceCapacity	*msExchResourceCapacity*	This property contains the capacity of this resource mailbox.	Non-negative integer	Get-Mailbox
ResourceCustom	Not applicable	This property contains custom properties for this resource mailbox.	Custom property defined by **Set-ResourceConfig**	Get-Mailbox
RetainDeletedItemsFor	*garbageCollPeriod*	This property contains the length of time to keep deleted items.	Time span: *dd.hh:mm:ss* where *dd* = days, *hh* = hours, *mm* = minutes, and *ss* = seconds	Get-Mailbox
RulesQuota	*msExchMDBRulesQuota*	This property contains the limit for the size of rules for this mailbox.	String	Get-Mailbox
SamAccountNam	*SamAccountNam*	This property	String	Get-CASMailbox

e	*e*	contains the logon name that is used to support client computers and servers running older versions of Microsoft Windows operating systems.		Get-DistributionGroup Get-Group Get-Mailbox Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User
SendOofMessageToOriginatorEnabled	*oOFReplyToOriginator*	This property specifies whether out-of-office messages from distribution group members are sent to the message sender.	• Boolean • $true or $false	Get-DistributionGroup Get-DynamicDistributionGroup
ServerName	Not applicable	This property contains the name of the server on which the object resides	Server name	Get-CASMailbox Get-Mailbox Get-MailboxStatistics Get-Recipient Get-UMMailbox
SimpleDisplayName	*displayNamePrintable*	This property contains an alternative display name of	• String • Wildcard character accepted	Get-Contact Get-DistributionGroup Get-

		the object when only a limited set of characters is permitted.		DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox Get-User
StateOrProvince	*st*	This property contains the state or province information that is defined for this recipient.	• String • Wildcard character accepted	Get-Contact Get-Recipient Get-User
StreetAddress	*streetAddress*	This property contains the street address that is defined for the user or contact.	String	Get-Contact Get-User
TelephoneAssistant	*telephoneAssistant*	This property contains the telephone number of the contact's assistant.	• String • Wildcard character accepted	Get-Contact Get-User

Title	*title*	This property contains the title of the recipient.	String	Get-Contact Get-Recipient Get-User
UMDtmfMap	*msExchUMDtmfMap*	This property contains a user-defined dual tone multi-frequency (DTMF) map for the UM-enabled user. DTMF is also referred to as *touch-tone*.	• String • Wildcard character accepted	Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox Get-UMMailbox Get-User
UMEnabled	Not applicable	This property specifies whether UM is enabled for this mailbox.	• Boolean • $true or $false	Get-Mailbox Get-Recipient Get-UMMailbox
UMMailboxPolicy	*msExchUMTemplateLink*	This property contains the UM mailbox policy for the mailbox. You use UM mailbox policies to set UM	String	Get-Recipient Get-UMMailbox

		settings for UM-enabled users, such as personal identification number (PIN) policies and dialing restrictions.		
UMRecipientDialPlanId	*msExchUMRecipientDialPlanLink*	This property contains the dial plan identifier for the mailbox, user, or contact.	DN	Get-Contact Get-UMMailbox Get-User
UseDatabaseQuotaDefaults	*mDBUseDefaults*	This property specifies whether the mailbox uses the quota attributes for the mailbox database in which this mailbox resides. The quota attributes are: ProhibitSendQuota, ProhibitSendReceiveQuota, IssueWarningQuota, and RulesQuota.	• Boolean • $true or $false	Get-Mailbox
UserPrincipalNa	*userPrincipalNam*	This property	• User logon name • User principal	Get-Mailbox

me	*e*	contains the UPN for this recipient. The UPN is the logon name for the user and consists of a user name and a suffix. Typically, the suffix is the domain name in which the user account resides.	name • Wildcard character accepted	Get-MailUser Get-Recipient Get-RemoteMailbox Get-User
WhenChanged	*WhenChanged*	This property contains the date and time stamp when the object was last changed.	Date-time stamp	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox

				Get-User
WhenCreated	*whenCreated*	This property contains the date and time stamp when the object was created.	Date-time stamp	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User
WindowsEmailAddress	*mail*	This property contains the Windows email address for this mailbox. This address is not used by Exchange.	• Email address • Wildcard character accepted	Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact

				Get-MailPublicFolder Get-MailUser Get-RemoteMailbox Get-User

Advanced filterable properties

Although the following table lists filterable properties that are not commonly used, they are included in this topic for reference purposes.

Property name	LDAP name	Cmdlets that accept this property
DeletedItemFlags	*deletedItemFlags*	Get-Mailbox
DeliverToMailboxAndForward	*deliverAndRedirect*	Get-Mailbox Get-MailPublicFolder
DirectReports	*directReports*	Get-Contact Get-User
ExchangeSecurityDescriptor	*msExchMailboxSecurityDescriptor*	Get-Mailbox
ExchangeUserAccountControl	*msExchUserAccountControl*	Get-Mailbox Get-MailUser Get-RemoteMailbox
HasActiveSyncDevicePartnership	Not applicable	Get-CASMailbox Get-Recipient
Id	*distinguishedName*	Get-CASMailbox Get-Contact Get-DistributionGroup

		Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User
ImapEnabled	Not applicable	Get-CASMailbox
IsSecurityPrincipal	Not applicable	Get-User
LdapRecipientFilter	*msExchDynamicDLFilter*	Get-DynamicDistributionGroup
LegacyExchangeDN	*legacyExchangeDN*	Get-CASMailbox Get-DistributionGroup Get-DynamicDistributionGroup Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox Get-UMMailbox
MAPIEnabled	Not applicable	Get-CASMailbox
MasterAccountSid	*msExchMasterAccountSid*	Get-Mailbox
Members	*member*	Get-Group

NTSecurityDescriptor	*ntSecurityDescriptor*	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User
ObjectCategory	*objectCategory*	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User

ObjectClass	*objectClass*	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailboxStatistics Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox Get-Recipient Get-UMMailbox Get-User
ObjectState	Not applicable	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailboxStatistics Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox

		Get-UMMailbox Get-User
OriginalId	Not applicable	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox Get-Recipient Get-UMMailbox Get-User
OriginalPrimarySmtpAddress	Not applicable	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox

		Get-UMMailbox Get-User
OriginalWindowsEmailAddress	Not applicable	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User
OWAEnabled	Not applicable	Get-CASMailbox
OWACalendarEnabled	Not applicable	Get-CASMailbox
OWAContactsEnabled	Not applicable	Get-CASMailbox
OWATasksEnabled	Not applicable	Get-CASMailbox
OWAJournalEnabled	Not applicable	Get-CASMailbox
OWANotesEnabled	Not applicable	Get-CASMailbox
OWARemindersAndNotificationsEnabled	Not applicable	Get-CASMailbox
OWAPremiumClientEnabled	Not applicable	Get-CASMailbox

OWASpellCheckerEnabled	Not applicable	Get-CASMailbox
OWASearchFoldersEnabled	Not applicable	Get-CASMailbox
OWASignaturesEnabled	Not applicable	Get-CASMailbox
OWAThemeSelectionEnabled	Not applicable	Get-CASMailbox
OWAJunkEmailEnabled	Not applicable	Get-CASMailbox
OWAUMIntegrationEnabled	Not applicable	Get-CASMailbox
OWAWSSAccessOnPublicComputersEnabled	Not applicable	Get-CASMailbox
OWAWSSAccessOnPrivateComputersEnabled	Not applicable	Get-CASMailbox
OWAUNCAccessOnPublicComputersEnabled	Not applicable	Get-CASMailbox
OWAUNCAccessOnPrivateComputersEnabled	Not applicable	Get-CASMailbox
OWAActiveSyncIntegrationEnabled	Not applicable	Get-CASMailbox
OWAChangePasswordEnabled	Not applicable	Get-CASMailbox
OWAAllAddressListsEnabled	Not applicable	Get-CASMailbox
OWARulesEnabled	Not applicable	Get-CASMailbox
OWAPublicFoldersEnabled	Not applicable	Get-CASMailbox
OWASMimeEnabled	Not applicable	Get-CASMailbox
OWARecoverDeletedItemsEnabled	Not applicable	Get-CASMailbox
PoliciesIncluded	*msExchPoliciesIncluded*	Get-Recipient

PopEnabled	Not applicable	Get-CASMailbox
ProtocolSettings	*protocolSettings*	Get-CASMailbox
PublicFolderRootUrl	*msExchPfRootUrl*	Get-MailPublicFolder
RawCanonicalName	*canonicalName*	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient Get-RemoteMailbox Get-UMMailbox Get-User
RawName	*name*	Get-CASMailbox Get-Contact Get-DistributionGroup Get-DynamicDistributionGroup Get-Group Get-Mailbox Get-MailContact Get-MailPublicFolder Get-MailUser Get-Recipient

		Get-RemoteMailbox Get-UMMailbox Get-User
RecipientContainer	*msExchDynamicDLBaseDN*	Get-DynamicDistributionGroup
ReportToManagerEnabled	*reportToOwner*	Get-DistributionGroup Get-DynamicDistributionGroup
ReportToOriginatorEnabled	*reportToOriginator*	Get-DistributionGroup Get-DynamicDistributionGroup
RequireAllSendersAreAuthenticated	*msExchRequireAuthToSendTo*	Get-DistributionGroup Get-DynamicDistributionGroup Get-MailContact Get-MailPublicFolder Get-MailUser Get-RemoteMailbox
ResourceType	Not applicable	Get-Mailbox Get-Recipient
ServerLegacyDN	*msExchHomeServerName*	Get-CASMailbox Get-Mailbox Get-Recipient Get-UMMailbox
Sid	*objectSid*	Get-Group Get-User
SidHistory	*SIDHistory*	Get-Group Get-User

SIPResourceIdentifier	Not applicable	Get-UMMailbox
UserAccountControl	*userAccountControl*	Get-Mailbox
WebPage	*wWWHomePage*	Get-Contact Get-User

Filterable properties for the -ContentFilter parameter

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-11-19*

This topic lists the filterable properties for the *ContentFilter* parameter. The *ContentFilter* parameter is used to export messages to a .pst file that match the filter. The *ContentFilter* parameter is used in the New-MailboxExportRequest cmdlet.

Filterable properties

Many of the properties for the *ContentFilter* parameter accept wildcard characters. If you use a wildcard character, use the **-like** operator instead of the **-eq** operator. The **-like** operator is used to find pattern matches in rich types, such as strings, whereas the **-eq** operator is used to find an exact match.

The following table contains a list of the filterable properties for the *ContentFilter* parameter. This table lists the name of the property, a description, the acceptable values, and a syntax example. For more information about OPATH filters, see Filters in recipient Shell commands.

Property	Description	Values	Example syntax
All	This property returns all messages that have a particular string in any of the indexed properties. For example, use this property if you want to	String Wildcard	`-ContentFilter {A`

	export all messages that have "Ayla" as the recipient, the sender, or have the name mentioned in the message body.		
Attachment	This property returns messages that have the specified string in the content of an attachment or in the attachment's file name.	String Wildcard	`-ContentFilter {A`
BCC	This property returns sent messages that have the specified recipient in the Bcc field.	Display name Alias SMTP address LegacyDN Wildcard	`-ContentFilter {(`
Body	This property returns messages that have the specified string within the message body.	String Wildcard	`-ContentFilter {B`
Category	This property returns messages that have a matching category. Categories are set by users or Inbox rules.	String Wildcard	`-ContentFilter {C`
CC	This property returns sent messages that have the specified	Display name Alias SMTP address	`-ContentFilter {(`

	recipient in the Cc field.	LegacyDN Wildcard	
Expires	This property returns messages that have a specified expiration time stamp.	Date-Time stamp	`-ContentFilter {`
HasAttachment	This property returns messages with or without attachments.	Boolean `$true` or `$false`	`-ContentFilter {`
Importance	This property returns messages that have a specified importance level.	0 or "Low" 1 or "Normal" 2 or "High"	`-ContentFilter {` `-ContentFilter {`
IsFlagged	This property returns messages that have been flagged by the user or Inbox rule.	Boolean `$true` or `$false`	`-ContentFilter {`
IsRead	This property returns messages that have been read or not read by the user.	Boolean `$true` or `$false`	`-ContentFilter {`
MessageKind	This property returns messages that are of the specified type.	Calendar Contact Doc Email Fax InstantMessage Journal	`-ContentFilter {` `-ContentFilter {`

		Note Post RSSFeed Task Voicemail	
MessageLocale	This property returns messages that are of the specified locale.	CultureInfo	`-ContentFilter {M` `-ContentFilter {M`
Participants	This property returns messages that have the specified recipient in the To, Bcc, or Cc fields.	Display name Alias SMTP address LegacyDN Wildcard	`-ContentFilter {(`
PolicyTag	This property returns messages that have a policy tag. The Exchange store persists policy tags as GUIDs. Therefore, the string can contain either an explicit GUID value, which is then searched by the PR_POLICY_TAG, or a wildcard string. If the supplied value isn't a GUID, the command uses Active Directory information	String Wildcard	`-ContentFilter {P`

	to resolve names to GUIDs.		
Received	This property returns messages that were received with the specified Received time stamp.	Date-Time stamp	`-ContentFilter {R` `{(Received -lt '(`
Sender	This property returns messages that were received from the specified sender.	Display name Alias SMTP address LegacyDN Wildcard	`ContentFilter {Se`
Sent	This property returns messages that were sent by with the specified Sent time stamp.	Date-Time stamp	`-ContentFilter {S` `-ContentFilter {(`
Size	This property returns messages that are of a specific size.	B (bytes) KB (kilobytes) MB (megabytes)	`-ContentFilter {S`
Subject	This property returns messages that have the specified string within the subject of the message.	String Wildcard	`-ContentFilter {S`
To	This property returns sent messages that have the specified	Display name Alias SMTP address	`-ContentFilter {T`

	recipient in the To field.	LegacyDN Wildcard	

Manage Permissions for Recipients

Exchange Server 2013 > Recipients >

Applies to: *Exchange Online*

Topic Last Modified: *2014-06-02*

You can use the EAC or the Shell to assign permissions to users or groups (called *delegates*) that allow them to open or send messages from other mailboxes. Permissions can be assigned to user mailboxes, linked mailboxes, resource mailboxes, and shared mailboxes. You can also assign permissions to distribution groups, dynamic distribution groups, and mail-enabled security groups to allow delegates to send messages on behalf of the group. You can assign delegates the following permissions to access mailboxes or send messages on behalf of mailboxes or groups:

- **Full Access** This permission allows a delegate to open a user's mailbox and access the contents of the mailbox. However, assigning the Full Access permission doesn't allow the delegate to send mail from the mailbox. You have to assign the delegate the Send As or the Send on Behalf permission to send mail.

The Full Access permission isn't available when configuring permissions for groups.

Note:

If you assign the Full Access permission to access a mailbox that is hidden from address lists, the delegate won't be able to open the mailbox.

- **Send As** This permission allows delegates to use the mailbox to send messages. After this permission is assigned to a delegate, any message that the delegate sends from the mailbox will appear to have been sent by the mailbox owner. However, this permission doesn't allow a delegate to sign in to the user's mailbox. It only allows users to open the mailbox. If this permission is assigned to a group, a message sent by the delegate will appear to have been sent by the group.
- **Send on Behalf** This permission also allows a delegate to use the mailbox to send messages. After this permission is assigned to a delegate, the **From** address in any message sent by the delegate indicates that the message was sent by the delegate on behalf of the mailbox owner.

The Send on Behalf permission isn't available when configuring permissions for shared mailboxes.

What do you need to know before you begin?

- Estimated time to complete each procedure: 2 minutes.
- Procedures in this topic require specific permissions. See each procedure for its permissions

information.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Assign permissions to a mailbox

As previously stated, you can assign delegates permissions to user mailboxes, linked mailboxes, resource mailboxes, and shared mailboxes. You can also use the Shell to assign delegates permissions to access a discovery mailbox.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Permissions and delegation" entry in the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Use the EAC to assign permissions

The following procedure shows how to assign permissions to a user mailbox. You follow a similar procedure to assign permissions to resource or shared mailboxes by navigating to the **Resources** or **Shared** page in the EAC and selecting the mailbox to assign the permissions to.

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of mailboxes, click the mailbox that you want to assign permissions for, and then click **Edit** .
3. On the mailbox properties page, click **Mailbox Delegation**.
4. To assign permissions to delegates, click **Add +** under the appropriate permission to display a page that lists all recipients in your Exchange organization that can be assigned the permission. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search** .

To remove a permission for a recipient, under the appropriate permission, select the recipient and then click **Remove −**.

5. Click **Save** to save your changes.

Use the EAC to bulk assign permissions

Use the following steps to bulk assign permissions.

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. Select the mailboxes that you wish to assign permissions to.
3. Click or tap **More options** in the right pane, and under **Mailbox Delegation** choose, **Add**.
4. On the **bulk add delegation** page, click or tap **Add +** under the appropriate permission to display a page that lists all recipients in your Exchange organization that can be assigned the

permission. Select the recipients you want, add them to the list, and then click **OK**.

To remove a permission for recipients, under the appropriate permission, select the recipients and then click **Remove** −.

Use the EAC to assign a user permission to send email from another user's mailbox

The following procedure shows how to assign a user permission to send email from another user's mailbox.

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of mailboxes, click the mailbox that you want to assign send as permissions for, and then click **Edit** ✎.
3. On the mailbox properties page, click **Mailbox Delegation**.
4. To assign permissions to delegates, click **Add** + under **Send As** or **Send on Behalf** to display a page that lists all recipients in your Exchange organization that can be assigned the permission. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search** 🔍.

The **Send As** permission allows the delegate to send email from this mailbox.

The **Send on Behalf** permission allows the delegate to send email on behalf of this mailbox. The **From** line in any message sent by a delegate indicates that the message was sent by the delegate on behalf of the mailbox owner.

Note:
If the user also needs to be able to open and view the content of that mailbox, you must also assign the user the **Full Access** permission.

5. Click **Save** to save your changes.

Use the EAC to assign a user permission to send email from a group

The following procedure shows how to assign a user permission to send email from a group.

1. In the EAC, navigate to **Recipients** > **Groups**.
2. In the list of groups, click the group that you want to assign send as permissions for, and then click **Edit** ✎.
3. On the group properties page, click **Group Delegation**.
4. To assign permissions to delegates, click **Add** + under **Send As** or **Send on Behalf** to display a page that lists all recipients in your Exchange organization that can be assigned the permission. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search** 🔍.

The **Send As** permission allows the delegate to send email from this group.

The **Send on Behalf** permission allows the delegate to send email on behalf of this group. The **From** line in any message sent by a delegate indicates that the message was sent by the delegate on behalf of the group.

5. Click **Save** to save your changes.

Use the EAC to assign full access permissions

The following procedure shows how to assign full access permissions to a user mailbox.

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list of mailboxes, click the mailbox that you want to assign full access permissions for, and then click **Edit** .
3. On the mailbox properties page, click **Mailbox Delegation**.
4. To assign permissions to delegates, click **Add +** under **Full Access** to display a page that lists all recipients in your Exchange organization that can be assigned the permission. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search** .

The **Full Access** permission allows a delegate to open a user's mailbox and access the contents of the mailbox.

Note:
Assigning the **Full Access** permission doesn't allow the delegate to send mail from the mailbox. You have to assign the delegate the **Send As** or the **Send on Behalf** permission to send mail.

5. Click **Save** to save your changes.

Use the Shell to assign permissions

The following sections show how to use the Shell to manage Full Access, Send As, and Send on Behalf permissions for mailboxes.

Manage the Full Access permission

The following examples show how to use the **Add-MailboxPermission** and **Remove-MailboxPermission** cmdlets to manage Full Access permissions.

This example assigns the delegate Raymond Sam the Full Access permission to the mailbox of Terry Adams.

```
Add-MailboxPermission -Identity "Terry Adams" -User
raymonds -AccessRights FullAccess -InheritanceType all
```

This example assigns Esther Valle the Full Access permission to the organization's default discovery search mailbox.

```
Add-MailboxPermission -Identity
"DiscoverySearchMailbox{D919BA05-46A6-415f-80AD-
7E09334BB852}" -User estherv -AccessRights FullAccess -
InheritanceType all
```

This example assigns members of the Helpdesk distribution group the Full Access permission to the Helpdesk Tickets shared mailbox.

```
Add-MailboxPermission "HelpdeskTickets" -User helpdesk -
```

```
AccessRights FullAccess -InheritanceType all
```

This example removes Jim Hance's Full Access permission to Ayla Kol's mailbox.

```
Remove-MailboxPermission -Identity ayla -User "Jim Hance" -
AccessRights FullAccess -Inheritance
```

For detailed syntax and parameter information, see the following topics:

- Add-MailboxPermission
- Remove-MailboxPermission

Manage the Send As permission

The following examples show how to manage Send As permissions in Exchange Server 2013 and in Exchange Online. In Exchange 2013, you have to use the **Add-ADPermission** and **Remove-ADPermission** cmdlets; in Exchange Online, you have to use the **Add-RecipientPermission** and **Remove-RecipientPermission** cmdlets. In both cases, you use the *Identity* parameter to specify the name of the mailbox on which the Send As permission should be added or removed and the *User* or *Trustee* parameter to specify the delegate (for example, a user or group) that will be assigned or unassigned the Send As permission.

Tip:
Use the **Get-Recipient** cmdlet to retrieve the *Name* property for the mailbox and the delegate. Use these values to assign the Send As permission.

Exchange Server 2013

This example assigns the Send As permission to the Helpdesk group on the shared mailbox Helpdesk Support Team.

```
Add-ADPermission -Identity helpdesksupport -User
helpdeskgroup -ExtendedRights "Send As"
```

This example removes the Send As permission for the user Pilar Pinilla on the mailbox of James Alvord.

```
Remove-ADPermission -Identity "James Alvord" -User pilarp -
ExtendedRights "Send As"
```

For detailed syntax and parameter information, see:

- Add-ADPermission
- Remove-ADPermission

Exchange Online

This example assigns the Send As permission to the Printer Support group on the shared mailbox named Contoso Printer Support.

```
Add-RecipientPermission -Identity "Contoso Printer Support"
-Trustee "Printer Support" -AccessRights SendAs
```

This example removes the Send As permission for the user Karen Toh on the mailbox for Yan Li.

```
Remove-RecipientPermission -Identity "Yan Li" -Trustee
"Karen Toh" -ExtendedRights SendAs
```

For detailed syntax and parameter information, see:

- **Add-RecipientPermission**
- **Remove-RecipientPermission**

Manage the Send on Behalf permission

The following examples show how to use the **Set-Mailbox** cmdlet to manage Send on Behalf permissions.

This example assigns the delegate Holly Holt the Send on Behalf permission to the mailbox of Sean Chai.

```
Set-Mailbox -Identity seanc@contoso.com -
GrantSendOnBehalfTo hollyh
```

This example removes the Send on Behalf permission on the Contoso Executives shared mailbox that was assigned to the Temporary Executive Assistants group.

```
Set-Mailbox "Contoso Executives" -GrantSendOnBehalfTo
@{remove="tempassistants@contoso.com"}
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this worked?

To verify that you've successfully assigned permissions to a mailbox or a shared mailbox, do one of the following:

- In the EAC:
  1. Navigate to **Recipients** > **Mailbox** or **Shared**, click the mailbox, and then click **Edit** ✎.
  2. On the mailbox properties page, click **Mailbox Delegation**.
  3. If you assigned permissions to a recipient, verify that the user or group is listed under the appropriate permission. If you removed permissions, verify that the recipient isn't listed under the appropriate permission.

Or

- In the Shell, run one of the following commands, depending on the permission you managed.
  - **Full Access**

```
Get-MailboxPermission -Identity <mailbox>
```

To verify whether a specific delegate is assigned the Full Access permission to a mailbox, run the following command.

```
Get-MailboxPermission -Identity <mailbox> -User <delegate>
```

- **Send As**

In Exchange Server 2013, run the following command.

```
Get-ADPermission -Identity <name of mailbox> -User
<delegate>
```

In Exchange Online, run the following command.

```
Get-RecipientPermission -Identity <mailbox> -Trustee
<delegate>
```

- **Send on Behalf**

```
Get-Mailbox -Identity <mailbox> | FL GrantSendOnBehalfTo
```

Assign permissions to a group

As previously stated, you can assign the Send As and Send on Behalf permissions to distribution groups, dynamic distribution groups, and mail-enabled security groups to allow delegates to send messages as the group or on behalf of the group.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Distribution groups" and "Dynamic distribution groups" entries in the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Use the EAC to assign permissions

1. In the EAC, navigate to **Recipients** > **Groups**.
2. In the list of groups, click the group that you want to assign permissions for, and then click **Edit** .
3. On the group properties page, click **Group Delegation**.
4. To assign permissions to delegates, click **Add +** under the appropriate permission to display a page that displays a list of all recipients in your Exchange organization that can be assigned the permission. Select the recipients you want, add them to the list, and then click **OK**. You can also search for a specific recipient by typing the recipient's name in the search box and then clicking **Search** .

To remove permission for a recipient, under the appropriate permission, select the recipient and then click **Remove −**.

5. Click **Save** to save your changes.

Use the Shell to assign permissions

The following sections show how to use the Shell to manage Send As and Send on Behalf permissions for groups.

Manage the Send As permission

The following examples show how to manage Send As permissions for groups in Exchange Server 2013 and in Exchange Online. In Exchange 2013, you have to use the **Add-ADPermission** and **Remove-ADPermission** cmdlets. In Exchange Online, you have to use the **Add-RecipientPermission** and **Remove-RecipientPermission** cmdlets. In both cases, you use the *Identity* parameter to specify the name of the group on which the Send As permission should be added or removed and the *User* or *Trustee* parameter to specify the delegate (for example, a user or group) that will be assigned or unassigned the Send As permission.

Tip:

Use the **Get-Recipient** cmdlet to retrieve the *Name* property for the group and the delegate. Use these values to assign the Send As permission.

Exchange Server 2013

This example assigns the Send As permission to the Sales Admins group for the group named Contoso Sales Info. This allows members of the sales admin group to send messages as the Contoso Sales Information group.

```
Add-ADPermission -Identity "Contoso Sales Info" -User
"Sales Admins" -ExtendedRights "Send As"
```

This example removes the Send As permission for the user Alan Shen on the group Corporate IT Admins.

```
Remove-ADPermission -Identity "Corporate IT Admins" -User
contoso\alans -ExtendedRights "Send As"
```

For detailed syntax and parameter information, see:

- Add-ADPermission
- Remove-ADPermission

Exchange Online

This example assigns the Send As permission to the Contoso Admins group on the dynamic distribution group named Emergency Broadcast Messages.

```
Add-RecipientPermission -Identity
emergencybroadcast@contoso.com -Trustee "Contoso Admins" -
AccessRights SendAs
```

This example removes the Send As permission for the user Walter Harp on the Printer Resources security group.

```
Remove-RecipientPermission -Identity "Printer Resources" -
Trustee walterh@contoso.com ExtendedRights SendAs
```

For detailed syntax and parameter information, see:

- **Add-RecipientPermission**
- **Remove-RecipientPermission**

Manage the Send on Behalf permission

The following examples show how to use the **Set-DistributionGroup** and **Set-DynamicDistributionGroup** cmdlets to manage Send on Behalf permissions for groups.

This example assigns the delegate Sara Davis the Send on Behalf permission to the Printer Support distribution group.

```
Set-DistributionGroup -Identity printersupport@contoso.com
-GrantSendOnBehalfTo sarad
```

This example assigns the delegate Administrator the Send on Behalf permission to the All Employees dynamic distribution group.

```
Set-DynamicDistributionGroup -Identity "All Employees" -
GrantSendOnBehalfTo administrator
```

This example removes the Send on Behalf permission on the All Employees dynamic distribution group that was assigned to the administrator.

```
Set-DynamicDistributionGroup "All Employees" -
GrantSendOnBehalfTo @{remove="administrator"}
```

For detailed syntax and parameter information, see:

- Set-DistributionGroup
- Set-DynamicDistributionGroup

How do you know this worked?

To verify that you've successfully assigned permissions to a group, do one of the following:

- In the EAC:
  1. Navigate to **Recipients** > **Groups**, click the group, and then click **Edit** ✎.
  2. On the group properties page, click **Group Delegation**.
  3. If you assigned permissions to a recipient, verify that the user or group is listed under the appropriate permission. If you removed permissions, verify that the recipient isn't listed under the appropriate permission.

Or

- In the Shell, run one of the following commands depending on the permission you managed.
  - **Send As**

In Exchange Server 2013, run the following command.

```
Get-ADPermission -Identity <name of group> -User <delegate>
```

In Exchange Online, run the following command.

```
Get-RecipientPermission -Identity <group> -Trustee
<delegate>
```

  - **Send on Behalf**

```
Get-DistributionGroup -Identity <group> | FL
GrantSendOnBehalfTo
```

Or

```
Get-DynamicDistributionGroup -Identity <group> | FL
GrantSendOnBehalfTo
```

Unsupported characters for Exchange 2013 object names

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-22*

This article describes characters that you can't use in object or component names in Exchange 2013. When you create names for objects or components in Exchange 2013, the names can't contain unsupported characters, even though you may be able to create an object using an unsupported character. Also, if you try to import or connect to objects whose names contain unsupported characters, you may receive an error message or experience unexpected behavior.

Unsupported characters

The following table lists characters that aren't supported for use in the names of Exchange-related objects or components. The table also lists the scenario in which problems may occur if unsupported characters are used. Note that there is a maximum length of 64 characters for each

object listed in the table.

Exchange object or component	Exchange scenario	Unsupported characters
Email domain name	Simple Mail Transfer Protocol (SMTP) connector	~ ` ! @ # $ % ^ & * ( ) + = { } | [ ] \ : " ; < > , . ? /
Host name of connector address space	Mail flow	. . (two periods)
Host name for Exchange servers	SMTP	_ (underscore)
Organization or site name	Running the Setup program or moving mailboxes	~ ` ! @ # $ % ^ & * ( ) _ + = { } | [ ] \ : " ; ' < > , . ? /
Organization internal directory name	Directory	~ ` ! @ # $ % ^ & * ( ) _ + = { } | [ ] \ : " ; ' < > , . ? /
Public folder tree name	Viewing and creating the public folder	: ;
Recipient name	SMTP	' "
Recipient policy SMTP address	Viewing the public folder hierarchy	~ ` ! @ # $ % ^ & * ( ) + = { } | [ ] \ : " ; < > , . ? /
Recipient policy SMTP address host name	Mail flow	. . (two periods)
Site internal directory name	Viewing the public folder hierarchy	? ( ) *
Smart host name	SMTP	Leading or trailing spaces

Automatic mailbox distribution

Exchange Server 2013 > Recipients >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-08-13*

When you create or move a mailbox, or mail-enable an existing user, that mailbox needs to be stored in a mailbox database. In Microsoft Exchange Server 2013, you have the option of letting Exchange choose the database for you using automatic mailbox distribution.

With automatic mailbox distribution, Exchange looks at the mailbox databases in your organization, excludes databases that aren't suitable using criteria discussed later in this topic, and then randomly chooses a database where the mailbox should be located. This process randomly distributes mailboxes across all of the suitable mailbox databases in your organization.

Automatic distribution is used when you don't specify the *Database* parameter on the **New-Mailbox** and **Enable-Mailbox** cmdlets or the *TargetDatabase* parameter on the **New-MoveRequest** cmdlet.

Note:

Automatic mailbox distribution is performed only when a mailbox is created on an Exchange 2013 server, moved to an Exchange 2013 server, or when a user is mail-enabled. The **New-Mailbox**, **New-MoveRequest**, and **Enable-Mailbox** cmdlets must be run from a server running Exchange 2013. Exchange doesn't redistribute mailboxes to distribute load across databases automatically based on server load.

The following process is used to find a suitable mailbox database where a new or moved mailbox should be located:

1. Exchange retrieves a list of all mailbox databases in the Exchange 2013 organization.
2. Any mailbox database that's marked for exclusion from the distribution process is removed from the available list of databases. You can control which databases are excluded. For more information, see Exclude Databases from Automatic Distribution later in this topic.
3. Any mailbox database that's outside of the database management scopes applied to the administrator performing the operation is removed from the list of available databases. For more information, see Database Scopes later in this topic.
4. Any mailbox database that's outside of the local Active Directory site where the operation is being performed is removed from the list of available databases.
5. From the remaining list of mailbox databases, Exchange chooses a database randomly. If the database is online and healthy, the database is used by Exchange. If it's offline or not healthy, another database is chosen at random. If no online or healthy databases are found, the operation fails with an error.

The process of selecting a mailbox database is performed by the Mailbox Resources Management Agent cmdlet extension agent. The `Mailbox Resources Management Agent` is one of several cmdlet extension agents that extend the functionality of running cmdlets. For more information about cmdlet extension agents, see Cmdlet extension agents.

If you never want mailboxes to be distributed automatically, you can disable the `Mailbox Resources Management Agent`. When you disable the agent, the change is applied to the entire Exchange

organization. For more information about how to disable cmdlet extension agents, see Manage cmdlet extension agents.

Exclude Databases from Automatic Distribution

By default, all online and healthy mailbox databases on Exchange 2013 servers in the local Active Directory site can be chosen by automatic mailbox distribution to store a new or moved mailbox. However, you might want to exclude some databases from the distribution process for various reasons. For example, you may designate a mailbox database as a journaling database in which only mailboxes you manually specify should be located. Or you might want to temporarily remove a database from rotation to perform scheduled maintenance. Exchange 2013 gives you the option to either permanently or temporarily exclude databases from the exclusion process using the *IsExcludedFromProvisioning* parameter that can be set using the **Set-MailboxDatabase** cmdlet.

Note:

Two other parameters, *IsSuspendedFromProvisioning* and *IsExcludedFromInitialProvisioning*, are also available on the **Set-MailboxDatabase** cmdlet. These parameters will be removed in a future release of Exchange and their use isn't supported.

The *IsExcludedFromProvisioning* parameter has have two valid values, `$True` and `$False`. When you set this property to `$True`, the mailbox database is excluded from the automatic distribution process. When you set it to `$False`, the mailbox database is included in the automatic distribution process. The default value is `$False`.

To exclude a mailbox database from automatic distribution, use the following command:

```
Set-MailboxDatabase <database name> -
IsExcludedFromProvisioning $True
```

When a mailbox database is excluded from automatic distribution, the only way to create a mailbox in, or move a mailbox to, the database is to use the *Database* parameter on the **New-Mailbox** and **Enable-Mailbox** cmdlets or the *TargetDatabase* parameter on the **New-MoveRequest** cmdlet.

Database Scopes

Database management scopes are an additional level of control over the automatic mailbox distribution process that are available in Exchange 2013. If a mailbox database is online and healthy, it's in the local Active Directory site, and it isn't excluded from the automatic distribution process, Exchange 2013 checks to see if the mailbox database is included in the database scope applied to the administrator running the cmdlet. If it's included in the database scope, it's included in the list of databases available to that administrator.

Database scopes are part of the Role Based Access Control (RBAC) permissions model. For more

information about RBAC and database scopes, see the following topics:

- Understanding Role Based Access Control
- Understanding management role scopes

Database scopes can be useful if you have many mailbox databases in your local Active Directory site that are available to automatic distribution, but you want to limit which databases can be used by certain sets of administrators. For example, your Exchange 2013 servers may serve several agencies but you only want to allow each agency to create or move mailboxes to mailbox databases that are allocated to them.

By default, all administrators in an Exchange 2013 organization can see all of the mailbox databases in the organization. To limit the databases that they can see, and therefore limit the databases they can potentially create mailboxes in or move mailboxes to, you must do the following:

1. Create a custom database management scope using the **New-ManagementScope** cmdlet that includes only the mailbox databases you want the administrator to use.
2. Associate the new database scope with a management role assignment in one of the following ways:
   - Add the new database scope to an existing management role assignment using the *CustomConfigWriteScope* parameter on the **Set-ManagementRoleAssignment** cmdlet. The database scope is now applied to the management role group, universal security group (USG), or user assigned the role assignment.
   - Create a management role assignment using the **New-ManagementRoleAssignment** cmdlet and use the *CustomConfigWriteScope* parameter to specify the new database scope. You can create a role assignment between a management role and a role group, USG, or user.
3. If you created a role assignment to a role group or USG, add users to the role group or USG so that the role assignment and database scope are applied to the users.
4. If applicable, remove the user (or users who are members of role groups or USGs you created in the preceding steps) you assigned the new role assignment to from any other role groups or USGs that might be assigned a database scope that contains databases you don't want them to access.
5. Verify that the administrators have access only to the databases they should have access to.

After you complete these steps, the administrators that are assigned role assignments with the database scopes you created will only be able to create mailboxes in or move mailboxes to the databases you specified.

For more information about how to use database scopes to limit which mailbox databases are available to administrators, see Control automatic mailbox distribution using database scopes.

Collaboration

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-07*

Exchange 2013 provides the following rich features that can help your end users easily collaborate in email:

- Site mailboxes
- Public folders
- Shared mailboxes
- Distribution groups

Each of these features has a different user experience and feature set and should be used based on what the user needs to accomplish and what your organization can provide. For example, site mailboxes provide great documentation collaboration features. However site mailboxes rely on SharePoint Server 2013, so if you aren't planning on deploying SharePoint, you should use public folders to share documents.

This topic compares these collaboration features to help you decide which features to offer your users.

Site mailboxes

A site mailboxe is functionally comprised of a SharePoint 2013 site membership (owners and members), shared storage through an Exchange 2013 mailbox for email messages, and a SharePoint 2013 site to store and share. Essentially, site mailboxes bring Exchange email and SharePoint documents together. For users, a site mailbox serves as a central filing cabinet for the project, providing a place to file project email and documents that can be accessed and edited only by site members. In addition, site mailboxes have a specified lifecycle and are optimized to be used for projects that have set start and end dates. To fully implement site mailboxes, end users must use Outlook 2013.

To learn more, see Site mailboxes.

Public folders

Public folders are designed for shared access and provide an easy and effective way to collect, organize, and share information with other people in your workgroup or organization.

Public folders organize content in a deep hierarchy that's easy to browse. Users discover interesting and relevant content by browsing through branches of the hierarchy that are relevant to them. Users always see the full hierarchy in their Outlook folder view. Public folders are a great technology for distribution group archiving. A public folder can be mail-enabled and added as a member of the distribution group. Email sent to the distribution group is automatically added to the public folder for later reference. Public folders also provide simple document sharing and don't require SharePoint Server 2013 to be installed in your organization. Finally, end users can use public folders

with the following supported Outlook clients: Outlook 2007, Outlook 2010, and Outlook 2013.

To learn more, see Public folders.

Shared mailboxes

A shared mailbox is a mailbox that multiple designated users can access to read and send email messages and to share a common calendar. Shared mailboxes can provide a generic email address (such as info@contoso.com or sales@contoso.com) that customers can use to inquire about your company. If the shared mailbox has the Send As permission assigned when a delegated user responds to the email message, it can appear as though the mailbox (for example, sales@contoso.com) is responding, not the actual user.

To learn more, see Shared mailboxes.

Groups

Groups (also called distribution groups) are a collection of two or more recipients that appears in the shared address book. When an email message is sent to a group, it's received by all members of the group. Distribution groups can be organized by a particular discussion subject (such as "Dog Lovers") or by users who share a common work structure that requires them to communicate frequently.

To learn more, see Recipients.

Which one to use?

The following table gives you a quick glance at each of the collaboration features to help you decide which one to use.

Site mailboxes	Public folders	Shared mailboxes	Groups	
Type of group	Users who work together as a team on a specific project with definitive start and end dates.	With the proper permissions, everyone in your organization can access and search public folders. Public folders are ideal for maintaining history	Delegates working on behalf of a virtual identity, and they can respond to email as that shared mailbox identity. Example: support@tailspintoys.com	Users who need to send email to a group of recipients with a common interest or characteristic.

		or distribution group conversations.		
Ideal group size	Small	Large	Small	Large
Access	Site mailbox owners and members.	Accessible by anyone in your organization.	Users can be granted Full Access and/or Send As permissions. If granted Full Access permissions, users must also add the shared mailbox to their Outlook profile to access the shared mailbox.	For distribution groups, members, must be manually added. For dynamic distribution groups, members are added based on filtering criteria.
Shared calendar?	No	Yes	Yes	No
Email arrives in user's personal Inbox?	No. Email arrives in the site mailbox.	No. Email arrives in the public folder.	No. Email arrives in the Inbox of the shared mailbox.	Yes. Email arrives in the Inbox of a distribution group member.
Supported clients	• Outlook 2013 • SharePoint 2013	• Outlook 2013 • Outlook 2010 • Outlook 2007	• Outlook 2013 • Outlook Web App • Outlook 2010 • Outlook 2007	• Outlook 2013 • Outlook Web App • Outlook 2010 • Outlook 2007

Site mailboxes

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-13*

Email and documents are traditionally kept in two unique and separate data repositories. Most organizations collaborate using both mediums. The challenge is that both email and documents are accessed using different clients. This usually results in a reduction in user productivity and a degraded user experience.

The *site mailbox* is a new concept in Microsoft Exchange 2013 that attempts to solve this problem. Site mailboxes improve collaboration and user productivity by allowing access to both Microsoft SharePoint 2013 documents and Exchange email using the same client interface. A site mailbox is functionally comprised of SharePoint 2013 site membership (owners and members), shared storage through an Exchange 2013 mailbox for email messages and a SharePoint 2013 site for documents, and a management interface that addresses provisioning and lifecycle needs.

Site mailboxes require Exchange 2013 and SharePoint Server 2013 integration and configuration. For more information about how to configure your Exchange 2013 organization to work with your SharePoint Server 2013 organization, see the following topics:

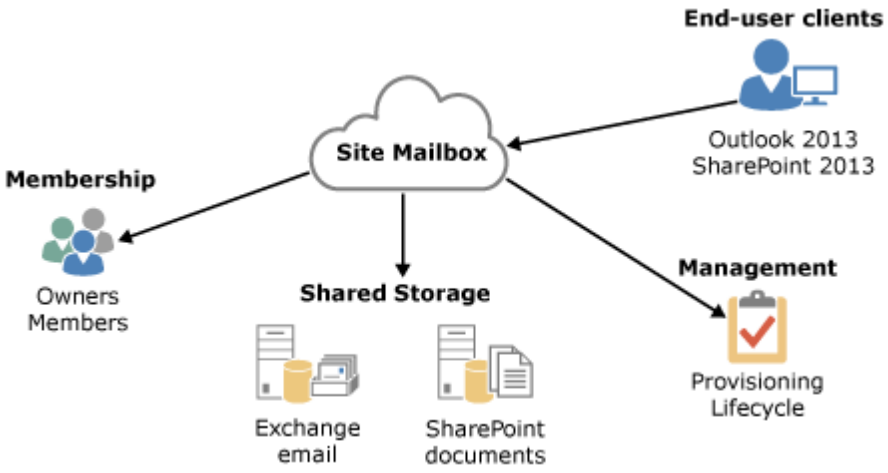
- Configure site mailboxes in SharePoint Server 2013.
- Integration with SharePoint and Lync

For more information about collaboration features in Exchange Server 2013, see Collaboration.

Contents

How do site mailboxes work?

When one project member files mail or documents using the site mailbox, any project member can then access the content. Site mailboxes are surfaced in Outlook 2013 and give users easy access to the email and documents for the projects they care about. Additionally, the same set of content can be accessed directly from the SharePoint site itself. With site mailboxes, the content is kept where it belongs. Exchange stores the email, providing users with the same message view for email conversations that they use every day for their own mailboxes. Meanwhile, SharePoint stores the documents, bringing document coauthoring and versioning to the table. Exchange synchronizes just enough metadata from SharePoint to create the document view in Outlook (e.g. document title, last modified date, last modified author, size).



Site mailbox provisioning policies

Site mailbox quotas can be set by using the **SiteMailboxProvisioningPolicy** cmdlets in the Exchange Management Shell. The Site mailbox provisioning policies only apply to the email that is sent to and from the site mailbox and the size of the site mailbox on the Exchange server. The document repository settings are configured in SharePoint. Although you can create multiple site mailbox provisioning policies using the **New-SiteMailboxProvisioningPolicy** cmdlet, only the default provisioning policy will be applied to all site mailboxes. You can't apply multiple policies within your organization. The provisioning policies allow you to set the following quotas:

Quota	Description	Default setting
IssueWarningQuota	The *IssueWarningQuota* parameter specifies the site mailbox size that triggers a warning message to the site mailbox	4.5 GB
MaxReceiveSize	The *MaxReceiveSize* parameter specifies the maximum size of email messages that can be received by the site mailbox.	36 MB
ProhibitSendReceiveQuota	The *ProhibitSendReceiveQuota* parameter specifies the size at which the site mailbox can no longer send or receive messages.	5 GB

For more information about how to configure site mailbox provisioning policies, see Manage site mailbox provisioning policies.

Return to top

Lifecycle policy and retention

The lifecycle of a site mailbox is managed through a SharePoint. It is through SharePoint that you should perform all site mailbox tasks such as creating and removing site mailboxes. In addition, you can create a SharePoint Lifecycle policy to manage the lifecycle of a site mailbox. For example, you can create a lifecycle policy in SharePoint that automatically closes all site mailboxes after 6 months. If the user still requires the use of the site mailbox, the user can reactivate the site mailbox through SharePoint. We recommend that you use the Lifecycle application is in the farm. Manually deleting active site mailboxes from Exchange will result in orphaned site mailboxes. .

When the lifecycle application in SharePoint closes a site mailbox, the site mailbox is retained for the period stated in the lifecycle policy in the closed state. The mailbox can then be reactivated by an end-user or by an administrator from SharePoint. After the retention period, the Exchange site mailbox that is housed in the mailbox database will have its name prepended with **MDEL:** to indicate that it has been marked for deletion. You will need to manually remove these site mailboxes from the mailbox database in order to free storage space and the alias. If you don't have the SharePoint Lifecycle Policy enabled, you'll lose the ability to determine which site mailboxes are marked for deletion. Until the site mailbox has been removed by an administrator, the content of the mailbox is still recoverable.

You can use the following command to search for and remove site mailboxes that have been marked for deletion.

```
Get-Mailbox MDEL:* | ?{$_.RecipientTypeDetails -eq
"TeamMailbox"} | Remove-Mailbox -Confirm:$false
```

Site mailboxes don't support retention at the item-level. Retention works on a project-level for site mailboxes, so when the entire site mailbox is deleted, the retained items will be deleted.

Compliance

Using the eDiscovery Console in SharePoint, site mailboxes can be part of the In-Place eDiscovery scope as you can do keyword searches against user mailboxes or site mailboxes. In addition, you can put a site mailbox on legal hold. For more info, see In-Place eDiscovery.

Return to top

Backup and restore

Backup and Restore for the Exchange site mailboxes housed on the mailbox server will use the same backup and restore method that you use for all Exchange mailboxes. For more information,

see Database availability groups.

For SharePoint documents, you should backup and restore into the same place. If you restore your SharePoint content to same URLs, then the site mailbox will continue to work and no additional configuration is needed. If you restore to a different URL, then you'll need to run **Set-SiteMailbox** cmdlet to update the *SharePointURL* property. We recommend that you don't restore SharePoint to a new forest.

Manage site mailbox provisioning policies

Exchange Server 2013 > Collaboration > Site mailboxes >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-21*

Site mailbox provisioning policies apply only to email that's sent to and from the site mailbox and to the size of the site mailbox on the Exchange server.

To learn more about site mailboxes, see Site mailboxes.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the"Site mailboxes" entry in the Sharing and collaboration permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.
- Although you can create multiple site mailbox provisioning policies, only the default provisioning policy will be applied to all site mailboxes. You can't apply multiple policies within your organization.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a site mailbox provisioning policy

This example creates the default provisioning policy SM_ProvisioningPolicy with the following settings:

- The warning quota for the site mailboxes is 9 GB.
- The site mailboxes are prohibited from receiving messages when the mailbox size reaches 10 GB.
- The maximum size of email messages that can be sent to site mailboxes is 50 MB.

```
New-SiteMailboxProvisioningPolicy -Name
SM_ProvisioningPolicy -IsDefault -IssueWarningQuota 9GB -
ProhibitSendReceiveQuota 10GB -MaxReceiveSize 50MB
```

View the settings of a site mailbox provisioning policy

This example returns detailed information about all site mailbox provisioning policies in your organization.

```
Get-SiteMailboxProvisioningPolicy | Format-List
```

This example returns all policies in your organization, but only displays the `IsDefault` information to identify which policy is the default policy.

```
Get-SiteMailboxProvisioningPolicy | Format-List IsDefault
```

Make changes to an existing site mailbox provisioning policy

This example changes the site mailbox provisioning policy named Default to allow the maximum size of email messages that can be received by the site mailbox to 25 MB. (When you install Exchange, a provisioning policy is created with the name **Default**.)

```
Set-SiteMailboxProvisioningPolicy -Identity Default -
MaxReceiveSize 25MB
```

This example changes the warning quota to 9.5 GB and the prohibit send and receive quota to 10 GB.

```
Set-SiteMailboxProvisioningPolicy -Identity Default -
IssueWarningQuota 9GB -ProhibitSendReceiveQuota 10GB
```

Configure a site mailbox name prefix

When a new site mailbox is created, by default its email address will have a prefix. The email address prefix allows you to easily search for and query site mailboxes and may help users recognize them as well. If you choose, you can disable the prefix, or change the prefix for your tenant in Office 365 or for a given forest in an on-premises deployment. With the default prefix behavior, if your site mailbox is created in Office 365, the default prefix is **SMO-**. Alternatively, if your site mailbox is created in your on-premises deployment, the prefix is **SM-**. The default behavior differs between these premises so that hybrid customers will not experience conflicts if site mailboxes are created in both locations and are then synced cross-premises.

This example disables the prefix naming by setting the *DefaultAliasPrefixEnabled* parameter to $false.

```
Set-SiteMailboxProvisioningPolicy -Identity Default -
DefaultAliasPrefixEnabled $false -AliasPrefix $null
```

This example changes the default provisioning policy and sets the *AliasPrefix* to FOREST01.

Note:

For deployments with multiple forests, it is recommended that a different prefix is used in each forest in order to prevent conflicts when objects are synced across forests, in the event that site mailboxes have been created with the same name in two or more forests.

```
Set-SiteMailboxProvisioningPolicy -Identity Default -
AliasPrefix FOREST01 -DefaultAliasPrefixEnabled $false
```

Note:

In the case of a hybrid deployment where you have Exchange on-premises and in Office 365, all cloud-based site mailboxes are created with the prefix **SMO-**. The prefixes are different in Office 365 and Exchange on-premises so that hybrid customers will not experience conflicts if site mailboxes are created in both locations and are then synced cross-premises.The AliasPrefix parameter takes precedence over the DefaultAliasPrefixEnabled parameter; therefore, if the *AliasPrefix* parameter is set to a valid, non-null string, each new site mailbox will have that string prepended to the alias.

Delete a site mailbox provisioning policy

This example deletes the default site mailbox policy that was created during Exchange Setup.

```
Remove-SiteMailboxProvisioningPolicy -Identity Default
```

Important:

You must first create and designate another default policy before you can remove the policy named **Default**.

For more information

For detailed syntax and parameter information, see the following topics:

New-SiteMailboxProvisioningPolicy

Get-SiteMailboxProvisioningPolicy

Set-SiteMailboxProvisioningPolicy

Remove-SiteMailboxProvisioningPolicy

Public folders

Exchange Server 2013 > Collaboration >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-18*

Public folders are designed for shared access and provide an easy and effective way to collect, organize, and share information with other people in your workgroup or organization. Public folders help organize content in a deep hierarchy that's easy to browse. Users will see the full hierarchy in Outlook, which makes it easy for them to browse for the content they're interested in.

Note:
Public folders are available in the following Outlook clients: Outlook Web App for Exchange 2013, Outlook 2007, Outlook 2010, Outlook 2013, and Outlook for Mac.

Public folders can also be used as an archiving method for distribution groups. When you mail-enable a public folder and add it as a member of the distribution group, email sent to the group is automatically added to the public folder for later reference.

Public folders aren't designed for the following purposes:

- **Data archiving** Users who have mailbox limits sometimes use public folders instead of mailboxes to archive data. This practice isn't recommended because it affects storage in public folders and undermines the goal of mailbox limits. Instead, we recommend that you use In-Place Archiving as your archiving solution.
- **Document sharing and collaboration** Public folders don't provide versioning or other document management features, such as controlled check-in and check-out functionality and automatic notifications of content changes. Instead, we recommend that you use SharePoint as your documentation sharing solution.

To learn more about public folders and other collaboration methods in Exchange 2013, see Collaboration.

To browse some frequently asked questions about public folders in Exchange 2013, see FAQ: Public folders.

For more information about the limits and quotas for public folders, see Limits for public folders.

For a list of public folder management tasks, see Public folder procedures.

Looking for the Exchange Online version of this topic? See **Public folders in Office 365 and Exchange Online**.

Contents

Public folder architecture

In Exchange 2013, public folders were re-engineered using mailbox infrastructure to take advantage of the existing high availability and storage technologies of the mailbox database. Public folder architecture uses specially designed mailboxes to store both the public folder hierarchy and the content. This also means that there's no longer a public folder database. High availability for the public folder mailboxes is provided by a database availability group (DAG). To learn more about DAGs, see Database availability groups.

The main architectural components of public folders are the public folder mailboxes, which can reside in one or more mailbox databases.

Public folder mailboxes

There are two types of public folder mailboxes: the *primary hierarchy mailbox* and *secondary hierarchy mailboxes*. Both types of mailboxes can contain content:

- **Primary hierarchy mailbox** The primary hierarchy mailbox is the one writable copy of the public folder hierarchy. The public folder hierarchy is copied to all other public folder mailboxes, but these will be read-only copies.
- **Secondary hierarchy mailboxes** Secondary hierarchy mailboxes contain public folder content as well and a read-only copy of the public folder hierarchy.

There are two ways you can manage public folder mailboxes:

- In the Exchange admin center (EAC), navigate to **Public folders** > **Public folder mailboxes**.
- In the Exchange Management Shell, use the ***-Mailbox** set of cmdlets. The following parameters

have been added to the New-Mailbox cmdlet to support public folder mailboxes:

- *PublicFolder* This parameter is used with the **New-Mailbox** cmdlet to create a public folder mailbox. When you create a public folder mailbox, a new mailbox is created with the mailbox type of `PublicFolder`. For more information, see Create a public folder mailbox.
- *HoldForMigration* This parameter is used only if you are migrating public folders from a previous version to Exchange 2013. For more information, see Migrate Public folders from previous versions later in this topic.
- *IsHierarchyReady* This parameter indicates whether the public folder mailbox is ready to serve the public folder hierarchy to users. It's set to `$True` only after the entire hierarchy has been synced to the public folder mailbox. If the parameter is set to $False, users won't use it to access the hierarchy. However, if you set the *DefaultPublicFolderMailbox* property on a user mailbox to a specific public folder mailbox, the user will still access the specified public folder mailbox even if the *IsHierarchyReady* parameter is set to `$False`.
- *IsExcludedFromServingHierarchy* This parameter prevents users from accessing the public folder hierarchy on the specified public folder mailbox. For load-balancing purposes, users are equally distributed across public folder mailboxes by default. When this parameter is set on a public folder mailbox, that mailbox isn't included in this automatic load balancing and won't be accessed by users to retrieve the public folder hierarchy. However, if you set the *DefaultPublicFolderMailbox* property on a user mailbox to a specific public folder mailbox, the user will still access the specified public folder mailbox even if the *IsExcludedFromServingHierarchy* parameter is set for that public folder mailbox.

A secondary hierarchy mailbox will serve only public folder hierarchy information to users if it's specified explicitly on the users' mailboxes using the *DefaultPublicFolderMailbox* property, or if the following conditions are met:

- The *IsHierarchyReady* property on the public folder mailbox is set to `$True`.
- The *IsExcludedFromServingHierarchy* on the public folder mailbox is set to `$False`.

Public folder hierarchy

The public folder hierarchy contains the folders' properties and organizational information, including tree structure. Each public folder mailbox contains a copy of the public folder hierarchy. There's only one writeable copy of the hierarchy, which is in the primary public folder mailbox. For a specific folder, the hierarchy information is used to identify the following:

- Permissions on the folder
- The folder's position in the public folder tree, including its parent and child folders

Note:
The hierarchy doesn't store information about email addresses for mail-enabled public folders. The email addresses are stored on the directory object in Active Directory.

Hierarchy synchronization

The public folder hierarchy synchronization process uses Incremental Change Synchronization (ICS), which provides a mechanism to monitor and synchronize changes to an Exchange store hierarchy or

content. The changes include creating, modifying, and deleting folders and messages. When users are connected to and using content mailboxes, synchronization occurs every 15 minutes. If no users are connected to content mailbox, synchronization will be triggered less often (every 24 hours).If a write operation such as a creating a folder is performed on the primary hierarchy, synchronization is triggered immediately (synchronously) to the content mailbox.

Important:
Because there's only one writeable copy of the hierarchy, folder creation is proxied to the hierarchy mailbox by the content mailbox users are connected to.

In a large organization, when you create a new public folder mailbox, the hierarchy must synchronize to that public folder before users can connect to it. Otherwise, users may see an incomplete public folder structure when connecting with Outlook. To allow time for this synchronization to occur without users attempting to connect to the new public folder mailbox, set the *IsExcludedFromServingHierarchy* parameter on the **New-Mailbox** cmdlet when creating the public folder mailbox. This parameter prevents users from connecting to the newly created public folder mailbox. When synchronization is complete, run the Set-Mailbox cmdlet with the *IsExcludedFromServingHierarchy* parameter set to `false`, indicating that the public folder mailbox is ready to be connected to. You can use also the Get-PublicFolderMailboxDiagnostics cmdlet to view the sync status by the *SyncInfo* and the *AssistantInfo* properties.

For more information, see Create a public folder.

Public folder content

Public folder content can include email messages, posts, documents, and eForms. The content is stored in the public folder mailbox but isn't replicated across multiple public folders mailboxes. All users access the same public folder mailbox for the same set of content. Although a full text search of public folder content is available, public folder content isn't searchable across public folders and the content isn't indexed by Exchange Search.

Considerations

Although there are many advantages to using Exchange 2013 public folders, there are some things to consider before implementing them in your organization:

- It's very important that you carefully plan you public folder deployment. Because there's only one replica of any particular public folder, planning is critical to avoid severe delays in accessing the folders. For example, you can plan to separate heavily used public folders across multiple public folder mailboxes for load balancing purposes.
- Exchange 2013 no longer supports public folder databases. Therefore, there's no coexistence with legacy public folders. As a result, Exchange 2013 is unable to read from the hierarchy stored in a public folder database on Exchange 2010 or Exchange 2007 servers.
- Outlook Web App is supported, but with limitations. You can add and remove favorite public

folders and perform item-level operations such as creating, editing, deleting posts, and replying to posts. However, you can't create or delete public folders from Outlook Web App.

- Although a full text search of public folder content is available, public folder content isn't searchable across public folders and the content isn't indexed by Exchange Search.
- You must use Outlook 2007 or later to access public folders on Exchange 2013 servers.
- Retention policies aren't supported for public folder mailboxes.

Migrate public folders from previous versions

If you already have Exchange 2010 SP3 or Exchange 2007 SP3 RU10 public folders in your organization prior to installing Exchange 2013, you must migrate those public folders to Exchange 2013. To do this, use the **PublicFolderMigrationRequst** cmdlets. For more information, see Migrate public folders to Exchange 2013 from previous versions. If your organization is moving to Exchange Online, you can migrate your public folders to the cloud and upgrade them at the same time. For details, see Migrate legacy public folders to Office 365 and Exchange Online.

Due to the changes in how public folders are stored, legacy Exchange mailboxes are unable to access the public folder hierarchy on Exchange 2013 servers or on Exchange Online. However, user mailboxes on Exchange 2013 servers or Exchange Online can connect to legacy public folders. Exchange 2013 public folders and legacy public folders can't exist in your Exchange organization simultaneously. This effectively means that there's no coexistence between versions. Migrating public folders to Exchange Server 2013 or Exchange Online is currently a one-time cutover process.

For this reason, it's recommended that prior to migrating your public folders, you should first migrate your legacy mailboxes to Exchange 2013 or Exchange Online. For more information about migrating mailboxes, see Mailbox moves in Exchange 2013, **Migrate all mailboxes to Exchange Online with a cutover migration**, and **Migrate mailboxes to Exchange Online with a staged migration**

Public folder moves

You can move public folders to a different public folder mailbox, and you can move public folder mailboxes to different mailbox databases. To move public folders to different public folder mailboxes, use the **PublicFolderMoveRequest** set of cmdlets. Subfolders under the public folder that's being moved won't be moved by default. If you want to move a branch of public folders, you can use the `Move-PublicFolderBranch.ps1` script that's installed by default with Exchange 2013. For more information, see Move a public folder to a different public folder mailbox.

In addition to moving public folders, you can move public folder mailboxes to different mailbox databases by using the **MoveRequest** set of cmdlets. This is the same set of cmdlets that are used for moving regular mailboxes. For more information, see Move a public folder mailbox to a different mailbox database.

PublicFolderMoveRequest cmdlets and the **MoveRequest** cmdlets use the Mailbox Replication

Service to move public folders asynchronously. That means that the cmdlet doesn't do the actual work and, during most of the move, the public folder and public folder mailboxes will still be available to users. Because the Mailbox Replication Service performs mailbox moves, import and export requests, and public folder move requests, it's important to consider throttling and workload management.

Public folder quotas

When created, public folder mailboxes automatically inherit the size limits of the mailbox database defaults. As a result, to accurately evaluate the current storage quota status when using the Get-Mailbox cmdlet, you must review at the *UseDatabaseQuotaDefaults* property in addition to the *ProhibitSendQuota*, *ProhibitSendReceiveQuota*, and *IssueWarningQuota* properties. If the *UseDatabaseQuotaDefaults* property is set to `true`, the per-mailbox settings are ignored and the mailbox database limits are used. If this property is set to `true` and the *ProhibitSendQuota*, *ProhibitSendReceiveQuota*, and *IssueWarningQuota* properties are set to `unlimited`, the mailbox size isn't really unlimited. Instead, you must use the **Get-MailboxDatabase** cmdlet and review the mailbox database storage limits to find out what the limits for the mailbox are. If the *UseDatabaseQuotaDefaults* property is set to `false`, the per-mailbox settings are used. In Exchange 2013, the default mailbox database quota limits are as follows:

- *Issue warning quota*: 1.9 GB
- *Prohibit send quota*: 2 GB
- *Prohibit receive quota*: 2.3 GB

To find the mailbox database quotas, run the Get-MailboxDatabase cmdlet.

To set the quotas on a public folder mailbox, use the Set-OrganizationConfig cmdlet.

Disaster recovery

Exchange 2013 public folders are built on mailbox infrastructure and use the same mechanisms for availability and redundancy. Every public folder mailbox can have multiple redundant copies with automatic failover, just like regular mailboxes. To learn more, see High availability and site resilience.

In addition to the overall disaster recovery scenario, you can also restore public folders in the following situations:

- **Soft-deleted public folder restore** The public folder was deleted but is still within the retention period.
- **Soft-deleted public folder mailbox restore** The public folder mailbox was deleted and is still within the mailbox retention period.
- **Public folder mailbox restore from a recovery database** You can recover an individual public folder mailbox from backup when the deleted mailbox retention period has elapsed. You then extract data from the restored mailbox and copy it to a target folder or merge it with another

mailbox.

In all of these situations, the public folder or public folder mailbox is recoverable by using the **MailboxRestoreRequest** cmdlets.

For more information, see Restore public folders and public folder mailboxes from failed moves.

FAQ: Public folders

Exchange Server 2013 > Collaboration > Public folders >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-18*

This topic provides you with a list of frequently asked questions regarding public folders in Exchange Server 2013. To learn more about public folders, see Public folders.

Have questions about public folders that aren't answered here? Send us an email at Ex2013HelpFeedback@microsoft.com.

Are public folders going away?

No. Public folders are great for Outlook integration, simple sharing scenarios, and for allowing large audiences to access the same data.

Which clients support public folders?

Outlook 2007, Outlook 2010, and Outlook 2013, and Outlook 2011 for Mac users can access public folders. However, users whose mailboxes are on Exchange 2013 servers won't be able to connect to Exchange 2007 or Exchange 2010 public folders from clients that use Exchange Web Services (EWS), such as Outlook for Mac. We recommend that you migrate legacy public folders to Exchange 2013 in order to maintain access for those users.

Are there any limitations in the using the clients?

Outlook Web App is supported, but with some limitations. You can add and remove favorite public folders (if they are Mail or Post public folders) and perform item level operations, such as creating, editing, deleting posts, and replying to posts. But, you can't do the following in Outlook Web App:

- Create or delete public folders
- Drag-and-drop content
- Access public folders located on servers running previous versions of Exchange

In a hybrid scenario, Outlook Web App and Outlook 2011 for Mac aren't supported for cross-premises public folders. Users must be in the same location as the public folders to access them with Outlook 2011 for Mac or Outlook Web App.

How can I store a very large hierarchy in a public folder mailbox?

For more information about public folder storage limits, see Limits for public folders.

How can I view the hierarchy public folder mailbox?

Run the following command:

```
Get-OrganizationConfig | Format-List
RootPublicFolderMailbox
```

For detailed syntax and parameter information, see Get-OrganizationConfig.

How can I create content mailboxes for public folders using Exchange Management Shell cmdlets?

Run the following command to create the first master hierarchy public folder mailbox and the secondary hierarchy mailboxes.

```
New-Mailbox -PublicFolder -Name <name of public folder>
```

For more detail, see Create a public folder.

In previous versions of Exchange, for each mailbox database there was an option to specify its public folder database. How will this work in Exchange 2013?

There is no database-level setting in Exchange 2013. Exchange 2013 has a mailbox-level ability to specify the public folder mailbox, but by default Exchange auto-calculates the per-user hierarchy mailbox.

How are public folder metric tools being used in Exchange 2013?

In Exchange 2013, you can use Get-PublicFolderStatistics and Get-PublicFolderItemStatistics cmdlets to get public folder metrics data. This is the same solution that we had in Exchange 2010, so nothing has changed here. Public folders don't require additional reporting add-ons.

Can public folders distinguish between internal versus third-party access to public folders?

In Exchange 2013, public folder permissions are managed by using Role Based Access Control (RBAC). Access control lists (ACLs) aren't used in Exchange 2013. You can use Get-PublicFolderStatistics and Get-PublicFolderItemStatistics cmdlets to keep track of accounts that are performing administrative tasks and then audit access accordingly. To learn more about RBAC, see Understanding Role Based Access Control.

Does mailbox audit logging work against public folders?

No. Not at this time.

What are the limits on public folders? What are the recommendations?

For more information about public folder limits, see Limits for public folders.

What are the recommendations for splitting public folder mailboxes? Should they stay on the same database?

In previous versions of Exchange, you could split public folders across public folder databases. You can decide whether to split the content of a public folder mailbox to a mailbox on the same mailbox database or a different database. Typically, a split is recommended to be on a separate database, because you want to balance storage and I\O.

Can you set retention policies on public folders?

Just like in previous versions of Exchange, you can set retention limits on items. For details, see Step 1 of Set up public folders in a new organization.

Can you specify which users can use a specific public folder mailbox?

In Exchange 2007 and Exchange 2010, you could specify which users had access to specific public folders. In Exchange 2013, you can set the default public folder mailbox per user. To do so, run the Set-Mailbox cmdlet with the *DefaultPublicFolderMailbox* parameter.

```
Set-Mailbox -Identity kweku@contoso.com -
DefaultPublicFolderMailbox "PF_Administration"
```

If the master hierarchy goes down, what's the user impact?

If the master hierarchy public folder mailbox goes down, users can view but not write to public folders. To help prevent the hierarchy from going down, we recommend that you include your public folders in a database availability group (DAG). To learn about DAGs, see Database availability groups.

Can you change which public folder mailbox is the master hierarchy mailbox?

No. If you try to change the master hierarchy mailbox, you'll receive an error.

Do public folders have full text searching capabilities?

Yes, full text search is available for public folders in Exchange 2013. However, you can't search across multiple public folders.

FAQs about public folder migration

This section contains frequently asked questions about public folder migration. For more information, see Migrate public folders to Exchange 2013 from previous versions.

After migration, what happens to the hierarchy on the source Exchange 2010 servers?

During the finalization stage in migration, a lock is placed on the source server to make it inaccessible to user. This lock remains in place to prevent users from accessing the source public folders after migration completes. Although you can release this lock, we don't recommend doing so because the changes can't be synced to Exchange 2013.

When you migrate public folders, what happens to existing public folder rules?

Public folder rules are migrated along with the data and are kept as public folder rules. They aren't converted to mailbox rules.

What happens if hierarchy changes are performed on the source after the initial .csv file was generated? How would these reflect on the destination?

The .csv file is used to determine the mapping between the source hierarchy and the destination mailbox. It contains only the top-level folders. Child folders under the top-level folders are automatically migrated. Therefore, if a new child folder is added, it's migrated during the process. If a new top-level folder is created, it will be created in the mailbox that contains the writable copy of the hierarchy.

During migration to Exchange 2013 public folders, if there's a long window of time between suspension and finalization, how can I force a delta sync so that users can access public folders during the final sync?

You can force a delta sync to occur before finalization (prior to locking the source) by running the following Shell command:

```
Resume-PublicFolderMigrationRequest \PublicFolderMigration
```

For detailed syntax and parameter information, see Resume-PublicFolderMigrationRequest.

For the migration of a geo-distributed hierarchy, how can I

make sure that the public folders are created in the location nearest to the target users?

As part of the migration process, a .csv file is generated (using the `publicfoldertomailboxmapgenerator.ps1` script). This file contains the folder-to-mailbox mapping for the new hierarchy. You can use this .csv file to create public folder mailboxes in the appropriate geographic location and modify the file to place the required folders in the appropriate mailbox so they are near the target users.

The input .csv file can be generated by running the script `AggregatePFData.ps1`, located in the directory <*Exchange Installation Directory*>\V15\Scripts. Run the script as follows:

```
.\AggregatePFData.ps1 | Select-Object -property
@{Name="FolderName"; Expression = {$_.Identity}},
@{Name="FolderSize"; Expression =
{$_.TotalItemSize.Value.ToBytes()}} | Export-CSV -Path
<Path followed by the name of the CSV>
```

Do existing public folder permissions migrate?

Yes, permissions automatically migrate at the folder level with the data. You don't have to perform this step separately.

Limits for public folders

Exchange Server 2013 > Collaboration > Public folders >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-25*

In Exchange Server 2013, we moved public folders from a traditional database architecture to a mailbox architecture. This shift allows public folders to benefit from things such as the resiliency of a Database Availability Group (DAG) and other mailbox enhancements made over the years. However, there are new limits and performance concerns that should be taken into account. In this document we provide some high level guidance for configuration options you have that could affect public folder performance and connectivity.

Limits

The following table lists the limits for public folders in on-premises Exchange Server 2013. Unless the limits are specifically stated as recommended, the values listed here are the supported limits for public folders.

Important:
Looking for Exchange Online limits for Office 365? See Exchange Online Limits.

Item	Limits	Notes
Total number of public folder mailboxes	100	Although you can create more than 100 public folder mailboxes, it isn't supported. Create a public folder mailbox
Total public folders in hierarchy	100,000	Although you can create more than 100,000 public folders, it isn't supported. Create a public folder
Sub-folders under the parent folder	1,000	While you can create more than 1,000 sub-folders under a parent folder, we don't recommend that you do so. *FolderHierarchyChildrenCountReceiveQuota* parameter on the Set-Mailbox cmdlet.
Folder depth	300	The folder depth is the number levels of nested folders that can exist in one branch of a public folder tree. *FolderHierarchyDepthRecieveQuota* parameter on the Set-Mailbox cmdlet.
Maximum messages per public folder	1 million	*MailboxMessagesPerFolderCountRecieveQuota* parameter on the Set-Mailbox cmdlet.

Maximum individual public folder size	10 GB	This limit doesn't include subfolders beneath a single folder. Configure storage quotas for a mailbox
Public folder mailbox size	100 GB	Configure storage quotas for a mailbox
Number of user logons per public folder mailbox	2,000 concurrent user logons	We recommend that you configure your hierarchy so that you have no more than 2,000 users per public folder mailbox. For example, if you have 20,000 users, you should have 10 public folder mailboxes.
Moved item retention	14 days recommended	Use the *DefaultPublicFolderMovedItemRetention* parameter on the **Set-OrganizationConfig** cmdlet.
Age limit	We recommend that you set this as the same default that you use for regular mailboxes.	These settings can be set at the following levels: • **Organizational level:** *DefaultPublicFolderAgeLimit* parameter on the **Set-OrganizationConfig** cmdlet. • **Mailbox level:** *AgeLimit* parameter on the **Set-Mailbox** cmdlet. • **Folder level:** *AgeLimit* parameter on the **Set-PublicFolder** cmdlet.

Deleted item retention	We recommend that you set this as the same default that you use for regular mailboxes.	These settings can be set at the following levels: • **Organizational level:** *DefaultPublicFolderMovedItemRetention* parameter on the **Set-OrganizationConfig** cmdlet. • **Mailbox level:** *RetainDeletedItemsFor* on the **Set-Mailbox** cmdlet. • **Folder level:** *RetainDeleteItemsFor* parameter on the **Set-PublicFolder** cmdlet.

Public folder procedures

Exchange Server 2013 > Collaboration > Public folders >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-06-03*

Set up public folders in a new organization

Migrate public folders to Exchange 2013 from previous versions

Migrate legacy public folders to Office 365 and Exchange Online

Configure legacy on-premises public folders for a hybrid deployment

Create a public folder mailbox

Create a public folder

Mail-enable or mail-disable a public folder

Update the public folder hierarchy

Remove a public folder

Move a public folder mailbox to a different mailbox database

Move a public folder to a different public folder mailbox

Restore public folders and public folder mailboxes from failed moves

View statistics for public folders and public folder items

Set up public folders in a new organization

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-02-24*

This topic shows you how to get public folders configured and running in a new organization or in an organization that has never previously had public folders.

Note:

For more information about the storage quotas and limits for public folders, see the following topics:

- For public folders in Office 365, see Exchange Online Limits.
- For public folders in on-premises Exchange Server 2013, see Limits for public folders.

For additional management tasks related to public folders in Exchange Server 2013, see Public folder procedures.

For additional management tasks related to public folders in Exchange Online, see **Public folder procedures in Office 365 and Exchange Online**.

What do you need to know before you begin?

- Estimated time to complete this task: 30 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Create the primary public folder mailbox

The primary public folder mailbox contains a writeable copy of the public folder hierarchy plus content and is the first public folder mailbox that you create for your organization. Subsequent public folder mailboxes will be secondary public folder mailboxes, which will contain a read-only copy of the hierarchy plus content.

For detailed steps, see Create a public folder mailbox.

Step 2: Create your first public folder

For detailed steps, see Create a public folder.

Step 3: Assign permissions to the public folder

After you create the public folder, you'll need to assign the **Owner** permissions level so that at least one user can access the public folder from the client and create subfolders. Any public folders created after this one will inherit the permissions of the parent public folder.

1. In the Exchange admin center (EAC), navigate to **Public folders** > **Public folders**.
2. In the list view, select the public folder.
3. In the details pane, under **Folder permissions**, click **Manage**.
4. In **Public Folder Permissions**, click **Add +**.
5. Click **Browse** to select a user.
6. In the **Permission level** list, select a level. At least one user should be an **Owner**.
7. Click **Save**.
8. You can add multiple users by clicking **Add +** and assigning the appropriate permissions using the steps above. You can also customize the permission level by selecting or clearing the check boxes. When you edit a predefined permission level such as **Owner**, the permission level will change to **Custom**.

For information about how to use the Shell to assign permissions to a public folder, see Add-PublicFolderClientPermission.

Step 4 (Optional): Mail-enable the public folder

If you want users to send mail to the public folder, you can mail-enable it. This step is optional. If you don't mail-enable the public folder, users can post messages to the public folder by dragging items into it from within Outlook.

1. In the EAC, navigate to **Public folders** > **Public folders**.
2. In the list view, select the public folder you want to mail-enable.
3. In the details pane, under **Mail settings – Disabled**, click **Enable**.

A warning displays asking if you are sure you want to enable mail for the public folder. Click **Yes**.

The public folder will be mail-enabled and the name of the public folder will become the alias of the public folder. If you have multiple recipients with that name, the public folder's alias will be appended with a number. For example, if you have a distribution group named SalesTeam and you create a public folder named SalesTeam and then mail-enable it, the alias of that public folder will be SalesTeam1.

For information about how to use the Shell to mail-enable a public folder, see Enable-MailPublicFolder.

Migrate public folders to Exchange 2013 from previous versions

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-24*

This topic describes how to migrate your public folders from Exchange Server 2010 SP3 or Exchange 2007 SP3 RU10 to Microsoft Exchange Server 2013 within the same forest.

Note:
This topic refers to the Exchange 2010 SP3 and Exchange 2007 SP3 RU10 servers as the *legacy Exchange server*.

You'll perform the migration by using the ***PublicFolderMigrationRequest** cmdlets (which use the Microsoft Exchange Mailbox Replication service to perform the migration tasks), in addition to the following PowerShell scripts:

- `Export-PublicFolderStatistics.ps1` This script creates the folder name-to-folder size mapping file.
- `Export-PublicFolderStatistics.psd1` This support file is used by the `Export-PublicFolderStatistics.ps1` script and should be downloaded to the same location.
- `PublicFolderToMailboxMapGenerator.ps1` This script creates the public folder-to-mailbox mapping file.
- `PublicFolderToMailboxMapGenerator.strings.psd1` This support file is used by the `PublicFolderToMailboxMapGenerator.ps1` script and should be downloaded to the same location.

Step 1: Download the migration scripts provides details about where to download these scripts.

For additional management tasks related to public folders, see Public folder procedures.

For details about how to migrate public folders to Exchange Online, see Migrate legacy public folders to Office 365 and Exchange Online.

What versions of Exchange are supported for migrating public folders to Exchange 2013?

Exchange supports moving your public folders from the following legacy versions of Exchange Server:

- Exchange Server 2010 SP3
- Exchange Server 2007 SP3 RU10

You can't migrate public folders directly from Exchange 2003. If you're running Exchange 2003 in your organization, you must move all public folder databases and replicas to Exchange 2007 SP3 RU10 or later. No public folder replicas can remain on Exchange 2003.

What do you need to know before you begin?

- You must be assigned the following permissions before you can perform this procedure:
  - In Exchange 2013, you must be a member of the Organization Management role group. For details, see Manage role groups.
  - In Exchange 2010, you must be a member of the Organization Management or Server Management role groups. For details, see Add Members to a Role Group.
  - In Exchange 2007, you need to be assigned the Exchange Organization Administrator role or the Exchange Server Administrator role. In addition, you must be assigned the Public Folder Administrator role and local Administrators group for the target server. For details, see How to Add a User or Group to an Administrator Role.
- Before you migrate, you should consider the Limits for public folders.
- Before you begin, we recommend that you read this topic in its entirety as downtime is required for some steps.
- Before you migrate your public folders, we recommend that you first move all user mailboxes to Exchange 2013. For details, see Mailbox moves in Exchange 2013.
- The Exchange 2010 server must be running SP3 or later.
- The Exchange 2007 server must be running SP3 RU10 or later.
- On the Exchange 2007 server, upgrade to Windows PowerShell 2.0 and WinRM 2.0 for Windows Server 2008 x64 Edition.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Download the migration scripts

1. Download all four of the Microsoft Exchange 2013 public folder migration scripts.
2. Save the scripts to the local computer on which you'll be running PowerShell from. For example, C:\PFScripts.

Step 2: Prepare for the migration

Perform the following prerequisite steps before you begin the migration.

Prerequisite steps on the legacy Exchange server

1. For verification purposes at the end of migration, we recommend that you first run the following Shell commands on the legacy Exchange server to take snapshots of your current public folder deployment.
   a. Run the following command to take a snapshot of the original source folder structure.

```
Get-PublicFolder -Recurse | Export-CliXML C:\PFMigration
\Legacy_PFStructure.xml
```

   b. Run the following command to take a snapshot of public folder statistics such as item count, size, and owner.

```
Get-PublicFolderStatistics | Export-CliXML C:\PFMigration
\Legacy_PFStatistics.xml
```

   c. Run the following command to take a snapshot of the permissions.

```
Get-PublicFolder -Recurse | Get-
PublicFolderClientPermission | Select-Object Identity,User
-ExpandProperty AccessRights | Export-CliXML C:\PFMigration
\Legacy_PFPerms.xml
```

Save the information from the preceding commands for comparison at the end of the migration.

2. If the name of a public folder contains a backslash ****, the public folders will be created in the parent public folder when migration occurs. Before you migrate, we recommend that you rename any public folders that have a backslash in the name.
   a. In Exchange 2010, to locate public folders that have a backslash in the name, run the following command:

```
Get-PublicFolderStatistics -ResultSize Unlimited | Where
{$_.Name -like "*\*"} | Format-List Name, Identity
```

   b. In Exchange 2007, to locate public folders that have a backslash in the name, run the following

command:

```
Get-PublicFolderDatabase | ForEach {Get-
PublicFolderStatistics -Server $_.Server | Where {$_.Name -
like "*\*"}}
```

c. If any public folders are returned, you can rename them by running the following command:

```
Set-PublicFolder -Identity <public folder identity> -Name
<new public folder name>
```

3. Make sure there isn't a previous record of a successful migration. If there is, you'll need to set that value to `$false`. If the value is set to `$true` the migration request will fail.

The following example checks the public folder migration status.

```
Get-OrganizationConfig | Format-List
PublicFoldersLockedforMigration,
PublicFolderMigrationComplete
```

If the status of the *PublicFoldersLockedforMigration* or *PublicFolderMigrationComplete* properties is `$true`, run the following command to set the value to `$false`.

```
Set-OrganizationConfig -
PublicFoldersLockedforMigration:$false -
PublicFolderMigrationComplete:$false
```

⚠**Warning:**
After resetting these properties, you must wait for Exchange to detect the new settings. This may take several minutes. .

For detailed syntax and parameter information, see the following topics:

- Get-PublicFolder
- Get-PublicFolderStatistics
- Get-PublicFolderDatabase
- Set-PublicFolder
- Get-PublicFolderClientPermission
- Get-OrganizationConfig
- Set-OrganizationConfig

Prerequisite steps on the Exchange 2013 server

1. Make sure there are no existing public folder migration requests. If there are, clear them. This step is a prerequisite and isn't required in all cases. It's only required if you think there may be an existing migration request in the pipeline. In any case, the following command below won't affect the new migration. The following example removes any existing public folder migration requests.

```
Get-PublicFolderMigrationRequest | Remove-
PublicFolderMigrationRequest -Confirm:$false
```

2. To make sure there are no existing public folders on the Exchange 2013 servers, run the following commands.

```
Get-Mailbox -PublicFolder
Get-PublicFolder
```

If the above commands return any public folders, use the following commands to remove the public folders.

```
Get-Mailbox -PublicFolder |
Where{$_.IsRootPublicFolderMailbox -eq $false} | Remove-
Mailbox -PublicFolder -Force -Confirm:$false
Get-Mailbox -PublicFolder | Remove-Mailbox -PublicFolder -
Force -Confirm:$false
```

For detailed syntax and parameter information, see the following topics:

- Get-PublicFolderMigrationRequest
- Remove-PublicFolderMigrationRequest
- Get-Mailbox
- Get-PublicFolder
- Get-MailPublicFolder
- Disable-MailPublicFolder
- Remove-PublicFolder
- Remove-Mailbox

Step 3: Generate the .csv files

1. On the legacy Exchange server, run the `Export-PublicFolderStatistics.ps1` script to create the folder name-to-folder size mapping file. The file will contain two columns: **FolderName** and **FolderSize**. The values for the **FolderSize** column will be in displayed bytes. For example, **\PublicFolder01,10000**.

```
.\Export-PublicFolderStatistics.ps1 <Folder to size map
path> <FQDN of source server>
```

   - *FQDN of source server* equals the fully qualified domain name of the Mailbox server where the public folder hierarchy is hosted.
   - *Folder to size map path* equals the file name and path on a network shared folder where you want the .csv file saved. You'll need to access this file from the Exchange 2013 server. If you specify only the file name, the file will be generated in its current location.
2. Run the `PublicFolderToMailboxMapGenerator.ps1` script to create the public folder-to-mailbox mapping file. This file is used to create the correct number of public folder mailboxes on the

Exchange 2013 Mailbox server.

Note:
If the name of a public folder contains a backslash ****, the public folders will be created in the parent public folder. We recommend that you review the .csv file and edit any names that contain the backslash.

```
.\PublicFolderToMailboxMapGenerator.ps1 <Maximum mailbox size in bytes> <Folder to size map path> <Folder to mailbox map path>
```

- *Maximum mailbox size in bytes* equals the maximum size you want to set for the new public folder mailboxes.

Note:
When specifying this setting, be sure to allow for expansion so the public folder mailbox has room to grow.

- *Folder to size map path* equals the file path of the .csv file you created when running the **Export-PublicFolderStatistics.ps1** script.
- *Folder To mailbox map path* equals the file name and path of the folder-to-mailbox .csv file that you'll create with this step. If you specify only the file name, the file will be generated in its current location.

Step 4: Create the public folder mailboxes on the Exchange 2013 server

Warning:
The name of the public folder mailboxes that you create must match the name of the **TargetMailbox** in the mapping file. You can edit the **TargetMailbox** names in the mapping file to match your organization's naming conventions.

1. Run the following command to create the first public folder mailbox on the Exchange 2013 Mailbox server. Public folder mailboxes contain the hierarchy information for a public folder, whereas the public folder contains the actual content. The first public folder mailbox that you create will be the master hierarchy mailbox. You need to create the first public folder mailbox in *HoldForMigration* mode.

```
New-Mailbox -PublicFolder <Name> -HoldForMigration:$true
```

2. Run the following command to create additional public folder mailboxes as needed based on the .csv file generated from the `PublicFoldertoMailboxMapGenerator.ps1` script. For example, if you open the .csv file, the public folders are named Mailbox1, Mailbox2, etc. If your last public folder is named Mailbox13, you'll need to create 13 public folder mailboxes.

If you need to create several public folder mailboxes, you can write a script to help automate the process. This example creates 25 public folder mailboxes.

```
$numberOfMailboxes = 25;
for($index =1 ; $index -le $numberOfMailboxes ; $index++)
{
    $PFMailboxName = "Mailbox"+$index;
 if($index -eq 1) {New-Mailbox -PublicFolder $PFMailboxName
-HoldForMigration:$true -
IsExcludedFromServingHiearchy:$true;}
else
{New-Mailbox -PublicFolder $PFMailboxName -
IsExcludedFromServingHierarchy:$true}
}
```

For detailed syntax and parameter information, see New-Mailbox.

Step 5: Start the migration request

The steps for migrating Exchange 2007 public folders are different from the steps for migrating Exchange 2010 public folders. Make sure that you follow the correct procedure.

Migrate Exchange 2007 public folders

1. Legacy system public folders such as OWAScratchPad and the schema-root folder subtree in Exchange 2007 won't be recognized by Exchange 2013 and will be treated as bad items. This will cause the migration to fail. As part of the migration request, you must specify a value for the BadItemLimit parameter. This value will vary depending on the number of public folder databases you have. The following commands will determine how many public folder databases you have and compute the BadItemLimit for the migration request.

```
$PublicFolderDatabasesInOrg = @(Get-PublicFolderDatabase)

$BadItemLimitCount = 5 + ($PublicFolderDatabasesInOrg.Count
-1)
```

2. From the Exchange 2013 Mailbox server, run the following command:

```
New-PublicFolderMigrationRequest -SourceDatabase (Get-
PublicFolderDatabase -Server <Source server name>) -CSVData
(Get-Content <Folder to mailbox map path> -Encoding Byte) -
BadItemLimit $BadItemLimitCount
```

3. To verify that the migration started successfully, run the following command.

```
Get-PublicFolderMigrationRequest | Get-
```

```
PublicFolderMigrationRequestStatistics -IncludeReport |
Format-List
```

You'll know that the command started successfully when the migration request reaches a status of *Queued* or *InProgress*. Depending on how much data is contained in the public folders, this command can take a long time to complete. If migration isn't being throttled due to the load on the destination server, the typical data copy rate can be 2 GB to 3 GB per hour.

4. You can periodically run the preceding command to check the status of the migration request. When the status reaches *AutoSuspended*, you can move to Step 6: Lock down the public folders on the legacy Exchange server for final migration (downtime required).

Migrate Exchange 2010 public folders

1. From the Exchange 2013 Mailbox server, run the following command:

```
New-PublicFolderMigrationRequest -SourceDatabase (Get-
PublicFolderDatabase -Server <Source server name>) -CSVData
(Get-Content <Folder to mailbox map path> -Encoding Byte)
```

2. To verify that the migration started successfully, run the following command.

```
Get-PublicFolderMigrationRequest | Get-
PublicFolderMigrationRequestStatistics -IncludeReport |
Format-List
```

You'll know that the command started successfully when the migration request reaches a status of *Queued* or *InProgress*. Depending on how much data is contained in the public folders, this command can take a long time to complete. If migration isn't being throttled due to the load on the destination server, the typical data copy rate can be 2 GB to 3 GB per hour.

3. You can periodically run the preceding command to check the status of the migration request. When the status reaches *AutoSuspended*, you can move to Step 6: Lock down the public folders on the legacy Exchange server for final migration (downtime required).

For detailed syntax and parameter information, see the following topics:

- New-PublicFolderMigrationRequest
- Get-PublicFolderDatabase
- Get-PublicFolderMigrationRequest
- Get-PublicFolderMigrationRequestStatistics

Step 6: Lock down the public folders on the legacy Exchange server for final migration (downtime required)

⚠**Warning:**
The amount of downtime required depends on how much new content was generated since the migration reached the *AutoSuspended* state. If a long time has passed between the

migration request reaching an *AutoSuspended* state and when you can finalization the migration, we recommend that you run the following command so you can synchronize the changes made since the original synchronization. This will reduce the amount of downtime required to finalize the migration.

```
Resume-PublicFolderMigrationRequest \PublicFolderMigration
```

Until this point in the migration, users have been able to access public folders. The next steps will log users off from the public folders and lock the folders while the migration completes its final synchronization. Users won't be able to access public folders during this process. Also, any mail sent to mail-enabled public folders will be queued and won't be delivered until the public folder migration is complete.

On the legacy Exchange server, run the following command to lock the legacy public folders for finalization.

```
Set-OrganizationConfig -
PublicFoldersLockedForMigration:$true
```

For detailed syntax and parameter information, see Set-OrganizationConfig.

If your organization has multiple public folder databases, you'll need to wait until public folder replication is complete to confirm that all public folder databases have picked up the `PublicFoldersLockedForMigration` flag and any pending changes users recently made to folders have converged across the organization. This may take several hours.

Step 7: Finalize the public folder migration (downtime required)

By default, when you run the **Set-PublicFolderMigrationRequest** cmdlet, it won't complete until you remove the *PreventCompletion* flag and resume the migration request.

```
Set-PublicFolderMigrationRequest -Identity
\PublicFolderMigration -PreventCompletion:$false
Resume-PublicFolderMigrationRequest -Identity
\PublicFolderMigration
```

For detailed syntax and parameter information, see Set-PublicFolderMigrationRequest and Resume-PublicFolderMigrationRequest.

Step 8: Test and unlock the public folder migration

After you finalize the public folder migration, you should run the following test to make sure that the migration was successful. This allows you to test the migrated public folder hierarchy before you switch to using Exchange 2013 public folders.

1. Run the following command to assign some test mailboxes to use any newly migrated public folder mailbox as the default public folder mailbox.

```
Set-Mailbox -Identity <Test User> -
DefaultPublicFolderMailbox <Public Folder Mailbox Identity>
```

2. Log on to Outlook 2007 or later with the test user identified in the previous step, and then perform the following public folder tests:
   a. View the hierarchy.
   b. Check permissions.
   c. Create and delete public folders.
   d. Post content to and delete content from a public folder.
3. If you run into any issues, see Roll back the migration later in this topic. If the public folder content and hierarchy is acceptable and functions as expected, run the following command to unlock the public folders for all other users.

```
Get-Mailbox -PublicFolder | Set-Mailbox -PublicFolder -
IsExcludedFromServingHierarchy $false
```

4. On the legacy Exchange server, run the following command to indicate that the public folder migration is complete:

```
Set-OrganizationConfig -PublicFolderMigrationComplete:$true
```

How do I know this worked?

In Step 2: Prepare for the migration, you were instructed to take snapshots of the public folder structure, statistics, and permissions before the migration began. The following steps will help verify that your public folder migration was successful by taking the same snapshots after the migration is complete. You can then compare the data in both files to verify success.

1. Run the following command to take a snapshot of the original source folder structure.

```
Get-PublicFolder -Recurse | Export-CliXML C:\PFMigration
\New_PFStructure.xml
```

2. Run the following command to take a snapshot of the public folder statistics such as item count, size, and owner.

```
Get-PublicFolderStatistics | Export-CliXML C:\PFMigration
\New_PFStatistics.xml
```

3. Run the following command to take a snapshot of the permissions.

```
Get-PublicFolder -Recurse | Get-
PublicFolderClientPermission | Select-Object Identity,User
```

```
-ExpandProperty AccessRights | Export-CliXML C:\PFMigration
\New_PFPerms.xml
```

Remove public folder databases from the legacy Exchange servers

After the migration is complete, and you have verified that your Exchange 2013 public folders are working as expected, you should remove the public folder databases on the legacy Exchange servers.

- For details about how to remove public folder databases from Exchange 2007 servers, see Removing Public Folder Databases.
- For details about how to remove public folder databases from Exchange 2010 servers, see Remove Public Folder Databases.

Roll back the migration

If you run into issues with the migration and need to reactivate your legacy Exchange public folders, perform the following steps:

⚠**Warning:**
After the migration is complete, any email to mail-enabled public folders, any change in public folder permissions or hierarchy, or content posted to public folders that occurred after the migration to Exchange 2013 will be lost. As a result, if you roll back the migration, you may lose public folder data or changes made on Exchange 2013 servers. To save this content, export the public folder content to a .pst file before you perform this procedure, and then import the .pst file after your rollback to the legacy public folders.

1. On the legacy Exchange server, run the following command to unlock the legacy Exchange public folders. This process may take several hours.

```
Set-OrganizationConfig -
PublicFoldersLockedForMigration:$False
```

2. On the Exchange 2013 server, run the following command to delete the public folder mailboxes.

```
Get-Mailbox -PublicFolder |
Where{$_.IsRootPublicFolderMailbox -eq $false} | Remove-
Mailbox -PublicFolder -Force -Confirm:$false
Get-Mailbox -PublicFolder | Remove-Mailbox -PublicFolder -
Force -Confirm:$false
```

3. On the legacy Exchange server, run the following command to set the `PublicFolderMigrationComplete` flag to `$false`.

```
Set-OrganizationConfig -
PublicFolderMigrationComplete:$False
```

Migrate legacy public folders to Office 365 and Exchange Online

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-25*

This topic describes how to migrate your public folders in a cutover or staged migration from Exchange Server 2010 Service Pack 3 (SP3) or Exchange 2007 SP3 RU10 to Office 365 or Exchange Online. For more information about migrating legacy public folders to Exchange Server 2013, see Migrate public folders to Exchange 2013 from previous versions.

Note:

This topic refers to the Exchange 2010 SP3 and Exchange 2007 SP3 RU10 servers as the *legacy Exchange server*. Also, the steps in this topic apply to both Exchange Online and Office 365. The terms may be used interchangeably in this topic.

We recommend that you don't use Outlook's PST export feature to migrate public folders to Office 365 or Exchange Online. Office 365 and Exchange online public folder mailbox growth is managed using an auto-split feature that splits the public folder mailbox when it exceeds size quotas. Auto-split can't handle the sudden growth of public folder mailboxes when you use PST export to migrate your public folders and you may have to wait for up to two weeks for auto-split to move the data from the primary mailbox. We recommend that you use the cmdlet-based instructions in this document to migrate public folders to Office 365 and Exchange Online. However, if you elect to migrate public folders using PST export, see the section Migrate Public Folders using PST files later in this topic.

You'll perform the migration by using the ***PublicFolderMigrationRequest** cmdlets, in addition to the following PowerShell scripts:

- `Export-PublicFolderStatistics.ps1` This script creates the folder name-to-folder size mapping file. You'll run this script on the legacy Exchange server.
- `Export-PublicFolderStatistics.psd1` This support file is used by the `Export-PublicFolderStatistics.ps1` script and should be downloaded to the same location.
- `PublicFolderToMailboxMapGenerator.ps1` This script creates the public folder-to-mailbox mapping file by using the output from the `Export-PublicFolderStatistics.ps1` script. You'll run

this script on the legacy Exchange server.

- `PublicFolderToMailboxMapGenerator.strings.psd1` This support file is used by the `PublicFolderToMailboxMapGenerator.ps1` script and should be downloaded to the same location.
- `Export-MailPublicFoldersForMigration.ps1` This script exports the mail-enabled public folder objects from the on-premises organization's Active Directory into a .csv file. You'll run this script on the legacy Exchange server.
- `Import-MailPublicFoldersForMigration.ps1` This script uses the .csv file generated by the `Export-MailPublicFoldersForMigration.ps1` script to import the mail-enabled public folder objects into Office 365 or Exchange Online. You'll run this script in Office 365.
- `MailPublicFolder.strings.psd1` This is a support file used by the Import and Export scripts and should be copied to the same location as the preceding scripts.

Step 1: Download the migration scripts provides details about where to download these scripts.

For additional management tasks related to public folders, see Public folder procedures.

What versions of Exchange are supported for migrating public folders to Office 365 and Exchange Online?

Exchange supports moving your public folders to Office 365 and Exchange Online from the following legacy versions of Exchange Server:

- Exchange 2010 SP3 or later
- Exchange 2007 SP3 RU10 or later

You can't migrate public folders directly from Exchange 2003. If you're running Exchange 2003 in your organization, you must move all public folder databases and replicas to Exchange 2007 SP3 RU10 or later. No public folder replicas can remain on Exchange 2003.

What do you need to know before you begin?

- In Office 365 and Exchange Online, you must be a member of the Organization Management role group. This role group is different from the permissions assigned to you when you subscribe to Office 365 or Exchange Online. For details about how to enable the Organization Management role group, see Manage role groups.
- In Exchange 2010, you must be a member of the Organization Management or Server Management RBAC role groups. For details, see Add Members to a Role Group.
- In Exchange 2007, you need to be assigned the Exchange Organization Administrator role or the Exchange Server Administrator role. In addition, you must be assigned the Public Folder Administrator role and local Administrators group for the target server. For details, see How to

Add a User or Group to an Administrator Role.

- On the Exchange 2007 server, upgrade to Windows PowerShell 2.0 and WinRM 2.0 for Windows Server 2008 x64 Edition.
- Before migration, if any public folder in your organization is greater than 2 GB, we recommend either deleting content from that folder or splitting it up into multiple public folders. If either of these options isn't feasible, we recommend that you do not move your public folders to Office 365 and Exchange Online.
- In Office 365 and Exchange Online, the default limit is 50 public folder mailboxes. Office 365 will allow you to automatically upgrade to 100 public folder mailboxes if you exceed this amount. If you need to exceed 100 public folder mailboxes, contact Office 365 support to request additional public folder mailboxes and your request will be evaluated.
- Before you migrate your public folders, we recommend that you first move all user mailboxes to Office 365 and Exchange Online. For details, see **Mailbox Migration to Exchange Online**.
- Outlook Anywhere must be enabled on the legacy Exchange server. For details about enabling Outlook Anywhere on Exchange 2010 servers, see Enable Outlook Anywhere. For details about enabling Outlook Anywhere on Exchange 2007 servers, see How to Enable Outlook Anywhere.
- You can't use the Exchange admin center (EAC) or the Exchange Management Console (EMC) to perform this procedure. On the legacy Exchange servers, you must use the Exchange Management Shell. For Exchange Online, you must use Exchange Online PowerShell. For more information, see **Connect to Exchange Online using remote PowerShell**.
- Before you begin, we recommend that you read this topic in its entirety as downtime is required for some steps.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Download the migration scripts

1. Download all four of the Microsoft Exchange 2013 public folder migration scripts.
2. Save the scripts to the local computer on which you'll be running PowerShell. For example, C:\PFScripts.
3. Download the following scripts from Microsoft Exchange 2013 Public Folders Directory Sync Support Scripts:
   a. `Export-MailPublicFoldersForMigration.ps1`
   b. `Import-MailPublicFoldersForMigration.ps1`
   c. `MailPublicFolder.strings.psd1`
4. Save the scripts to the same location you did for step 2. For example, C:\PFScripts.

Step 2: Prepare for the migration

Perform the following prerequisite steps before you begin the migration.

Prerequisite steps on the legacy Exchange server

1. On the legacy Exchange server, make sure that routing to the mail-enabled public folders that will exist in Office 365 or Exchange Online continues to work until all DNS caches over the Internet are updated to point to the Office 365 or Exchange Online DNS where your organization now resides. To do this, run the following command to configure an accepted domain with a well-known name that will properly route email messages to the Office 365 or Exchange Online domain.

```
New-AcceptedDomain -Name
"PublicFolderDestination_78c0b207_5ad2_4fee_8cb9_f373175b3f
99" -DomainName contoso.onmicrosoft.com -DomainType
InternalRelay
```

2. If the name of a public folder contains a backslash ****, the public folders will be created in the parent public folder when migration occurs. Before you migrate, we recommend that you rename any public folders that have a backslash in the name.
   a. In Exchange 2010, to locate public folders that have a backslash in the name, run the following command:

```
Get-PublicFolderStatistics -ResultSize Unlimited | Where
{$_.Name -like "*\*"} | Format-List Name, Identity
```

   b. In Exchange 2007, to locate public folders that have a backslash in the name, run the following command:

```
Get-PublicFolderDatabase | ForEach {Get-
PublicFolderStatistics -Server $_.Server | Where {$_.Name -
like "*\*"}}
```

   c. If any public folders are returned, you can rename them by running the following command:

```
Set-PublicFolder -Identity <public folder identity> -Name
<new public folder name>
```

3. Make sure there isn't a previous record of a successful migration. If there is, you'll need to set that value to `$false`. If the value is set to `$true`, the migration request will fail.
   a. The following example checks the public folder migration status.

```
Get-OrganizationConfig | Format-List
PublicFoldersLockedforMigration,
PublicFolderMigrationComplete
```

b. If the status of the *PublicFoldersLockedforMigration* or *PublicFolderMigrationComplete* properties is `$true`, run the following command to set the value to `$false`.

```
Set-OrganizationConfig -
PublicFoldersLockedforMigration:$false -
PublicFolderMigrationComplete:$false
```

⚠**Warning:**
After resetting these properties, you must wait for Exchange to detect the new settings. This may take up to two hours to complete.

4. For verification purposes at the end of migration, we recommend that you first run the following Shell commands on the legacy Exchange server to take snapshots of your current public folder deployment.
   a. Run the following command to take a snapshot of the original source folder structure.

```
Get-PublicFolder -Recurse | Export-CliXML C:\PFMigration
\Legacy_PFStructure.xml
```

   b. Run the following command to take a snapshot of public folder statistics such as item count, size, and owner.

```
Get-PublicFolderStatistics -ResultSize Unlimited | Export-
CliXML C:\PFMigration\Legacy_PFStatistics.xml
```

   c. Run the following command to take a snapshot of the permissions.

```
Get-PublicFolder -Recurse | Get-
PublicFolderClientPermission | Select-Object Identity,User
-ExpandProperty AccessRights | Export-CliXML C:\PFMigration
\Legacy_PFPerms.xml
```

Save the information from the preceding commands for comparison at the end of the migration.

For detailed syntax and parameter information, see the following topics:

- New-AcceptedDomain
- Get-PublicFolder
- Get-PublicFolderDatabase
- Set-PublicFolder
- Get-PublicFolderStatistics
- Get-PublicFolderClientPermission
- Get-OrganizationConfig
- Set-OrganizationConfig

Prerequisite steps in Office 365 or Exchange Online

1. Make sure there are no existing public folder migration requests. If there are, clear them. This step is a prerequisite and isn't required in all cases. It's only required if you think there may be an

existing migration request in the pipeline. In any case, the following command won't affect the new migration. The following example removes any existing public folder migration requests.

Important:

Before removing the migration request, it is important to understand why there was an existing one. You can run the following command to determine when a previous request was made and diagnose any problems that may have occurred. You may need to communicate with other administrators in your organization to determine why the change was made.

```
Get-PublicFolderMigrationRequest | Get-PublicFolderMigrationRequestStatistics -IncludeReport | Format-List
```

```
Get-PublicFolderMigrationRequest | Remove-PublicFolderMigrationRequest -Confirm:$false
```

2. Make sure no public folders or public folder mailboxes exist in Office 365.

Important:

If you do see public folders in Office 365 or Exchange Online, it is important to determine why they are there and who in your organization started a public folder hierarchy before removing the public folders and public folder mailboxes.

a. In Office 365 or Exchange Online PowerShell, run the following command to see if any public folders mailboxes exist.

```
Get-Mailbox -PublicFolder
```

b. If the command didn't return any public folder mailboxes, continue to Step 3: Generate the CSV files. If the command returned any public folders mailboxes, run the following command to see if any public folders exist:

```
Get-PublicFolder
```

c. If you have any public folders in Office 365 or Exchange Online, run the following PowerShell command to remove them.

```
Get-MailPublicFolder | where {$_.EntryId -ne $null}|
Disable-MailPublicFolder -Confirm:$false
Get-PublicFolder -GetChildren \ | Remove-PublicFolder -
Recurse -Confirm:$false
```

d. After the public folders are remvoved, run the following commands to remove all public folder mailboxes.

```
$hierarchyMailboxGuid = $(Get-
OrganizationConfig).RootPublicFolderMailbox.HierarchyMailbo
xGuid
Get-Mailbox -PublicFolder:$true | Where-Object
{$_.ExchangeGuid -ne $hierarchyMailboxGuid} | Remove-
```

```
Mailbox -PublicFolder -Confirm:$false
Get-Mailbox -PublicFolder:$true | Where-Object
{$_.ExchangeGuid -eq $hierarchyMailboxGuid} | Remove-
Mailbox -PublicFolder -Confirm:$false
```

For detailed syntax and parameter information, see the following topics:

- Get-PublicFolderMigrationRequest
- Remove-PublicFolderMigrationRequest
- Get-Mailbox
- Get-PublicFolder
- Get-MailPublicFolder
- Disable-MailPublicFolder
- Remove-PublicFolder
- Remove-Mailbox

Step 3: Generate the .csv files

1. On the legacy Exchange server, run the `Export-PublicFolderStatistics.ps1` script to create the folder name-to-folder size mapping file. This script must always be run by a local administrator. The file will contain two columns: **FolderName** and **FolderSize**. The values for the **FolderSize** column will be displayed in bytes. For example, **\PublicFolder01,10000**.

```
.\Export-PublicFolderStatistics.ps1  <Folder to size map
path> <FQDN of source server>
```

   - *FQDN of source server* equals the fully qualified domain name of the Mailbox server where the public folder hierarchy is hosted.
   - *Folder to size map path* equals the file name and path on a network shared folder where you want the .csv file saved. Later in this topic, you'll need to use the Exchange Online PowerShell to access this file. If you specify only the file name, the file will be generated in the current PowerShell directory on the local computer.
2. Run the `PublicFolderToMailboxMapGenerator.ps1` script to create the public folder-to-mailbox mapping file. This file is used to calculate the correct number of public folder mailboxes in Exchange Online.

```
.\PublicFolderToMailboxMapGenerator.ps1 <Maximum mailbox
size in bytes> <Folder to size map path> <Folder to mailbox
map path>
```

   - *Maximum mailbox size in bytes* equals the maximum size you want to set for the new public folder mailboxes. In Exchange Online, the maximum size of public folder mailboxes is 50 GB. We recommend that you set this setting to 15 GB so that each public folder mailbox has room for growth. If you have one single public folder that exceeds 2 GB, that public folder won't get added to the .csv file. The following are some options for how you can fix this issue:

- Before you run the script, delete public folder content to reduce the size to 2 GB or less.
- Before you run the script, split the public folder into multiple public folders that are each 2 GB or less.
- If the public folder is greater than 2 GB but no more than 30 GB, you can manually add it to the .csv file after you've run the script. The public folder will be created in Exchange Online.

Note:
If the public folder is greater than 30 GB and deleting content or splitting it into multiple public folders isn't feasible, we recommend that you don't move your public folders to Exchange Online.

- *Folder to size map path* equals the file path of the .csv file you created when running the `Export-PublicFolderStatistics.ps1` script.
- *Folder to mailbox map path* equals the file name and path of the folder-to-mailbox .csv file that you'll create with this step. If you specify only the file name, the file will be generated in the current PowerShell directory on the local computer.

Step 4: Create the public folder mailboxes in Exchange Online

Warning:
The names of the public folder mailboxes that you create must match the name of the **TargetMailbox** in the mapping file. You can edit the **TargetMailbox** names in the mapping file to match your organization's naming conventions.

1. Run the following command to create the primary public folder mailbox in Exchange Online. The first public folder mailbox that you create will be the primary hierarchy mailbox. You must create the first public folder mailbox in *HoldForMigration* mode. In addition, Exchange will automatically exclude the public folder mailboxes from the serving hierarchy so that the public folders won't be available to Exchange Online users.

```
New-Mailbox -PublicFolder <Name> -HoldForMigration:$true
```

2. Run the following command to create additional public folder mailboxes as needed based on the .csv file generated from the `PublicFoldertoMailboxMapGenerator.ps1` script. For example, if you open the .csv file, the public folders are named Mailbox1, Mailbox2, etc. If your last public folder is named Mailbox13, you'll need to create 13 public folder mailboxes. The maximum number of public folder mailboxes that you can create is 50.

If you need to create several public folder mailboxes, you can write a script to help automate the process. This example creates 15 public folder mailboxes.

```
$numberOfMailboxes = 15;
for($index =1 ; $index -le $numberOfMailboxes ; $index++)
{
```

```
    $PFMailboxName = "Mailbox"+$index;
 if($index -eq 1) {New-Mailbox -PublicFolder $PFMailboxName
-HoldForMigration:$true}
else
{New-Mailbox -PublicFolder $PFMailboxName}
}
```

For detailed syntax and parameter information, see New-Mailbox.

Step 5: Start the migration request

1. On the legacy Exchange server, run the following command to create the .xml file that will export the set of mail-enabled public folders from Active Directory.

```
.\Export-MailPublicFoldersForMigration.ps1
<mail_publicfolders.xml>
```

2. On the legacy Exchange server, get the following information that's needed to run the migration request:
   a. Find the `LegacyExchangeDN` of the user's account who is a member of the Public Folder Administrator role. This will be the same user whose credentials you need in step 3 of this procedure.

```
Get-Mailbox <PublicFolder_Administrator_Account> | Select-
Object LegacyExchangeDN
```

   b. Find the FQDN of any Mailbox server that has a public folder database.

```
Get-ExchangeServer <public folder server> | Select-Object -
Expand ExchangeLegacyDN
```

   c. Find the FQDN of the Outlook Anywhere host name. If you have multiple instances of Outlook Anywhere, we recommend that you select the instance that is either closest to the migration endpoint or the one that is closest to the public folder replicas in the legacy Exchange organization. The following command will find all instances of Outlook Anywhere:

```
Get-OutlookAnywhere | Format-Table
Identity,ExternalHostName
```

3. In Office 365 PowerShell, run the following commands to pass the information that you was returned in the previous step to variables that will then be used in the migration request.
   a. Pass the credential of a user who has administrative permissions on the legacy Exchange server into the variable `$source_credential`. The migration request that's run in Exchange Online will use this credential to gain access to your legacy Exchange servers to copy the content over.

```
$source_credential = Get-Credential <source_domain
\PublicFolder_Administrator_Account>
```

b. Use the `ExchangeLegacyDN` of the migration user on the legacy Exchange server that you found in step 2a and pass it into the variable `$source_remoteMailboxLegacyDN`.

```
$source_remoteMailboxLegacyDN = "<paste the value here>"
```

c. Use the `ExchangeLegacyDN` of the public folder server that you found in step 2b above and pass it into the variable `$source_remotePublicFolderServerLegacyFQDN`.

```
$source_remotePublicFolderServerLegacyFQDN = "<paste the
value here>"
```

d. Use the External Host Name of Outlook Anywhere that you found in step 2c above and pass it into the variable `$source_OutlookAnywhereExternalHostName`.

```
$source_OutlookAnywhereExternalHostName = "<paste the value
here>"
```

4. In Exchange Online PowerShell, run the following command to import the migration .xml file.

```
.\Import-MailPublicFoldersForMigration.ps1
<mail_publicfolders.xml>
```

5. Finally, in Exchange Online PowerShell, run the following command to start the migration request.

Note:

The authentication method in the following shell example needs to match your Outlook Anywhere settings, otherwise the command will fail.

```
New-PublicFolderMigrationRequest -OutlookAnywhereHostName:
$source_OutlookAnywhereExternalHostName -CSVData (Get-
Content <folder_mapping.csv> -Encoding Byte) -
RemoteCredential: $source_credential -
RemoteMailboxLegacyDN: $source_remoteMailboxLegacyDN -
RemoteMailboxServerLegacyDN:
$source_remotePublicFolderServerLegacyFQDN -
AuthenticationMethod Basic
```

Where the *<folder_mapping.csv>* file is the file that was generated in Step 3: Generate the .csv files.

6. To verify that the migration started successfully, in Exchange Online PowerShell, run the following command.

```
Get-PublicFolderMigrationRequest | Get-
```

```
PublicFolderMigrationRequestStatistics -IncludeReport |
Format-List
```

You'll know that the command started successfully when the migration request reaches a status of *Queued* or *InProgress*. Depending on how much data is contained in the public folders, this command can take a long time to complete. If migration isn't being throttled due to the load on the destination server or network performance, the typical data copy rate can be 2 GB to 3 GB per hour. The actual data speeds may vary depending on server configuration and network latency of the on-premises organization.

7. You can periodically run the preceding command to check the status of the migration request. When the status reaches *AutoSuspended*, you can move to Step 6: Lock down the public folders on the legacy Exchange server for final migration (downtime required).

For detailed syntax and parameter information, see the following topics:

- Get-Mailbox
- Get-ExchangeServer
- Get-OutlookAnywhere
- New-PublicFolderMigrationRequest
- Get-PublicFolderDatabase
- Get-PublicFolderMigrationRequest
- Get-PublicFolderMigrationRequestStatistics

Step 6: Lock down the public folders on the legacy Exchange server for final migration (downtime required)

⚠Warning:

The amount of downtime required depends on how much new content was generated since the migration reached the *AutoSuspended* state. If a long time has passed between the migration request reaching an *AutoSuspended* state and when you can finalize the migration, we recommend that you run the **Resume-PublicFolderMigrationRequest** cmdlet from Exchange Online PowerShell so you can synchronize the changes made since the original synchronization. This will reduce the amount of downtime required to finalize the migration.

```
Resume-PublicFolderMigrationRequest \PublicFolderMigration
```

Until this point in the migration, users have been able to access public folders. The next steps will log users off from the legacy public folders and lock the folders while the migration completes its final synchronization. Users won't be able to access public folders during this process. Also, any mail sent to mail-enabled public folders will be queued and won't be delivered until the public folder migration is complete.

On the legacy Exchange server, run the following command to lock the legacy public folders for finalization.

```
Set-OrganizationConfig -
```

```
PublicFoldersLockedForMigration:$true
```

For detailed syntax and parameter information, see Set-OrganizationConfig.

If your organization has multiple public folder databases, you'll need to wait until public folder replication is complete to confirm that all public folder databases have picked up the `PublicFoldersLockedForMigration` flag and any pending changes users recently made to folders have converged across the organization. This may take several hours.

Step 7: Finalize the public folder migration (downtime required)

By default, when you run the **Set-PublicFolderMigrationRequest** cmdlet in the Office 365 or Exchange Online PowerShell, it won't complete until you remove the *PreventCompletion* flag and resume the migration request.

```
Set-PublicFolderMigrationRequest -Identity
\PublicFolderMigration -PreventCompletion:$false
Resume-PublicFolderMigrationRequest -Identity
\PublicFolderMigration
```

For detailed syntax and parameter information, see the following topics:

Set-PublicFolderMigrationRequest

Resume-PublicFolderMigrationRequest

Set-OrganizationConfig

Step 8: Test and unlock the public folder migration

After you finalize the public folder migration, you should run the following test to make sure that the migration was successful. This allows you to test the migrated public folder hierarchy before you switch to using Office 365 or Exchange Online public folders.

1. In Office 365 or Exchange Online PowerShell, assign some test mailboxes to use any newly migrated public folder mailbox as the default public folder mailbox.

```
Set-Mailbox -Identity <Test User> -
DefaultPublicFolderMailbox <Public Folder Mailbox Identity>
```

2. Log on to Outlook 2007 or later with the test user identified in the previous step, and then perform the following public folder tests:
   - View the hierarchy.
   - Check permissions.
   - Create and delete public folders.

- Post content to and delete content from a public folder.

3. If you run into any issues, see Roll back the migration later in this topic. If the public folder content and hierarchy is acceptable and functions as expected, run the following Exchange Online PowerShell command to unlock the public folders for all other users.

```
Get-Mailbox -PublicFolder | Set-Mailbox -PublicFolder -IsExcludedFromServingHierarchy $false
```

Important:
Don't use the *IsExcludedFromServingHierarchy* parameter after initial migration validation is complete as this parameter is used by the automated storage management service for Exchange Online.

4. On the legacy Exchange server, run the following command to indicate that the public folder migration is complete:

```
Set-OrganizationConfig -PublicFolderMigrationComplete:$true
```

For detailed syntax and parameter information, see the following topics:

Set-Mailbox

Get-Mailbox

Set-OrganizationConfig

How do I know this worked?

In Step 2: Prepare for the migration, you were instructed to take snapshots of the public folder structure, statistics, and permissions before the migration began. The following steps will help verify that your public folder migration was successful by taking the same snapshots after the migration is complete. You can then compare the data in both files to verify success.

1. In Exchange Online PowerShell, run the following command to take a snapshot of the new folder structure.

```
Get-PublicFolder -Recurse | Export-CliXML  C:\PFMigration\Cloud_PFStructure.xml
```

2. In Exchange Online PowerShell, run the following command to take a snapshot of the public folder statistics such as item count, size, and owner.

```
Get-PublicFolderStatistics -ResultSize Unlimited | Export-CliXML  C:\PFMigration\Cloud_PFStatistics.xml
```

3. In Exchange Online PowerShell, run the following command to take a snapshot of the permissions.

```
Get-PublicFolder -Recurse | Get-
PublicFolderClientPermission | Select-Object Identity,User
-ExpandProperty AccessRights | Export-CliXML  C:
\PFMigration\Cloud_PFPerms.xml
```

Remove public folder databases from the legacy Exchange servers

After the migration is complete, and you have verified that your Exchange Online public folders are working as expected, you should remove the public folder databases on the legacy Exchange servers.

- For details about how to remove public folder databases from Exchange 2007 servers, see Removing Public Folder Databases.
- For details about how to remove public folder databases from Exchange 2010 servers, see Remove Public Folder Databases.

Roll back the migration

If you run into issues with the migration and need to reactivate your legacy Exchange public folders, perform the following steps.

⚠**Warning:**
If you roll your migration back to the legacy Exchange servers, you will lose any email that was sent to mail-enabled public folders or content that was posted to public folders after the migration. To save this content, you must export the public folder content to a .pst file and then import it to the legacy public folders when the rollback is complete.

1. On the legacy Exchange server, run the following command to unlock the legacy Exchange public folders. This process may take several hours.

```
Set-OrganizationConfig -
PublicFoldersLockedForMigration:$False
```

2. In Exchange Online PowerShell, run the following commands to remove all Exchange Online public folders.

```
$hierarchyMailboxGuid = $(Get-
OrganizationConfig).RootPublicFolderMailbox.HierarchyMailbo
xGuid
Get-Mailbox -PublicFolder:$true | Where-Object
{$_.ExchangeGuid -ne $hierarchyMailboxGuid} | Remove-
Mailbox -PublicFolder -Confirm:$false -Force
```

```
Get-Mailbox -PublicFolder:$true | Where-Object
{$_.ExchangeGuid -eq $hierarchyMailboxGuid} | Remove-
Mailbox -PublicFolder -Confirm:$false -Force
```

3. On the legacy Exchange server, run the following command to set the `PublicFolderMigrationComplete` flag to `$false`.

```
Set-OrganizationConfig -
PublicFolderMigrationComplete:$False
```

Migrate Public Folders to Office 365 by using Outlook PST export

We recommend that you don't use Outlook's PST export feature to migrate public folders to Office 365 or Exchange Online if your on-premises public folder hierarchy is greater than 30 GB. Office 365 online public folder mailbox growth is managed using an auto-split feature that splits the public folder mailbox when it exceeds size quotas. Auto-split can't handle the sudden growth of public folder mailboxes when you use PST export to migrate your public folders and you may have to wait for up to two weeks for auto-split to move the data from the primary mailbox. In addition, consider the following before using Outlook PST to export public folders to Office 365 or Exchange Online:

- Public folder permissions will be lost during this process. Capture the current permissions before migration and manually add them back once the migration is completed.
- If you use complex permissions or have many folders to migrate, we recommend that you use the cmdlet method for migration.
- Any item and folder changes made to the source public folders during the PST export migration will be lost. Therefore, we recommend that you use the cmdlet method if this export and import process will take a long time to complete.

If you still want to migrate your public folders by using PST files, follow these steps to ensure a successful migration.

1. Use the instructions in Step 1: Download the migration scripts to download the migration scripts. You only need to download the `PublicFolderToMailboxMapGenerator.ps1` file.
2. Follow step 2 of Step 3: Generate the .csv files to create the public folder-to-mailbox mapping file. This file is used to calculate the correct number of public folder mailboxes in Exchange Online.
3. Create the public folder mailboxes that you'll need based on the mapping file. For more information, see Create a public folder mailbox.
4. Use the **New-PublicFolder** cmdlet to create the top-most public folder in each of the public folder mailboxes by using the *Mailbox* parameter.
5. Export and import the PST files using Outlook.
6. Set the permissions on the public folders using the EAC. For more information, follow Step 3: Assign permissions to the public folder in the Set up public folders in a new organization topic.

Warning:

If you've already started a PST migration and have run into an issue where the primary mailbox is full, you have two options for recovering the PST migration:

1. Wait for the auto-split to move the data from the primary mailbox. This may take up to two weeks. However, all the public folders in a completely filled public folder mailbox won't be able to receive new content until the auto-split completes.
2. Create a public folder mailbox and then use the **New-PublicFolder** cmdlet with the *Mailbox* parameter to create the remaining public folders in the secondary public folder mailbox. This example creates a new public folder named PF201 in the secondary public folder mailbox.

```
New-PublicFolder -Name PF201 -Mailbox SecondaryPFMbx
```

Configure legacy on-premises public folders for a hybrid deployment

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-26*

In a hybrid deployment, your users can be in Exchange Online, on-premises, or both and your public folders are either in Exchange Online or on-premises. Public folders can only reside in one place, so you must decide whether your public folders will be in Exchange Online or on-premises. They can't be in both locations. Public folder mailboxes are synchronized to Exchange Online by the Directory Synchronization service. However, mail-enabled public folders aren't synchronized across premises.

This topic describes how to synchronize mail-enabled public folders when your users are in Exchange Online and your Exchange 2010 SP3 or Exchange 2007 SP3 RU10 public folders are on-premises.

Note:

This topic refers to the Exchange 2010 SP3 and Exchange 2007 SP3 RU10 servers as the *legacy Exchange server*.

You will sync your mail-enabled public folders using the following scripts, which are initiated by a Windows task that runs in the on-premises environment:

1. `Export-MailPublicFoldersForMigration.ps1` This script exports the mail-enabled public folder objects from the on-premises organization's Active Directory into a .XML file. You'll run this script on the legacy Exchange server.

2. `Import-MailPublicFoldersForMigration.ps1` This script uses the .XML file generated by the `Export-MailPublicFoldersForMigration.ps1` script to import the mail-enabled public folder objects into Exchange Online. You'll run this script in Exchange Online.
3. `MailPublicFolder.strings.psd1` This is a support file used by the Import and Export scripts and should be copied to the same location as the preceding scripts.

When you complete this procedure your on-premises and Exchange Online users will be able to access the same on-premises public folder infrastructure.

What hybrid versions of Exchange will work with public folders?

The following table describes the version and location combinations of user mailboxes and public folders that are supported. "Hybrid not applicable" is still a supported scenario, but is not considered a hybrid scenario since both the public folders and the users are residing in the same location.

	On-Premises Exchange 2007 or Exchange 2010 User Mailbox	**On-Premises Exchange 2013 User Mailbox**	**Exchange Online User Mailbox**
On-Premises Exchange 2007 or Exchange 2010 Public Folders	Hybrid not applicable	Hybrid not applicable	Supported
On-Premises Exchange 2013 Public Folders	Hybrid not applicable	Hybrid not applicable	Supported
Exchange Online Public Folders	Not supported	Supported	Hybrid not applicable

A hybrid configuration with Exchange 2003 public folders is not supported. If you're running Exchange 2003 in your organization, you must move all public folder databases and replicas to Exchange 2007 SP3 RU10 or later. No public folder replicas can remain on Exchange 2003.

What do you need to know before you begin?

1. These instructions assume that you have used the Hybrid Configuration Wizard to configure and synchronize your on-premises and Exchange Online environments and that the DNS records used for most users' AutoDiscover references an on-premises end-point. For more information, see

Hybrid Configuration wizard.

2. These instructions assume that Outlook Anywhere is enabled and functional on the on-premises legacy Exchange server(s). For information on how to enable Outlook Anywhere, see Outlook Anywhere.
3. Implementing legacy public folder coexistence for a hybrid deployment of Exchange with Office 365 may require you to fix conflicts during the import procedure. Conflicts can happen due to non-routable email address assigned to mail enabled public folders, conflicts with other users and groups in Office 365, and other attributes.
4. These instructions assume your Exchange Online organization has been upgraded to a version that supports public folders.
5. In Exchange Online, you must be a member of the Organization Management role group. This role group is different from the permissions assigned to you when you subscribe to Exchange Online. For details about how to enable the Organization Management role group, see Manage role groups.
6. In Exchange 2010, you must be a member of the Organization Management or Server Management RBAC role groups. For details, see Add Members to a Role Group
7. In Exchange 2007, you need to be assigned the Exchange Organization Administrator role or the Exchange Server Administrator role. In addition, you must be assigned the Public Folder Administrator role and local Administrators group for the target server. For details, see How to Add a User or Group to an Administrator Role
8. If you have Exchange Server 2007 running on Windows Server 2008 x64, then you must upgrade to Windows PowerShell 2.0 and WinRM 2.0 for Windows Server 2008 x64 Edition. If you have Exchange Server 2007 running on Windows Server 2003 x64, then you must upgrade to Windows PowerShell 2.0. For more information, see Update for Windows Server 2003 x64 Edition..
9. In order to access public folders cross-premises, users must upgrade their Outlook clients to the November 2012 Outlook public update or later.
    a. To download the November 2012 Outlook update for Outlook 2010, see Update for Microsoft Outlook 2010 (KB2687623) 32-Bit Edition.
    b. To download the November 2012 Outlook Update for Outlook 2007, see Update for Microsoft Office Outlook 2007 (KB2687404).
10. Outlook 2011 for Mac is not supported for cross-premises public folders. Users must be in the same location as the public folders to access them with Outlook 2011 for Mac. In addition, users whose mailboxes are in Exchange Online won't be able to access on-premises public folders using Outlook Web App.

Step 1: Make remote public folders discoverable

1. If your public folders are on Exchange 2010 or later servers, then you need to install the Client Access Server role on all mailbox servers that have a public folder database. This allows the Microsoft Exchange RpcClientAccess service to be running, which allows for all clients to access public folders. The client access role isn't required for Exchange 2007 public folder servers, and this step isn't necessary. For more information, see Install Exchange Server 2010. This step isn't necessary for Exchange 2007 public folders.

Note:

This server doesn't have to be part of the Client Access load balancing. For more information, see Understanding Load Balancing in Exchange 2010.

2. Create an empty mailbox database on each public folder server.

For Exchange 2010, run the following command. This command excludes the mailbox database from the mailbox provisioning load balancer. This prevents new mailboxes from automatically being added to this database.

```
New-MailboxDatabase -Server <PFServerName_with_CASRole> -
Name <NewMDBforPFs> -IsExcludedFromProvisioning $true
```

For Exchange 2007, run the following command:

```
New-MailboxDatabase -StorageGroup "<PFServerName>
\StorageGroup>" -Name <NewMDBforPFs>
```

Note:

We recommend that the only mailbox that you add to this database is the proxy mailbox that you'll create in step 3. No other mailboxes should be created on this mailbox database.

3. Create a proxy mailbox within the new mailbox database and hide the mailbox from the address book. The SMTP of this mailbox will be returned by AutoDiscover as the *DefaultPublicFolderMailbox* SMTP, so that by resolving this SMTP the client can reach the legacy exchange server for public folder access.

```
New-Mailbox -Name <PFMailbox1> -Database <NewMDBforPFs>

Set-Mailbox -Identity <PFMailbox1> -
HiddenFromAddressListsEnabled $true
```

4. For Exchange 2010, enable AutoDiscover to return the proxy public folder mailboxes. This step isn't necessary for Exchange 2007.

```
Set-MailboxDatabase <NewMDBforPFs> -RPCClientAccessServer
<PFServerName_with_CASRole>
```

5. Repeat the preceding steps for every public folder server in your organization.

Step 2: Download the scripts

1. Download the following files from Microsoft Exchange 2013 Public Folders Directory Sync Support Scripts:
   - `Export-MailPublicFoldersForMigration.ps1`
   - `Import-MailPublicFoldersForMigration.ps1`
   - `MailPublicFolder.strings.psd1`

2. Save the files to the local computer on which you'll be running PowerShell. For example, C:\PFScripts.

Step 3: Configure directory synchronization

The Directory Synchronization service doesn't synchronize mail-enabled public folders. Running the following two scripts will synchronize the mail-enabled public folders across premises.

1. On the legacy Exchange server, run the following command to create the .XML file that will export the set of mail-enabled public folders from Active Directory.

```
.\Export-MailPublicFoldersForMigration.ps1
<mail_publicfolders.xml>
```

Where `mail_publicfolders.xml` is the file name and path to a network shared folder that can be accessed from Exchange Online.

2. In Exchange Online PowerShell, run the following command to import the migration .XML file.

```
.\Import-MailPublicFoldersForMigration.ps1
<mail_publicfolders.xml>
```

Note:

We recommend that you run these scripts daily to synchronize your mail-enabled public folders.

Step 4: Configure Exchange Online users to access on-premises public folders

The final step in this procedure is to configure the Exchange online organization and to allow access to the legacy on-premises public folders.

Enable the exchange online organization to access the on-premises public folders. You will point to all of the proxy public folder mailboxes that you created in Step 1: Make remote public folders discoverable.

```
Set-OrganizationConfig -PublicFoldersEnabled Remote -
RemotePublicFolderMailboxes
PFMailbox1,PFMailbox2,PFMailbox3
```

Note:

You must wait until ActiveDirectory synchronization has completed to see the changes. This process can take up to 3 hours to complete. If you don't want to wait for the recurring synchronizations that occur every three hours, you can force directory synchronization at any time. For detailed steps to do force directory synchronization, see Force directory

synchronization.

How do I know this worked?

1. Log on to Outlook for a user who is in Exchange Online and perform the following public folder tests:
   - View the hierarchy.
   - Check permissions
   - Create and delete public folders.
   - Post content to and delete content from a public folder.

Configure legacy public folders where user mailboxes are on Exchange 2013 servers

Collaboration > Public folders > Public folder procedures >

Topic Last Modified: *2014-05-23*

In Exchange Server 2013, you'll need to perform this task to make sure that users can access public folders if you have Exchange 2013 users accessing Exchange 2010 or earlier public folders (also known as legacy public folders).

What do you need to know before you begin?

Users whose mailboxes are on Exchange Server 2013 won't be able to access legacy public folders from Outlook Web App or Outlook for Mac.

Step 1: Make the Exchange 2010 public folders discoverable

1. If your public folders are on Exchange 2010 or later servers, then you need to install the Client Access Server role on all mailbox servers that have a public folder database. This allows the Microsoft Exchange RpcClientAccess service to be running, which allows for all clients to access public folders. The client access role isn't required for Exchange 2007 public folder servers, and this step isn't necessary. For more information, see Install Exchange Server 2010.

Note:

This server doesn't have to be part of the Client Access load balancing. For more information, see Understanding Load Balancing in Exchange 2010.

2. Create an empty mailbox database on each public folder server.

For Exchange 2010, run the following command. This command excludes the mailbox database from the mailbox provisioning load balancer. This prevents new mailboxes from automatically being added to this database.

```
New-MailboxDatabase -Server <PFServerName_with_CASRole> -
Name <NewMDBforPFs> -IsExcludedFromProvisioning $true
```

For Exchange 2007, run the following command:

```
New-MailboxDatabase -StorageGroup "<PFServerName>
\StorageGroup>" -Name <NewMDBforPFs>
```

Note:

We recommend that the only mailbox that you add to this database is the proxy mailbox that you'll create in step 3. No other mailboxes should be created on this mailbox database.

3. Create a proxy mailbox within the new mailbox database and hide the mailbox from the address book. The SMTP of this mailbox will be returned by AutoDiscover as the *DefaultPublicFolderMailbox* SMTP, so that by resolving this SMTP the client can reach the legacy exchange server for public folder access.

```
New-Mailbox -Name <PFMailbox1> -Database <NewMDBforPFs>

Set-Mailbox -Identity <PFMailbox1> -
HiddenFromAddressListsEnabled $true
```

4. For Exchange 2010, enable AutoDiscover to return the proxy public folder mailboxes. This step isn't necessary for Exchange 2007.

```
Set-MailboxDatabase <NewMDBforPFs> -RPCClientAccessServer
<PFServerName_with_CASRole>
```

5. Repeat the preceding steps for every public folder server in your organization.

Step 2: Configure user mailboxes to access the legacy public folders

The final step in this procedure is to configure the user mailboxes to allow access to the legacy on-premises public folders.

Enable the Exchange Server 2013 on-premises users to access the legacy public folders. You will point to all of the proxy public folder mailboxes that you created in Step 1: Make the Exchange

2010 public folders discoverable. Run the following command from an Exchange 2013 server with the CU5 or higher update.

```
Set-OrganizationConfig -PublicFoldersEnabled Remote -
RemotePublicFolderMailboxes
ProxyMailbox1,ProxyMailbox2,ProxyMailbox3
```

Note:

You must wait until ActiveDirectory synchronization has completed to see the changes. This process may take several hours.

How do I know this worked?

Log on to Outlook for a user whose mailbox is on an Exchange Server 2013 CU5 or higher server and perform the following public folder tests:

1. Make sure that the Outlook client is running.
2. Hold down the CTRL key, and then right-click on the Outlook icon in the notification area on the right side of the Windows task bar.
3. Select **Test E-Mail Auto Configuration...**
4. Make sure the Text E-mail Auto Configuration tool returns the following information in the XML tab:
   - <PublicFolderInformation>
   - <SmtpAddress><SMTP Address for public folder mailbox</SmtpAddress>
   - </PublicFolderInformation>
5. In the Outlook client, peform the following tasks:
   - View the public folder hierarchy.
   - Check permissions.
   - Create and delete public folders.
   - Post content to and delete content from a public folder.

Create a public folder mailbox

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-02-24*

Before you can create a public folder, you must first create a public folder mailbox. Public folder mailboxes contain the hierarchy information plus the content for public folders. The first public folder mailbox you create will be the primary hierarchy mailbox, which contains the only writable copy of the hierarchy. Any additional public folder mailboxes you create will be secondary

mailboxes, which contain a read-only copy of the hierarchy.

Note:

For more information about the storage quotas and limits for public folders, see the following topics:

- For public folders in Office 365, see Exchange Online Limits.
- For public folders in on-premises Exchange Server 2013, see Limits for public folders.

For additional management tasks related to public folders in Exchange 2013, see Public folder procedures.

For additional management tasks related to public folders in Exchange Online, see **Public folder procedures in Office 365 and Exchange Online**.

What do you need to know before you begin?

- Estimated time to complete: less than 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to create a public folder mailbox

1. Navigate to **Public folders** > **Public folder mailboxes**, and then click **Add +**.
2. In **Public Folder Mailbox**, provide a name for the public folder mailbox.
3. Click **Save**.

Use the Shell to create a public folder mailbox

This example creates the primary public folder mailbox.

```
New-Mailbox -PublicFolder -Name MasterHierarchy
```

This example creates a secondary public folder mailbox. The only difference between creating the primary hierarchy mailbox and a secondary hierarchy mailbox is that the primary mailbox is the first one created in the organization. You can create additional public folder mailboxes for load balancing purposes.

```
New-Mailbox -PublicFolder -Name Istanbul
```

For detailed syntax and parameter information, see New-Mailbox.

How do you know this worked?

To verify that you have successfully created the primary public folder mailbox, run the following Shell command:

```
Get-OrganizationConfig | Format-List
RootPublicFolderMailbox
```

For detailed syntax and parameter information, see Get-OrganizationConfig.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Create a public folder

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-02-24*

Public folders are designed for shared access and provide an easy and effective way to collect, organize, and share information with other people in your workgroup or organization.

By default, a public folder inherits the settings of its parent folder, including the permissions settings.

Note:

For more information about the storage quotas and limits for public folders, see the following topics:

- For public folders in Office 365, see Exchange Online Limits.
- For public folders in on-premises Exchange Server 2013, see Limits for public folders.

For additional management tasks related to managing public folders, see Public folder procedures.

For additional management tasks related to public folders, see **Public folder procedures in Office 365 and Exchange Online**.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.
- You can't create a public folder unless you've first created a public folder mailbox. For more

information about how to create a public folder mailbox, see Create a public folder mailbox.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to create a public folder

When using the EAC to create a public folder, you'll only be able to set the name and the path of the public folder. To configure additional settings, you'll need to edit the public folder after it's created.

1. Navigate to **Public folders** > **Public folders**.
2. If you want to create this public folder as a child of an existing public folder, click the existing public folder in the list view. If you want to create a top-level public folder, skip this step.
3. Click **Add +**.
4. In **Public Folder**, type the name of the public folder.

Important:
Don't use a backslash (\) in the name when creating a public folder.

5. In the **Path** box, verify the path to the public folder. If this isn't the desired path, click **Cancel** and follow Step 2 of this procedure.
6. Click **Save**.

Use the Shell to create a public folder

This example creates a public folder named Reports in the path Marketing\2013.

```
New-PublicFolder -Name Reports -Path \Marketing\2013
```

Important:
Don't use a backslash (\) in the name when creating a public folder.

For detailed syntax and parameter information, see New-PublicFolder.

How do you know this worked?

To verify that you've successfully created a public folder, do the following:

- In the EAC, click **Refresh** to refresh the list of public folders. Your new public folder should be displayed in the list.
- In the Shell, run any of the following commands:

```
Get-PublicFolder -Identity \Marketing\2013\Reports |
Format-List
```

```
Get-PublicFolder -Identity \Marketing\2013 -GetChildren

Get-PublicFolder -Recurse
```

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Mail-enable or mail-disable a public folder

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-18*

Public folders are designed for shared access and provide an easy and effective way to collect, organize, and share information with other people in your workgroup or organization. Mail-enabling a public folder allows users to post to the public folder by sending an email message to it. When a public folder is mail-enabled additional settings become available for the public folder in the Exchange admin center (EAC), such as email addresses and mail quotas. In the Shell, before a public folder is mail-enabled, you use the **Set-PublicFolder** cmdlet to manage all of its settings. After the public folder is mail-enabled, you use the **Set-PublicFolder** and the **Set-MailPublicFolder** cmdlets to manage the settings.

If you want users on the Internet to send mail to a mail-enabled public folder, you need to set addition permissions using the **Add-PublicFolderClientPermission** cmdlet.

For additional management tasks related to managing public folders, see Public folder procedures.

For additional management tasks related to public folders, see **Public folder procedures in Office 365 and Exchange Online**.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- To ensure that users on the Internet can send e-mail messages to a mail-enabled public folder, the public folder needs to have at least the *CreateItems* access right granted to the Anonymous account. If you want to learn how to do this, check out Allow anonymous users to send email to a mail-enabled public folder.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration

permissions topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to mail-enable or mail-disable a public folder

1. Navigate to **Public folders** > **Public folders**.
2. In the list view, select the public folder that you want to mail-enable or mail-disable.
3. In the details pane, under **Mail settings**, click **Enable** or **Disable**.
4. A warning box displays asking if you are sure you want to enable or disable email for the public folder. Click **Yes** to continue.

If you want external users to send mail to this public folder, make sure you follow the steps in Allow anonymous users to send email to a mail-enabled public folder.

Use the Shell to mail-enable a public folder

This example mail-enables the public folder Help Desk.

```
Enable-MailPublicFolder -Identity "\Help Desk"
```

This example mail-enables the public folder Reports under the Marketing public folder, but hides the folder from address lists.

```
Enable-MailPublicFolder -Identity "\Marketing\Reports" -
HiddenFromAddressListsEnabled $True
```

If you want external users to send mail to this public folder, make sure you follow the steps in Allow anonymous users to send email to a mail-enabled public folder.

For detailed syntax and parameter information, see Enable-MailPublicFolder.

Use the Shell to mail-disable a public folder

This example mail-disables the public folder Marketing\Reports.

```
Disable-MailPublicFolder -Identity "\Marketing\Reports"
```

For detailed syntax and parameter information, see Disable-MailPublicFolder.

Allow anonymous users to send email to a mail-enabled public folder

You can use either Outlook or the Shell to set permissions on a public folder's Anonymous account. You can't use the EAC to set permissions on the Anonymous account.

Use Outlook to set permissions for the Anonymous account

1. Open Outlook using an account that's been granted Owner permissions on the email-enabled public folder you want anonymous users to send mail to.
2. Navigate to **Public folders - <user's name>**.
3. Navigate to the public folder you want to change.
4. Right-click on the public folder, click **Properties** and then select the **Permissions** tab.
5. Select the **Anonymous** account, select **Create items** under **Write**, and then click **OK**.

Use the Shell to set permissions for the Anonymous account

This example sets the `CreateItems` permission for the Anonymous account on the "Customer Feedback" mail-enabled public folder.

```
Add-PublicFolderClientPermission "\Customer Feedback" -
AccessRights CreateItems -User Anonymous
```

For detailed syntax and parameter information, see Add-PublicFolderClientPermission.

Update the public folder hierarchy

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Online*

Topic Last Modified: *2014-04-03*

You only need to update the public folder hierarchy if you want to manually invoke the hierarchy synchronizer and the mailbox assistant. Both these are invoked at least once every 24 hours for each public folder mailbox in the organization. The hierarchy synchronizer is invoked every 15 minutes if any users are logged on to a secondary mailbox through Microsoft Outlook or a Microsoft Exchange Web Services client.

For additional management tasks related to public folders in Exchange Online, see **Public folder procedures in Office 365 and Exchange Online**.

For additional management tasks related to public folders in Exchange Server 2013, see Public folder procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.
- You can't perform this procedure in the EAC. You must use the Shell.
- We recommend that when you run this command with the *InvokeSynchronizer* parameter, you use the *SuppressStatus* parameter. If you don't use this parameter in the command, the output will display status messages every 3 seconds for up to one minute. Until the minute passes, you can't use that instance of the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Update the public folder hierarchy

This example updates the public folder hierarchy on the public folder mailbox PF_marketing and suppresses the command's output.

```
Update-PublicFolderMailbox -Identity PF_marketing -
InvokeSynchronizer -SuppressStatus
```

This example updates all public folder mailboxes and suppresses the command's output.

```
Get-Mailbox -PublicFolder | Update-PublicFolderMailbox
InvokeSynchronizer -SuppressStatus
```

Remove a public folder

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-13*

You may need to remove public folders that are no longer being used in your organization. To help determine which public folders should be removed, see View statistics for public folders and public folder items.

For additional management tasks related to managing public folders, see Public folder procedures.

For additional management tasks related to public folders, see **Public folder procedures in Office 365 and Exchange Online**.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.
- You can't delete a mail-enabled public folder. Before you can delete it, you must first disable email for the public folder. For more information, see Mail-enable or mail-disable a public folder.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to remove a public folder

1. Navigate to **Public folders** > **Public folders**.
2. In the list view, select the public folder you want to delete, and then click **Delete** .
3. A warning box displays asking if you're sure you want to delete the public folder. Click **Yes** to continue.

Use the Shell to delete a public folder

This example deletes the public folder Help Desk\Resolved. This command assumes that the Resolved public folder doesn't have any subfolders.

```
Remove-PublicFolder -Identity "\Help Desk\Resolved"
```

This example tests the previous command without making any modifications.

```
Remove-PublicFolder -Identity "\HelpDesk\Resolved" -WhatIf
```

This example removes the public folder Marketing and all its subfolders because the command runs recursively.

```
Remove-PublicFolder -Identity "\Marketing" -Recurse:$True
```

For detailed syntax and parameter information, see Remove-PublicFolder.

Move a public folder mailbox to a different mailbox database

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-13*

You may need to move a public folder mailbox to a different mailbox database for load balancing purposes or for moving resources closer to their geographical location. Similar to moving a regular mailbox, you use the **MoveRequest** cmdlets to move a public folder mailbox. For more information about moving mailboxes, see Mailbox moves in Exchange 2013.

For additional management tasks related to public folders, see Public folder procedures.

What do you need to know before you begin?

- Estimated time to complete depends on the size of the public folder mailbox.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox move and migration permissions" section in the Recipients Permissions topic.
- You can't perform this procedure in the EAC. You can only perform this procedure in the Shell.
- Depending on the size of the public folder mailbox, the move may take several hours to complete. During that time, users won't be able to access the public folders. Users also won't have access to public folders for a brief period while the folder is in the "Completion in Progress" state.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a move request

The **New-MoveRequest** cmdlet queues the public folder mailbox into the Microsoft Exchange Mailbox Replication service queue. When the Microsoft Exchange Mailbox Replication service is

available, it will pick up the move request and begin moving the public folder mailbox. It will complete the entire request from beginning to end.

This example begins the move request for the public folder mailbox PF_SanFrancisco to the mailbox database MBX_DB01.

```
New-MoveRequest -Identity "PF_SanFrancisco" -TargetDatabase MBX_DB01
```

For detailed syntax and parameter information, see New-MoveRequest.

Create a move request to complete at a later time

During the final stage of the move request, when it's the `CompletionInProgress` phase, users will be locked out the public folder mailbox. If needed, you can suspend the move request before it reaches that phase and resume it at a time when users won't be impacted.

This example begins the move request for the public folder mailbox PF_SanFrancisco to the mailbox database MBX_DB01, and suspends it when the move request is ready to complete.

```
New-MoveRequest -Identity "PF_SanFrancisco" -TargetDatabase MBX_DB01 -SuspendWhenReadyToComplete
```

For detailed syntax and parameter information, see New-MoveRequest.

This example retrieves the status of the ongoing mailbox move for the public folder mailbox PF_SanFrancisco.

```
Get-MoveRequest -Identity "PF_SanFrancisco"
```

For detailed syntax and parameter information, see Get-MoveRequest.

When the move request reaches the status of Suspended, you can resume the request. This example resumes the move request for the public folder mailbox PF_SanFrancisco.

```
Resume-MoveRequest -Identity "PF_SanFrancisco"
```

For detailed syntax and parameter information, see Resume-MoveRequest.

How do you know this worked?

To verify that the move request was successfully created, run the following command:

```
Get-MoveRequestStatistics -Identity PF_SanFrancisco | Format-List Status
```

A status of `Completed` indicates that the move request was successful.

If the move was unsuccessful, you may need to restore the public folder. For more information, see Restore public folders and public folder mailboxes from failed moves.

Move a public folder to a different public folder mailbox

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-13*

If the content of a public folder mailbox begins to exceed your mailbox quotas, you may need to move public folders to a different public folder mailbox. There are a couple ways to do this. To move one or more public folders that don't contain subfolders, you can use the **PublicFolderMoveRequest** cmdlets. If you need to move an entire public folder branch (which includes the parent public folder and all subfolders), you can use the `Move-PublicFolderBranch.ps1` script that's available when you install Exchange 2013.

For additional management tasks related to public folders see Public folder procedures.

What do you need to know before you begin?

- Estimated time to complete this task depends on the size of the public folder.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the"Public folders" entry in the Sharing and collaboration permissions topic.
- You can't use the EAC to perform these procedures. You must use the Shell.
- If the folder you're moving has subfolders, those subfolders won't be moved by default. If you want to move a public folder and all its subfolders, use the **Move-PublicFolderBranch.ps1** script.
- Moving public folders only moves the physical contents of the public folder; it doesn't change the logical hierarchy.
- Depending on the size of the public folder and the amount of content it contains, the move may take several hours to complete. During that time, users will be able to access the public folders. However, users won't be able to access the public folders for a brief period while the folder is in the "Completion in Progress" state.
- You can perform only one public folder move request at a time. You must use the **Remove-PublicFolderMoveRequest** cmdlet to remove the request after it's complete.
- To check the status of an ongoing public folder move request, run the Get-PublicFolderMoveRequest cmdlet.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Move a single public folder

This example starts the move request for the public folder \CustomerEnagagements from the public folder mailbox DeveloperReports to DeveloperReports01

```
New-PublicFolderMoveRequest -Folders \DeveloperReports
\CustomerEngagements -TargetMailbox DeveloperReports01
```

For detailed syntax and parameter information, see New-PublicFolderMoveRequest.

Move multiple public folders

This example begins the move request for public folders under the \Dev public folder branch to the target public folder mailbox DeveloperReports01. This example doesn't move the public folder \Dev.

```
New-PublicFolderMoveRequest -Folders \Dev
\CustomerEngagements,\Dev\RequestsforChange,\Dev\Usability
-TargetMailbox DeveloperReports01
```

For detailed syntax and parameter information, see New-PublicFolderMoveRequest.

Move a branch of public folders

This example uses the `Move-PublicFolderBranch.ps1` script to move a branch of public folders. This starts the move request for the public folder \Dev and all its subfolders to the public folder mailbox DeveloperReports01. The script is located in the scripts folder and must be run from that location.

```
CD $env:ExchangeInstallPath\scripts
.\Move-PublicFolderBranch.ps1 -FolderRoot \Dev -
TargetPublicFolderMailbox DeveloperReports01
```

How do you know this worked?

To verify that the public folder move request was successful, run the following command:

```
Get-PublicFolderMoveRequest | Format-List Status
```

A status of `Completed` indicates that the move request was successful.

If the move request was unsuccessful, you may need to restore the public folder or its contents. For more information, see Restore public folders and public folder mailboxes from failed moves.

Restore public folders and public folder mailboxes from failed moves

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-13*

If a move request for a public folder or public folder mailbox fails, you can restore the folder or mailbox as long as the following conditions apply:

- **Failed public folder move** A soft-deleted copy of the public folder still exists in the source public folder mailbox and is still within the retention period.
- **Failed public folder mailbox move** A soft-deleted copy of the public folder mailbox still exists in the source mailbox database and is still within the mailbox retention period.

If the mailbox retention period has elapsed, you can recover an individual public folder mailbox from backup. You then extract data from the restored mailbox and copy it to a target folder or merge it with another mailbox. For more information, see Restore data using a recovery database.

For additional management tasks related to public folders, see Public folder procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox restore request" entry in the Recipients Permissions topic.
- You can't use the EAC to perform this procedure. You must use the Shell.
- To create a restore request, you must provide the values for the *DisplayName*, *LegacyDN*, or *MailboxGUID* parameters for the soft-deleted public folder mailbox.
- By default, dumpster folders will be restored along with regular folders. This can be prevented by using the *ExcludeDumpster* parameter.
- Restoring public folder mailboxes differs from restoring regular mailboxes in that folders won't

be created if they don't exist in the target mailbox. Missing folders will be displayed in a warning message at the end of restore request.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Restore a soft-deleted public folder

This example restores the public folder \Dev\CustomerEnagagements to the target public folder mailbox Development01.

```
New-MailboxRestoreRequest -SourceStoreMailbox Development -
SourceDatabase MBX_DB01 -TargetMailbox Development01 -
AllowLegacyDNMismatch -IncludeFolders \Dev
\CustomerEngagements
```

For detailed syntax and parameter information, see New-MailboxRestoreRequest.

Restore a soft-deleted public folder mailbox

This example restores the public folder mailbox PF_Singapore to the new public folder mailbox PF_Singapore_Restore.

```
New-MailboxRestoreRequest -SourceStoreMailbox PF_Singapore
-SourceDatabase MBX_DB01 -TargetMailbox
PF_Singapore_Restore -AllowLegacyDNMismatch
```

For detailed syntax and parameter information, see New-MailboxRestoreRequest.

View statistics for public folders and public folder items

Collaboration > Public folders > Public folder procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-14*

This topic explains how to retrieve statistics about a public folder, such as the display name, creation time, last user modified time, last user access, and item size. You can use this information to make decisions about deleting or retaining public folders.

Note:
In the Exchange admin center (EAC), you can view some of the quota and usage information for public folders by navigating to **Public Folders** > **Edit** > **Mailbox usage**. However, this information is incomplete, and we recommend that you use the Shell to view public folder statistics.

For additional management tasks related to managing public folders, see Public folder procedures.

For additional management tasks related to public folders, see **Public folder procedures in Office 365 and Exchange Online**.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.
- You can't use the EAC to retrieve public folder statistics.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to retrieve public folder statistics

This example returns the statistics for the public folder Marketing with a piped command to format the list.

```
Get-PublicFolderStatistics -Identity \Marketing | Format-List
```

Note:
The value for the *Identity* parameter must include the path to the public folder. For example, if the public folder Marketing existed under the parent folder Business, you would provide the following value: `\Business\Marketing`

For detailed syntax and parameter information, see Get-PublicFolderStatistics.

Use the Shell to view statistics for public folder items

You can view the following information about items within a public folder:

- Type of item
- Subject
- Last user modification time
- Last user access time
- Creation time
- Attachments
- Message size

You can use this information to make decisions about what actions to take for your public folders, such as which public folders to delete. For example, you may want to delete a public folder if the items haven't been accessed for over two years, or you may want to convert a public folder that's being used as a document repository to another client access application.

This example returns default statistics for all items in the public folder Pamphlets under the path \Marketing\2013. Default information includes item identity, creation time, and subject.

```
Get-PublicFolderItemStatistics -Identity "\Marketing\2013
\Pamphlets"
```

This example returns additional information about the items within the public folder Pamphlets, such as subject, last modification time, creation time, attachments, message size, and the type of item. It also includes a piped command to format the list.

```
Get-PublicFolderItemStatistics -Identity "\Marketing\2010
\Pamphlets" | Format-List
```

For detailed syntax and parameter information, see Get-PublicFolderItemStatistics.

Use the Shell to export the output of the Get-PublicFolderItemStatistics cmdlet to a .csv file

This example exports the output of the cmdlet to the PFItemStats.csv file that includes the following information for all items within the public folder \Marketing\Reports:

- Subject of the message (`Subject`)
- Date and time that the item was last modified (`LastModificationTime`)
- Whether the item has attachments (`HasAttachments`)
- Type of item (`ItemType`)
- Size of the item (`MessageSize`)

```
Get-PublicFolderItemStatistics -Identity "\Marketing
\Reports" | Select
Subject,LastModificationTime,HasAttachments,ItemType,Messag
eSize | Export-CSV C:\PFItemStats.csv
```

For detailed syntax and parameter information, see Get-PublicFolderItemStatistics.

Shared mailboxes

Exchange Server 2013 > Collaboration >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-13*

A shared mailbox is a mailbox that multiple users can use to read and send email messages. Shared mailboxes can also be used to provide a common calendar, allowing multiple users to schedule and view vacation time or work shifts.

Why set up a shared mailbox?

- Provides a generic email address (for example, info@contoso.com or sales@contoso.com), that customers can use to inquire about your company.
- Allows departments that provide centralized services to employees (for example, help desk, human resources, or printing services), to respond to employee questions.
- Allows multiple users to monitor and reply to email sent to an email address (for example, an address used specifically by the help desk).

What are shared mailboxes?

A shared mailbox is a type of user mailbox that doesn't have its own user name and password. As a result, users can't log into it them directly. To access a shared mailbox, users must first be granted Send As or Full Access permissions to the mailbox. Once that's done, users sign into their own mailboxes and then access the shared mailbox by adding it to their Outlook profile. In Exchange 2003 and earlier, shared mailboxes were just a regular mailbox to which an administrator could grant delegate access. Beginning in Exchange 2007, shared mailboxes became their own recipient type:

- **RecipientType:** UserMailbox
- **RecipientTypeDetails:** SharedMailbox

In previous version of Exchange, creating a shared mailbox was a multi-step process in which you had to use the Exchange Management Shell to complete some of the tasks. In Exchange 2013, you can use the Exchange admin center (EAC) to create a shared mailbox in one step. For details, see

Create a shared mailbox. In fact, the EAC has a feature area devoted entirely to shared mailboxes. Just navigate to **Recipients** > **Shared mailboxes** to view all the management tasks for shared mailboxes.

You can use the following permissions with a shared mailbox.

- **Full Access** The Full Access permission lets a user log into the shared mailbox and act as the owner of that mailbox. While logged in, the user can create calendar items; read, view, delete, and change email messages; create tasks and calendar contacts. However, a user with Full Access permission can't send email from the shared mailbox unless they also have Send As or Send on Behalf permission.
- **Send As** The Send As permission lets a user impersonate the shared mailbox when sending mail. For example, if Kweku logs into the shared mailbox Marketing Department and sends an email, it will look like the Marketing Department sent the email.
- **Send on Behalf** The Send on Behalf permission lets a user send email on behalf of the shared mailbox. For example, if John logs into the shared mailbox Reception Building 32 and sends an email, it look like the mail was sent by "John on behalf of Reception Building 32". You can't use the EAC to grant Send on Behalf permissions, you must use Set-Mailbox cmdlet with the *GrantSendonBehalf* parameter.

Converting shared mailboxes

In previous versions of Exchange, you could use a regular mailbox as a delegated mailbox. If you have delegated mailboxes, you can use the Shell to convert those delegate mailboxes to shared mailboxes. For details, see Convert a Mailbox.

Create a shared mailbox

Exchange Server 2013 > Collaboration > Shared mailboxes >

Applies to: *Office 365 Enterprise, Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-10-19*

Estimated time to complete: 5 minutes

Shared mailboxes makes it easy for a group of people in your company to monitor and send email from a common account, such as info@contoso.com or support@contoso.com. When a person in the group replies to a message sent to the shared mailbox, the email looks like it was sent by the shared mailbox, not from the individual user. You can read Open and use a shared mailbox to see some examples on how people are using shared mailboxes.

To learn more about shared mailboxes, see Shared mailboxes.

Important:

Creating a shared mailbox is now a one-step process using the Exchange admin center. Looking for how to create a shared mailbox in other versions of Exchange? Check out these topics:

- **Office 365 Small Business Admin** Create and use shared mailboxes.
- **Exchange 2010** Allow Mailbox Access.
- **Exchange 2007** How to Allow Mailbox Access.

Use the EAC to create a shared mailbox

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "User mailboxes" entry in the Recipients Permissions topic.

1. Go to **Recipients** > **Shared** > **Add +**.
2. Fill-in the required fields:
   - **Display name**
   - **Email address**
3. To grant Full Access or Send As permissions, click **Add +**, and then select the users you want to grant permissions to. You can use the **CTRL** key to select multiple users. Confused about which permission to use? See Which permission should you use? later in this topic.

Note:

The Full Access permission allows a user to open the mailbox as well as create and modify items in it. The Send As permission allows anyone other than the mailbox owner to send email from this shared mailbox. Both permissions are required for successful shared mailbox operation.

4. Click **Save** to save your changes and create the shared mailbox.

Use the EAC to edit shared mailbox delegation

1. Go to **Recipients** > **Shared** > **Edit** .
2. Click **Mailbox delegation**
3. To grant or remove Full Access and Send As permissions, click **Add +** or **Remove –** and then select the users you want to grant permissions to.

Note:

The Full Access permission allows a user to open the mailbox as well as create and modify items in it. The Send As permission allows anyone other than the mailbox owner to send email from this shared mailbox. Both permissions are required for successful shared mailbox operation.

4. Click **Save** to save your changes.

Use the Shell to create a shared mailbox

This example creates the shared mailbox Sales Department and grants Full Access and Send on Behalf permissions for the security group MarketingSG. Users who are members of the security

group will be granted the permissions to the mailbox.

Note:
This example assumes that you've already created the security group MarketingSG and that security group is mail-enabled. See Manage mail-enabled security groups.

```
New-Mailbox -Shared -Name "Sales Department" -DisplayName
"Sales Department" -Alias Sales | Set-Mailbox -
GrantSendOnBehalfTo MarketingSG | Add-MailboxPermission -
User MarketingSG -AccessRights FullAccess -InheritanceType
All
```

For detailed syntax and parameter information, see New-Mailbox.

Which permissions should you use?

You can use the following permissions with a shared mailbox.

- **Full Access** The Full Access permission lets a user log into the shared mailbox and act as the owner of that mailbox. While logged in, the user can create calendar items; read, view, delete, and change email messages; create tasks and calendar contacts. However, a user with Full Access permission can't send email from the shared mailbox unless they also have Send As or Send on Behalf permission.
- **Send As** The Send As permission lets a user impersonate the shared mailbox when sending mail. For example, if Kweku logs into the shared mailbox Marketing Department and sends an email, it will look like the Marketing Department sent the email.
- **Send on Behalf** The Send on Behalf permission lets a user send email on behalf of the shared mailbox. For example, if John logs into the shared mailbox Reception Building 32 and sends an email, it look like the mail was sent by "John on behalf of Reception Building 32". You can't use the EAC to grant Send on Behalf permissions, you must use Set-Mailbox cmdlet with the *GrantSendonBehalf* parameter.

More information

For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Email addresses and address books

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-26*

Recipients (which include users, resources, contacts, and groups) are any mail-enabled object in Active Directory to which Microsoft Exchange can deliver or route messages. For a recipient to send or receive email messages, the recipient must have an email address. Address books are the method by which users find each other in order to send email. There are many different methods for organizing address books. See Key terminology for detailed descriptions of address book features in Exchange Server 2013.

Key terminology

The following terms define the core components associated with email addresses and address books in Exchange 2013.

address book policies

Address book policies (ABPs) allow you to segment users into specific groups to provide customized views of your organization's global address list (GAL). When creating an ABP, you assign a GAL, an offline address book (OAB), a room list, and one or more address lists to the policy. You can then assign the ABP to mailbox users, providing them with access to a customized GAL in Outlook and Outlook Web App. The goal is to provide a simpler mechanism to accomplish GAL segmentation for on-premises organizations that require multiple GALs.

address lists

An address list is a subset of a GAL. Each address list is a collection of one or more types of mail-enabled recipients (for example, users, contacts, groups, public folders, conferencing, and other resources). You can use address lists to organize recipients and resources, making it easier for users to find the recipients and resources they need. Address lists are updated dynamically. Therefore, when new recipients are added to your organization, they're automatically added to the appropriate address lists.

email address policies

Email address policies generate the primary and secondary email addresses for your recipients so they can receive and send email. By default, Exchange contains an email address policy for every mail-enabled user.

hierarchical address books

The hierarchical address book (HAB) enables end users to look for recipients in their address book using an organizational hierarchy, such as seniority or management structure. Normally, users are limited to the default GAL and its associated recipient properties and the structure of the GAL often

doesn't accurately reflect the management or seniority relationships for recipients in your organization. Being able to customize an HAB that maps to your organization's unique business structure provides your users with an efficient method for locating internal recipients.

offline address books

An offline address book (OAB) is a copy of a collection of address lists that has been downloaded so that a Microsoft Outlook user can access the information it contains while disconnected from the server.

Email address and address book documentation

The following table contains links to topics that will help you learn about and manage email addresses and address books in Exchange 2013.

Topic	Description
Address lists	Learn more about address lists and GALs as methods for organizing your recipients for easy end user access.
Address book policies	Learn more about how to separate your address lists and GALs into separate virtual organizations.
Details templates	Learn more about customizing the address cards in Outlook.
Email address policies	Learn more about proxy email addresses to make recipients more discoverable.
Hierarchical address books	Learn more about how to customize the GAL and address lists to meet your organization’s unique business structure.
Offline address books	Learn more about providing users with offline access to your organization’s address lists.

Address book policies

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-19*

Global address list (GAL) segmentation (also known as *GAL segregation*) is the process whereby administrators can segment users into specific populations to provide customized views of their organization's GAL. Address book policies (ABPs) allow you to segment users into specific groups to provide customized views of your organization's global address list (GAL). When creating an ABP, you assign a GAL, an offline address book (OAB), a room list, and one or more address lists to the policy. You can then assign the ABP to mailbox users, providing them with access to a customized GAL in Outlook and Outlook Web App. The goal is to provide a simpler mechanism to accomplish GAL segmentation for on-premises organizations that require multiple GALs. .

Note:

ABPs are intended to optimize the GAL and address lists for each group of users, not make it difficult for them to see each other or to communicate with other users in your organization. ABPs create only a virtual separation of users from a directory perspective, not a legal separation.

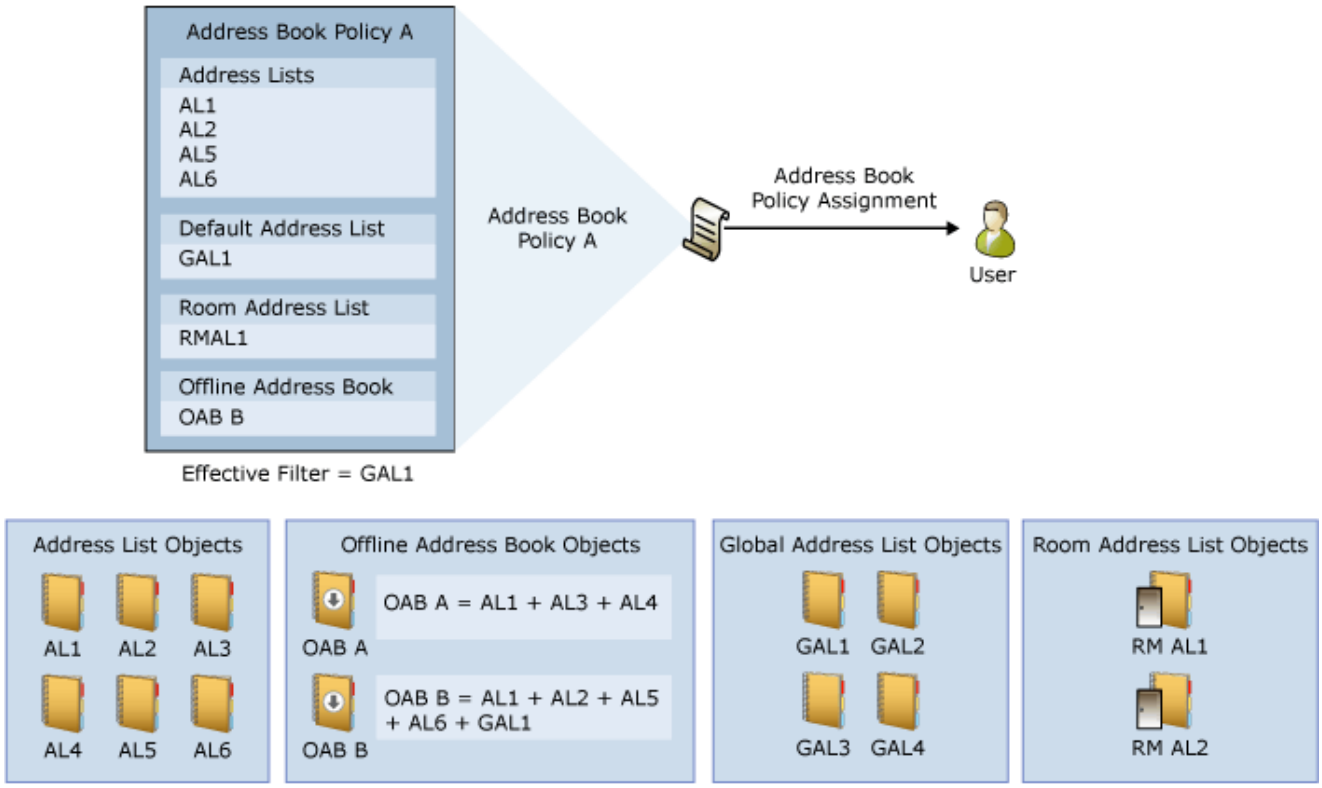
Contents

How ABPs Work

ABPs contain the following lists:

- One GAL
- One OAB
- One room list (for booking purposes)
- One or more address lists

In the following figure, Address Book Policy A consists of a subset of the various address objects that exist in the organization (shown in the bottom half of the figure). The resulting scope of an ABP is equal to that of the GAL contained in the policy, in this case GAL1. When the ABP is created and assigned to a user, the address objects in the ABP become the scope of the objects the user is able to view.



You can use the following methods to assign ABPs to individual mailbox users:

New or existing mailbox?	Shell
New	New-Mailbox cmdlet with the *AddressBookPolicy* parameter
Existing	Set-Mailbox cmdlet with the *AddressBookPolicy* parameter

ABPs take effect when a user's client application connects to a Client Access server in Exchange 2013. If you change the ABP, the updated ABP doesn't take effect until the user restarts or reconnects their client or until you restart the RPC Client Access servers on the Exchange 2013 Mailbox server.

Address Book Policy Routing Agent

In an Exchange organization that doesn't use ABPs, the following things occur when an email is created in Outlook or Outlook Web App and sent to another recipient in your Exchange organization:

1. The email address resolves. For example, if you type **kweku@contoso.com** in the **To** field, the SMTP email address would resolve to the user's display name **Kweku Ako-Adjei**.
2. You can view the other person's contact card. After the name resolves, you can double-click the user's name and view their contact information, such as office and phone number.

If you're using ABPs, and you don't want users in separate virtual organizations to view each other's potentially private information, you can turn on the Address Book Policy Routing agent. The ABP Routing agent is a Transport agent that controls how recipients are resolved in your organization.

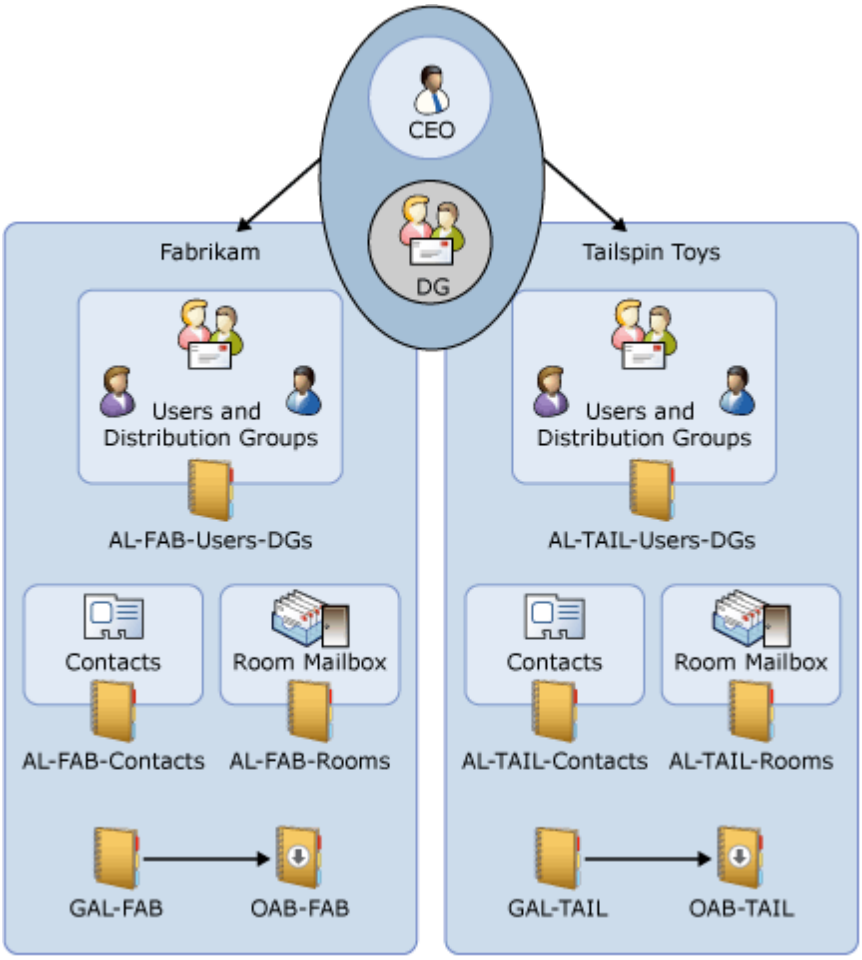
When the ABP Routing agent is installed and configured, users that are assigned to different GALs appear as external recipients in that they can't view external recipients' contact cards.

For details about how to turn on the ABP Routing agent in Exchange Online, see **Turn on address book policy routing**.

For details about how to turn on the ABP Routing agent in Exchange Server, see Install and configure the Address Book Policy Routing agent.

ABP Example

In the following diagram, Fabrikam and Tailspin Toys share the same Exchange organization and the same CEO. The CEO is the only employee common to both companies.



This configuration contains three ABPs:

- One contains Fabrikam employees and the CEO
- One contains Tailspin Toys employees and the CEO
- One contains only the CEO

The ABPs adhere to the following rules:

- The users in Tailspin Toys can only see Tailspin Toys employees and the CEO when they browse the GAL.
- The users in Fabrikam can only see Fabrikam employees and the CEO when they browse the GAL.
- The CEO can see all Fabrikam and Tailspin Toys employees when browsing the GAL.

- Users who view the CEO's group membership can see only groups that belong to the user's company. They won't see groups that exist in the other company.

Entourage, Outlook for Mac, and ABPs

ABPs won't function for Entourage users or Outlook for Mac users who are connected to their corporate network. When inside the corporate network, Entourage and Outlook for Mac clients connect directly to the global catalog server and query Active Directory directly instead of using the Client Access server. However, Outlook for Mac 2011 clients that connect from the Internet can use an OAB or Exchange Web Services (EWS). As a result, these clients can search the GAL based on the assigned ABP. To learn more about administering Outlook for Mac 2011, see Planning for Outlook for Mac 2011

For more information

Scenario: Deploying address book policies

Address book policy procedures in Exchange Online

Scenario: Deploying address book policies

Exchange Server 2013 > Email addresses and address books > Address book policies >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-12*

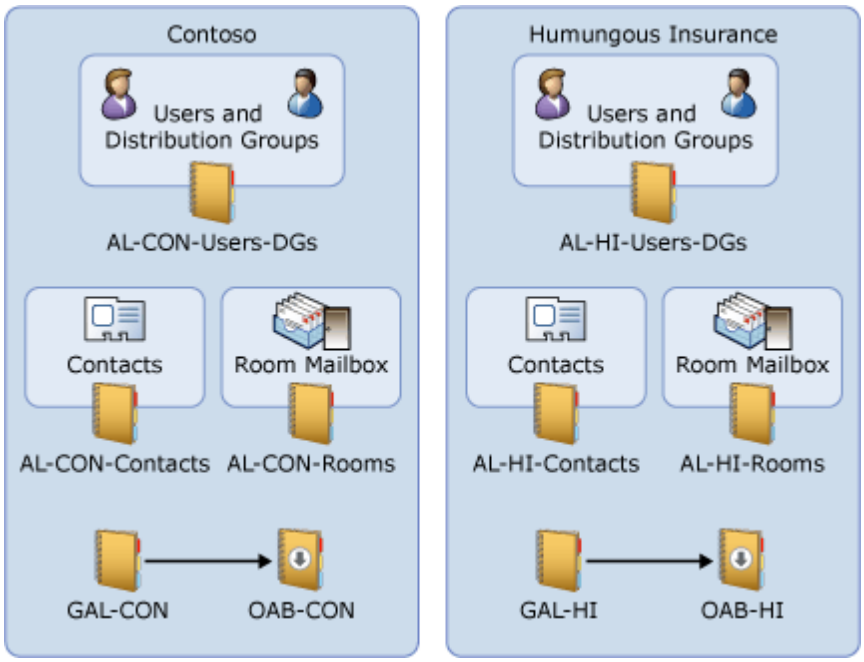
Deployment Scenarios

The following three scenarios describe possible deployment solutions for three different organization types. Although there are many more scenarios, the most popular ones are covered here. The address lists and global address lists (GALs) in these scenarios were created based on filters, such as Custom Attributes, that grouped the objects logically.

Scenario 1: Two separate companies - one Exchange organization

This scenario is applicable to government agencies, divisions, or departments that share

infrastructure, but no reporting chain and have no common employees. In addition, the divisions don't have any special security or privacy concerns. In this scenario, two address book policies (ABPs) are created where employees can only see members of the same organization when the view the GAL or look at membership of other distribution groups. In addition, no users will be members of distribution groups that span the entire organization.



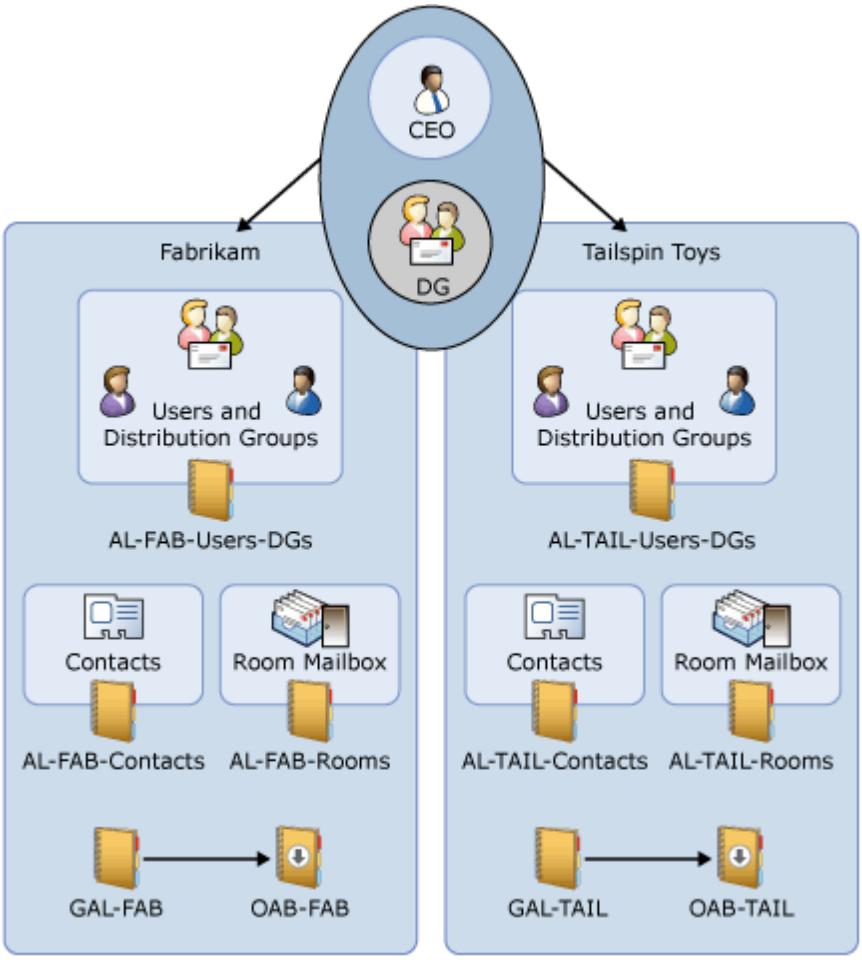
The Contoso and Humungous Insurance ABPs were created using the following address lists, global address lists, room lists, and OABs, which were created using a recipient filter that grouped the objects with a filter such as Custom Attribute. Because the two companies are separate without any interaction between the two, there aren't any address lists in common.

	Contoso	Humungous Insurance
Address Lists	AL_CON_Groups AL_CON_Users AL_CON_Contacts	AL_HI_Groups AL_HI_Users AL_HI_Contacts
Global address list	GAL_CON	GAL_HI
Room address list	AL_CON_Rooms	AL_HI_Rooms
Offline address book (OAB)	OAB_CON	OAB_HI

Scenario 2: Two companies sharing a CEO

In this scenario, Fabrikam and Tailspin Toys share the same Exchange organization and the same CEO. The CEO is the only common person between the two companies. This scenario requires three ABPs that have the following characteristics:

- The users in Tailspin Toys can only see Tailspin Toys users when they browse the GAL.
- The users in Fabrikam can only see Fabrikam users when they browse the GAL.
- In each company, there is a SeniorLeaders distribution group that includes the senior leaders of that company and the CEO.
- Users who look at the CEO's group membership will only see groups that belong to the user's company. They won't see groups not in their own company.
- Three ABPs are created: **Fab**, **Tail**, and **CEO**.



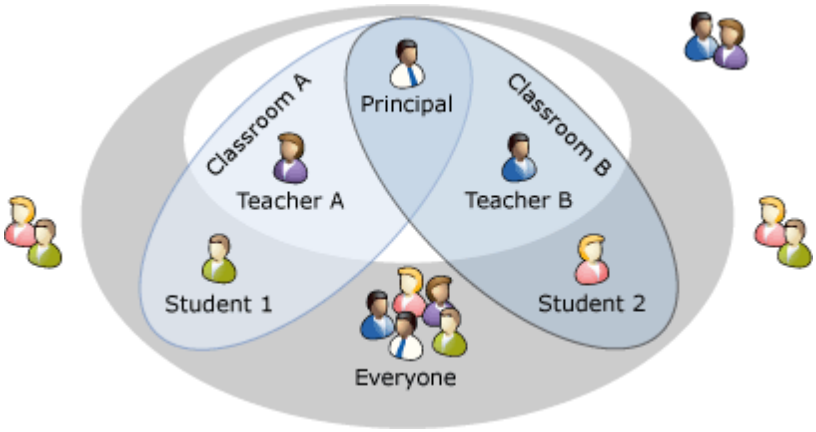
	Fabrikam	Tailspin Toys	CEO
Address lists	AL_FAB_Users_DGs AL_FAB_Contacts	AL_TAIL_Users_DGs AL_TAIL_Contacts	AL_FAB_Users_DGs AL_FAB_Contacts AL_TAIL_Users_DGs AL_TAIL_Contacts
Global address list	GAL_FAB	GAL_TAIL	Default GAL
Room address list	AL_FAB_Rooms	AL_TAIL_Rooms	Default All Rooms
Offline address book (OAB)	OAB_FAB	OAB_TAIL	Default OAB

When the CEO is added to the distribution groups in each organization and falls within the scope of each company's ABP, then the CEO becomes visible to each company. The CEO can create distribution groups that span both companies and will be visible within each company's GAL, but members of the distribution group will only be able to view the members of the group that are within their own organization.

Scenario 3: Education

This scenario is applicable to schools or universities where a division of class rooms is necessary to ensure the privacy of the students. The Education scenario has the following characteristics:

- Students in each class can only see other students in their class, their teacher, and the principal.
- Teachers can only students in their own classrooms.
- Teachers can see all other teachers and the principal.
- Distribution groups are created for each class's parents and the faculty.



	Students_ClassA	Teachers_ClassA	Principal
Address Lists	AL_ClassAAL_Principal	AL_ClassAAL_AllTeachersAL_AllGroupsAL_Principal	AL_ClassA AL_ClassB AL_AllTeachers AL_AllStudents AL_AllGroups
Global address list	GAL_StudentsClassA	GAL_TeachersClassA	GAL_Everyone
Room address list	AL_BlankRoom	AL_BlankRoom	Default All Rooms
Offline address book (OAB)	OAB_StudentsClassA	OAB_TeachersClassA	Default OAB

Considerations and best practices

Consider the following when using ABPs in your organization:

- For ABPs to work correctly, the user mailbox to which you apply the ABP must be on an Exchange 2010 SP3 or an Exchange 2013 server.
- Don’t run the Exchange 2010 Client Access server role on the global catalog server. Doing so results in Active Directory being used for Name Service Provider Interface (NSPI) instead of the Microsoft Exchange Address Book service. You can run Exchange 2013 server roles on a global catalog server and have ABPs work correctly, however we don’t recommend installing Exchange on a domain controller.
- You can't use hierarchical address books (HABs) and ABPs simultaneously. To learn more, see Hierarchical address books.
- Any user assigned an ABP should exist in their own GAL.
- If you allow client applications to access Active Directory directly through LDAP, they will bypass the logic built into ABPs. Because Outlook for Mac 2011 and Entourage 2008 use direct LDAP queries to access Active Directory, those client applications won’t function properly with ABPs if a domain controller or global catalog server is specified or provided to them by the Autodiscover service. Outlook for Mac 2011 can use EWS or a local OAB to access directory information. However, if Outlook for Mac 2011 can directly access an LDAP service, it will attempt to do so.
- The GAL used in an ABP must, at a minimum, contain all of the address lists, including the room address list, defined and specified in an ABP. Don’t create a GAL that contains fewer objects than any of the address lists in the same ABP.
- We recommend creating distribution groups that don't cross virtual organization boundaries. Creating distribution groups that contain members of multiple virtual organizations results in the following issues:
  - If group members request delivery or read receipts when sending mail to the distribution group, they’ll be able to see the email addresses of the group members in other virtual organizations
  - If an encrypted message is sent to the distribution group and some group members don’t have valid digital IDs, the sender will receive a warning message that includes the total number of members who don’t have valid IDs and a list of their email addresses. However, if some of those members without valid digital IDs are in a different organization than the sender, the warning message will include the correct count but won’t include the email addresses of the members in the other organization. As a result, the total count won’t match the list of member addresses.

For example, let’s say a distribution group contains five members total from two organizations, Agency A and Agency B. Three group members are from Agency A, and one of those members has as invalid digital ID. The other two members are from Agency B, and both of them have invalid digital IDs. If a member from Agency A sends an encrypted message to the distribution group, that member will receive a warning message stating that there are a total of three recipients without valid digital IDs. However, only the email address for the recipient from Agency A will be listed in

the warning message.

  - ABP's don't apply to the **Get-Group** cmdlets. Therefore, any user or process that is able to run **Get-Group** will see all members of any group they have access to.

We recommend that you modify the group management settings of the OWA Options so users can't use Outlook Web App to manage groups. To prevent users from using OWA Options to manage groups, exclude the users from the MyDistributionGroupMembership RBAC role. For details, see MyDistributionGroupMembership role.

  - If you allow users to use Outlook or Outlook Web App to manage groups, the group owners must have full visibility to the group membership list.

- All ABPs must contain a room address list. However, if your organization doesn't use room address lists, you can create a default empty room address list.
- Deploying ABPs doesn't prevent users in one virtual organization from sending email to users in another virtual organization. If you want to prevent users from sending email across organizations, we recommend that you create a transport rule. For example, to create a transport rule that prevents Contoso users from receiving messages from Fabrikam users, but still allows Fabrikam's senior leadership team to send messages to Contoso users, run the following Shell command:

```
New-TransportRule -Name "StopFabrikamtoContosoMail" -
FromMemberOf "AllFabrikamEmployees" -SentToMemberOf
"AllContosoEmployees" -DeleteMessage -ExceptIfFrom
seniorleadership@fabrikam.com
```

- If you want to enforce a feature similar to ABP in the Lync client, you can set the `msRTCSIP-GroupingID` attribute on specific user objects. For details, see PartitionByOU Replaced with msRTCSIP-GroupingID topic.

General deployment steps

Migrating from address list segmentation to ABPs

If your organization configured the Exchange 2007 address list segregation solution in place by using the instructions in the white paper Configuring Virtual Organizations and Address List Segregation in Exchange 2007, you should first migrate to Exchange Server 2010 using the steps outlined in Migrate to Exchange Server 2010 Address Book Policies from Exchange Server 2007 Address List Segregation. This procedure will require some down-time for your organization and you will therefore need to plan accordingly.

New deployment of ABPs

If your organization is deploying Exchange 2013 ABPs and hasn't used the Exchange 2007 address list segregation, you can use these instructions to deploy ABPs in your organization.

The steps in this section will walk you through Scenario 2: Two companies sharing a CEO. In this scenario, two companies (Fabrikam and Tailspin Toys) are separate but share a CEO and senior leadership team.

Step 1: Install and configure the Address Book Policy Routing agent

If you're using ABPs, and you don't want users in separate virtual organizations to view each other's potentially private information, you can turn on the Address Book Policy Routing agent. The Address Book Policy Routing agent is a Transport agent that runs on the Mailbox server that controls how recipients are resolved in the organization. When the Address Book Policy Routing Agent is installed and configured, users that are assigned different GALs appear as external recipients in that they can't view external recipients' contact cards.

For detailed instructions, see Install and configure the Address Book Policy Routing agent.

Step 2: Divide your virtual organizations

You'll need to develop a way to divide your organizations. We recommend using the CustomAttribute1-15 property on the mailboxes, contacts, and groups instead of the pre-canned conditional attributes such as Company, Department, or StateOrProvince to divide the virtual organizations for the following reasons:

- Not all recipient types of objects have precanned conditional attributes in Active Directory. For example, Distribution Group and Dynamic Distribution Group do not support company, department, or state attributes.
- Not all precanned conditional attributes are exposed in cmdlets for some recipients. For example, the *Company*, *department*, and *StateOrProvince* parameters are not available on the exposed in cmdlets for mail users, contacts, distribution groups, and mail-enabled public folders.
- Multiple cmdlets are required to segregate recipient when you use the pre-canned conditional attribute. For example, you need to run Set-User to tag *Company*, *Department*, *StateOrProvince* for a UserMailbox after you run **New-Mailbox** or **Set-Mailbox** cmdlets.
- The *CustomAttributeX* parameters are all exposed in the Set-* cmdlet for each recipient type, we can complete all segregation for that type via single Set- cmdlet
- CustomAttributeX attributes are explicitly reserved for customization of an organization and are entirely under the control of the organization administrators.

Another best practice to consider implementing when segregating your organization is to use company identifiers in the names of distribution groups and dynamic distribution groups. Exchange has a Group Naming policy feature that will automatically add a suffix or prefix to the name of the distribution group based on many attributes of the user creating the distribution group including the creator of the distribution group's Company, StateorProvince, Title, and CustomAttribute1 to CustomAttribute15. The group naming policy is especially important if you are allowing users to create their own distribution groups. For more information, see Create a distribution group naming policy.

Group naming policies don't apply to dynamic distribution groups, so you will need to manually

segregate them and manually apply a naming policy.

Step 3: Create the address lists, GALs, and OABs

When you create the address lists and global address lists do not use "IncludedRecipient" and "ConditionalX" parameters, such as ConditionalCompany and ConditionalCustomAttribute5. You should use a recipient filter instead. You must use the Shell to create recipient filters. For more information about Recipient Filters, see Recipient filtering on Edge Transport servers

In creating the ABP, you will create multiple address lists based on how you want your users to view the address lists in Outlook or Outlook Web App. This organization has four address lists:

- AL_FAB_Users_DGs
- AL_FAB_Contacts
- AL_TAIL_Users_DGs
- AL_TAIL_Contacts

This example creates the address list AL_TAIL_Users_DGs. The address list contains all users and distribution groups where CustomAttribute15 equals TAIL.

```
New-AddressList -Name "AL_TAIL_Users_DGs" -RecipientFilter
{((RecipientType -eq 'UserMailbox') -or (RecipientType -eq
"MailUniversalDistributionGroup") -or (RecipientType -eq
"DynamicDistributionGroup")) -and (CustomAttribute15 -eq
"TAIL")}
```

For more information about creating address lists by using recipient filters, see Create an address list by using recipient filters.

In order to create an ABP, you have to provide a room address list. If your organization doesn't have resource mailboxes such as room or equipment mailboxes, we suggest that you create a blank room address list. The following example creates a blank room address list because there are no room mailboxes in the organization.

```
New-AddressList -Name AL_BlankRoom -RecipientFilter {(Alias
-ne $null) -and ((RecipientDisplayType -eq
'ConferenceRoomMailbox') -or (RecipientDisplayType -eq
'SyncedConferenceRoomMailbox'))}
```

However, in this scenario, Fabrikam and Contoso both have room mailboxes. This example creates room list for Fabrikam by using a recipient filter where CustomAttribute15 equals FAB.

```
New-AddressList -Name AL_FAB_Room -RecipientFilter {(Alias
-ne $null) -and (CustomAttribute15 -eq "FAB")-and
(RecipientDisplayType -eq 'ConferenceRoomMailbox') -or
(RecipientDisplayType -eq 'SyncedConferenceRoomMailbox')}
```

The global address list used in an ABP must be a superset of the address lists. Do not create a GAL with fewer objects than exists in any or all of the address lists in the ABP. This example creates the global address list for Tailspin Toys that includes all of the recipients that exists in the address lists and room address list.

```
New-GlobalAddressList -Name "GAL_TAIL" -RecipientFilter
{(CustomAttribute15 -eq "TAIL")}
```

For more information, see Create a global address list.

When you create the OAB you should include the appropriate GAL when providing the *AddressLists* parameter of New- or Set-OfflineAddressBook to ensure no entry is unexpectedly missed. Basically, you can customize the set of entries that a user will see or reduce the download size of the OAB by specifying a list of AddressLists in AddressLists of New/Set-OfflineAddressBook. However, if you want users to see the full set of GAL entries in OAB, make sure that you include the GAL in the AddressLists.

This example creates the OAB for Fabrikam named OAB_FAB.

```
New-OfflineAddressBook -Name "OAB_FAB" -AddressLists
"GAL_FAB"
```

For more information, see Create an offline address book.

Step 4: Create the ABPs

After you've created all of the required objects you can then create the ABP. This example creates the ABP named ABP_TAIL.

```
New-AddressBookPolicy -Name "ABP_TAIL" -AddressLists
"AL_TAIL_Users_DGs"," AL_TAIL_Contacts" -OfflineAddressBook
"\OAB_TAIL" -GlobalAddressList "\GAL_TAIL" -RoomList
"\AL_TAIL_Rooms"
```

For more information, see Create an address book policy.

Step 5: Assign the ABPs to the mailboxes

Assigning the ABP to the user is the last step in the process. ABPs take effect when a user's application connects to the Microsoft Exchange Address Book service on the Client Access server. If the user is already connected to Outlook or Outlook Web App when the ABP is applied to their account, they will need to close and restart the client application before they can see their new address lists and GAL.

This example assigns ABP_FAB to all mailboxes where CustomAttribute15 equals "FAB".

```
Get-Mailbox -resultsize unlimited | where
{$_.CustomAttribute15 -eq "TAIL"} | Set-Mailbox -
AddressBookPolicy "ABP_TAIL"
```

For more information, see Assign an address book policy to mail users.

Address book policy procedures

Exchange Server 2013 > Email addresses and address books > Address book policies >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-11*

Create an address book policy

Assign an address book policy to mail users

Change the settings of an address book policy

Remove an address book policy

Install and configure the Address Book Policy Routing agent

Email addresses and address books > Address book policies > Address book policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-01-09*

The Address Book Policy Routing agent is a Transport agent that runs on the Mailbox server that controls how recipients are resolved in your organization. When the ABP Routing agent is installed and configured, users that are assigned to different GALs appear as external recipients in that they can't view external recipients' contact cards.

For additional management tasks related to ABPs, see Address book policy procedures.

Looking for the Exchange Online version of this topic? See **Turn on address book policy routing**.

What do you need to know before you begin?

- Estimated time to complete this task: 15 minutes.

- After the ABP Routing agent is installed and configured, it may take up to 30 minutes for email in the organization to be evaluated by agent.
- You can't use the EAC to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Install the ABP Routing agent

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport Agents" entry in the Mail flow permissions topic.

Install the ABP Routing agent by running the following command. This is the exact command and syntax you'll need to use.

```
Install-TransportAgent -Name "ABP Routing Agent" -
TransportAgentFactory
"Microsoft.Exchange.Transport.Agent.AddressBookPolicyRoutin
gAgent.AddressBookPolicyRoutingAgentFactory" -AssemblyPath
$env:ExchangeInstallPath\TransportRoles\agents
\AddressBookPolicyRoutingAgent
\Microsoft.Exchange.Transport.Agent.AddressBookPolicyRoutin
gAgent.dll
```

You'll get a warning that the Transport service needs to be restarted for your changes to take effect, but perform Step 2 first so you only have to restart the Transport service once.

For detailed syntax and parameter information, see Install-TransportAgent.

Step 2: Enable the Transport Routing agent

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport Agents" entry in the Mail flow permissions topic.

After the ABP Routing agent is installed, you need to enable it by running the following command:

```
Enable-TransportAgent "ABP Routing Agent"
```

For detailed syntax and parameter information, see Enable-TransportAgent.

Step 3: Restart the Transport service and verify the ABP Routing agent is installed and enabled

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport Agents" entry in the Mail flow permissions topic.

1. Restart the Transport service by running the following command.

```
Restart-Service MSExchangeTransport
```

2. After the service has restarted, verify that the ABP Routing agent is installed and enabled by running the following cmdlet.

```
Get-TransportAgent
```

If the ABP Routing agent is listed, the agent has been correctly installed.

For detailed syntax and parameter information, see Get-TransportAgent.

Step 4: Enable the ABP Routing agent

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Transport configuration" entry in the Mail flow permissions topic.

The final step in this process is to enable ABP routing for the organization. Run the following command.

```
Set-TransportConfig -AddressBookPolicyRoutingEnabled $true
```

For detailed syntax and parameter information, see Set-TransportConfig.

Create an address book policy

Email addresses and address books > Address book policies > Address book policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-19*

Address book policies (ABPs) allow you to segment users into specific groups to provide customized views of your organization's global address list (GAL). When creating an ABP, you assign a GAL, an offline address book (OAB), a room list, and one or more address lists to the

policy. You can then assign the ABP to mailbox users, providing them with access to a customized GAL in Outlook and Outlook Web App. The goal is to provide a simpler mechanism to accomplish GAL segmentation for on-premises organizations that require multiple GALs. To learn more about ABPs, see Address book policies.

For additional management tasks related to ABPs, see Address book policy procedures.

Interested in scenarios that use this procedure? See Scenario: Deploying address book policies.

What do you need to know before you begin?

- Estmated time to complete: Less than 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address book policies" entry in the Email address and address book permissions topic.
- By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.
- Creating an ABP for an organization is a multi-step process that requires planning. For more information, see Scenario: Deploying address book policies.
- You can't use the Exchange Administration Center (EAC) to create ABPs. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.
- Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to create an ABP

This example creates an ABP with the following settings:

- **Name:** All Fabrikam ABP
- **GAL:** All Fabrikam
- **OAB:** Fabrikam-All-OAB
- **Room list:** All Fabrikam Rooms
- **Address lists:** All Fabrikam, All Fabrikam Mailboxes, All Fabrikam DLs, and All Fabrikam Contacts

```
New-AddressBookPolicy -Name "All Fabrikam ABP" -
AddressLists "\All Fabrikam","\All Fabrikam
Mailboxes","\All Fabrikam DLs","\All Fabrikam Contacts" -
OfflineAddressBook \Fabrikam-All-OAB -GlobalAddressList
"\All Fabrikam" -RoomList "\All Fabrikam Rooms"
```

For detailed syntax and parameter information, see New-AddressBookPolicy.

For more information

Assign an address book policy to mail users

Assign an address book policy to mail users

Email addresses and address books > Address book policies > Address book policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-11*

After you create an address book policy (ABP), you must assign it to mailbox users. Users aren't assigned a default ABP when their user account is created. If you don't assign an ABP to a user, the global address list (GAL) for your entire organization will be accessible to the user through Outlook and Outlook Web App. To learn more, see Address book policies.

For additional management tasks related to ABPs, see Address book policy procedures.

Interested in scenarios that use this procedure? See Scenario: Deploying address book policies.

What do you need to know before you begin?

- Estmated time to complete: Less than 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address book policies" entry in the Email address and address book permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to assign an ABP to a mailbox user

1. Navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the user that you want to assign the policy to, and then click **Edit** .

3. Click **Mailbox features**.
4. In the **Address book policy** list, select the ABP that you want to apply to this user.
5. Click **Save**.

Use the EAC to assign an ABP to multiple mailbox users

1. Navigate to **Recipients** > **Mailboxes**.
2. In the list view use the Ctrl key to select multiple users.
3. In the details pane, click **More options**.
4. Under **Address Book Policy**, click **Update**.
5. In the **Select Address Book Policy** list, select the ABP that you want to apply to these users.
6. Click **Save**.

Use the Shell to assign an ABP to mailbox users

This example assigns the ABP All Fabrikam to the existing mailbox user joe@fabrikam.com.

```
Set-Mailbox -Identity joe@fabrikam.com -AddressBookPolicy
"All Fabrikam"
```

This example assigns the ABP ABP_EngineeringDepartment to all mailbox users whose `CustomAttribute11` value contains "Engineering Department".

```
Get-Mailbox -Filter {(CustomAttribute11 -like "Engineering
Department")} | Set-Mailbox -AddressBookPolicy
ABP_EngineeringDepartment
```

For detailed syntax and parameter information, see Set-Mailbox and Get-Mailbox.

Change the settings of an address book policy

Email addresses and address books > Address book policies > Address book policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-16*

After you create an address book policy (ABP), you can view or modify the name and the assigned global address list (GAL), offline address book (OAB), room list, and address lists.

For additional management tasks related to ABPs, see Address book policy procedures.

What do you need to know before you begin?

- Estmated time to complete: Less than 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address book policies" entry in the Email address and address book permissions topic.
- Creating an ABP for an organization is a multi-step process that requires planning. For more information, see Scenario: Deploying address book policies
- You can't use the Exchange Administration Center (EAC) to configure ABPs. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.
- Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Change the OAB, room list, and GAL for an ABP

This example changes the OAB, room list, and GAL that will be used by mailbox users who are assigned the ABP named All Fabrikam ABP.

```
Set-AddressBookPolicy -Identity "All Fabrikam ABP" -
OfflineAddressBook \Fabrikam-OAB-2 -GlobalAddressList "\All
Fabrikam GAL" -RoomList "\All Fabrikam Rooms"
```

Add an address list to an existing ABP

This example adds the address lists Contoso-Chicago and Contoso-Seattle to the ABP named ABPContoso.

```
Set-AddressBookPolicy -Identity "ABPContoso" -AddressLists
@{Add="Contoso-Chicago","Contoso-Seattle"}
```

Remove an address list from an ABP

This example removes the address lists Fabrikam-HR and Fabrikam-Finance from the ABP named ABPFabrikam.

```
Set-AddressBookPolicy -Identity "ABPFabrikam" -AddressLists
@{Remove="Fabrikam-HR","Fabrikam-Finance"}
```

Replace an address list in an ABP

This example replaces the address lists GovernmentAgencyA-ALL and GovernmentAgenceB-ALL with address lists GovernmentAgencyA-Atlanta and GovernmentAgencyA-Moscow for the ABP named GovernmentAgencyA.

```
Set-AddressBookPolicy -Identity GovernmentAgencyA -
AddressLists @{Remove="GovernmentAgencyA-
ALL","GovernmentAgencyB-All";Add="GovernmentAgencyA-
Atlanta","GovernmentAgencyA-Moscow"}
```

For more information

For detailed syntax and parameter information, see Set-AddressBookPolicy.

Remove an address book policy

Email addresses and address books > Address book policies > Address book policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-25*

Use this procedure to remove an address book policy (ABP).

For additional management tasks related to ABPs, see Address book policy procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address book policies" entry in the Email address and address book permissions topic.
- You can't remove an ABP if it's assigned to a user's mailbox or to a soft-deleted mailbox. To determine if an ABP is assigned to a user, run the following Shell command:

```
Get-Mailbox | Where $._AddressBookPolicy -eq <AddressBookPolicyName>
```

To determine if an ABP is assigned to a soft-deleted mailbox, run the following command:

```
Get-Mailbox -SoftDeletedMailbox | Where $._AddressBookPolicy -eq <AddressBookPolicyName>
```

- To remove an ABP from a user's mailbox, you can use the **Mailbox features** page of the mailbox's properties or the **Set-Mailbox** cmdlet.
- You can't use the Exchange Administration Center (EAC) to remove an ABP. You must use the

Shell.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to remove an ABP

This example removes the ABP ABP_TailspinToys.

```
Remove-AddressBookPolicy -Identity "ABP_TailspinToys"
```

For detailed syntax and parameter information, see Remove-AddressBookPolicy.

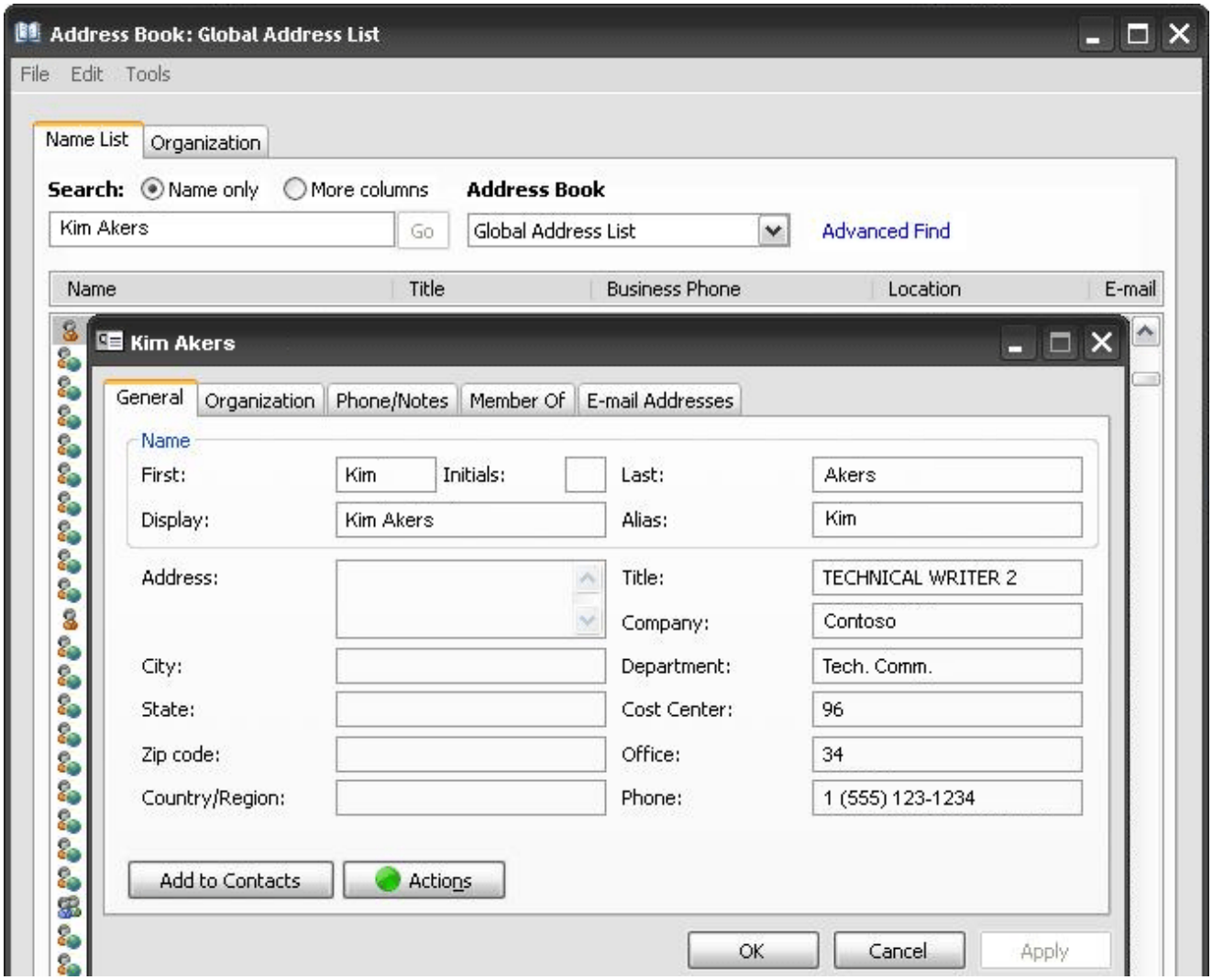
Details templates

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-12*

Details templates control the appearance of the object properties that are accessed by using address lists in an email client, such as Microsoft Outlook. For example, when a user opens an address list in Outlook, the properties of the recipients in that address list are presented as defined by the details template that exists in your Exchange organization. The following figure illustrates the properties of the recipient Kim Akers as it appears in Outlook 2013. Using the Details Templates Editor in Exchange 2013 Toolbox, you can modify the organization of and content within the various objects that appear on this property page.

Default details template as viewed from Outlook 2013



You can use the default details template or you can customize the template to better suit the needs of your users. The objects can be customized by changing field sizes, adding or removing fields, adding or removing tabs, and rearranging fields. The layout of these templates may vary by language. Use the Details Templates Editor to customize the following Outlook objects:

- Contacts
- Users
- Groups
- Mailbox agents
- Public folders
- Search dialog boxes

For More Information

Customize details templates

Restore a details template to the default configuration

Customize details templates

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-25*

Use the Details Templates Editor to customize the client-side graphical user interface (GUI) presentation of object properties that are accessed by using address lists in Microsoft Outlook. For example, when a user opens an address list in Outlook, the properties of a particular object are presented as defined by the details template in the Exchange organization. The objects can be customized by changing field sizes, adding or removing fields, adding or removing tabs, and rearranging fields. The layout of these templates may vary by language.

What do you need to know before you begin?

- There is no undo option in the details template editor. If you make a mistake, you will need to revert back to the previous version. For more information, see Restore a details template to the default configuration.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Details templates" entry in the Email address and address book permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Customize the details template

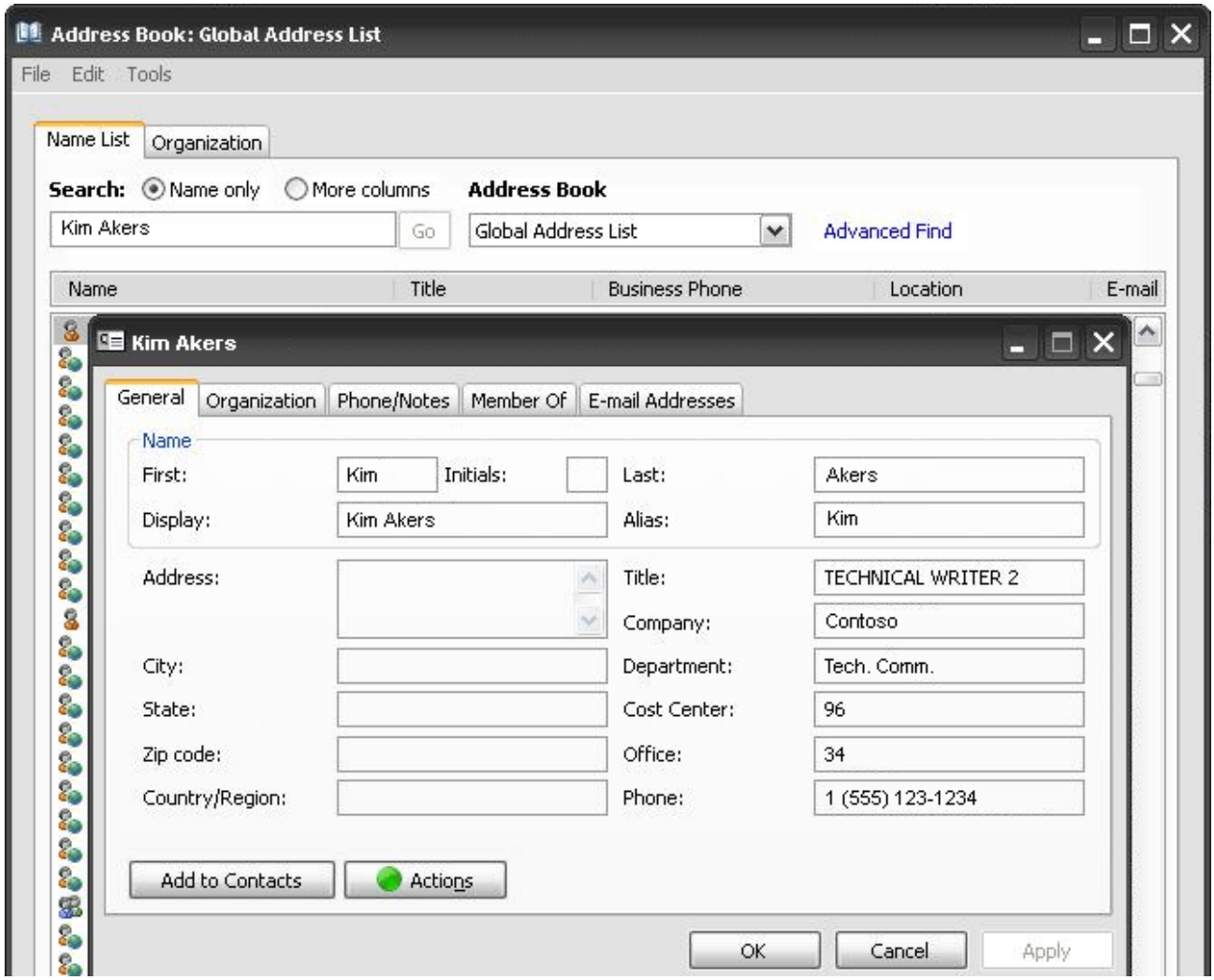
1. In the **Exchange Toolbox**, click **Details Templates Editor**, and then in the action pane, click **Open Tool**.
2. In the console tree of the Details Templates Editor, click **Details Templates**.

In the details pane, the following columns are displayed:

- **Language** This column lists the language in which the template was created.
- **Template Type** This column lists the type of template that you can customize.
- **Identity** This column lists the unique identity of the template.
- **Created** This column lists the date and time that the template was created.
- **Modified** This column lists the date and time that the template was last modified.

3. To edit a template, click the template you want, and then, in the action pane, click **Edit**. For example, the **English (United States)** contacts details template is shown in the following figure.

Default details template as viewed from Outlook 2013



4. After you click **Edit**, there are several tasks you can perform to customize a details template:
   - To move an object in the designer pane, select the object, and then drag it to its new location on the template. As you move the object, you are provided with alignment lines.
   - To change a label's text, select the label in the design pane. In the properties pane, type the new text in the **Text** box. To create keyboard shortcuts, you can use the ampersand (&) symbol. Place the ampersand (&) before the letter that you want to use as the shortcut.
   - To change the size of an object, select the object, and then drag the sizing handles until the object is the shape and size you want.
   - To delete an object, select the object, and then press the DELETE key.

Note:

The Details Templates Editor doesn't contain an **Undo** button, nor can you use a keyboard shortcut to undo an action. To undo an addition you made to the template, you must use the DELETE key. To undo a deletion, you must reapply the setting. You can also revert to the original settings by exiting the Details Templates Editor without saving your changes. If you want to undo changes after you have saved, you can restore the template. When you restore a template, all customization is lost, and the template is restored to its original configuration. For more information about how to restore the details template, see Restore a details template to the default configuration.

   - To add an "Edit" text boxes, list boxes, multi-valued drop-down boxes, or multi-valued list boxes, in the toolbox pane, drag the object to the design pane. Set the attribute of the object by clicking the attribute drop-down box in the properties pane and then selecting the attribute

that will be used by Exchange.

Note:

You must link the object to an attribute for it to be used by Exchange. In addition, the attribute determines the content that is displayed to the end user in Outlook. If you don't select an attribute, a random attribute is selected automatically.

- To add a group box, drag the object to the design pane. Then, in the properties pane, type a name in the **Text** box. Use group boxes to group similar objects.
- To add a tab to the template, right-click an existing tab, and then click **Add Tab**. A blank tab appears. To name the tab, type the name in the **Text** box in the properties pane.
- To remove a tab from the template, right-click the tab, and then click **Remove Tab**. A warning appears. Click **OK** to confirm that you want to remove the tab.
- To change the tabbing order of the objects on a tab so that users can use the TAB key to navigate the objects in the order you want, select the object in the design pane. Then, in the properties pane, use the **TabIndex** box to change the order.

Note:

To make sure that users cannot use the TAB key to access the labels of an object (for example **Name** or **Alias**), change the order of the labels so that they are last in the tabbing order.

5. To save changes to the details template, on the **File** menu, click **Save**.
6. To close the template, on the **File** menu, click **Exit**.

Restore a details template to the default configuration

Exchange Server 2013 > Email addresses and address books > Details templates >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-12*

The Details Templates Editor doesn't contain an **Undo** button, nor can you use a keyboard shortcut to undo an action. To undo an addition you made to the template, you must use the DELETE key. To undo a deletion, you must reapply the setting. You can also revert to the original settings by exiting the Details Templates Editor without saving your changes. If you want to undo changes after you have saved, you can restore the template. When you restore a template, all customization is lost, and the template is restored to its original configuration.

This topic explains how to use the Exchange 2013 Toolbox or the Exchange Management Shell to restore a details template to its default configuration.

To learn more about details templates, see Details templates.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Details templates" entry in the Email address and address book permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Exchange Toolbox to restore a details template to its default configuration

1. Click **Start** > **All Programs** > **Microsoft Exchange Server 2013** > **Exchange Toolbox**.
2. In **Exchange Toolbox**, click **Details Templates Editor**, and then, in the action pane, click **Open Tool**.
3. In the **Details Templates Editor**, in the details pane, select the template you want to restore, and then in the action pane, click **Restore**.
4. Click **Yes** to confirm that you want to restore the template to its original state. All customization will be lost.

Use the Shell to restore a details template to its default configuration

This example restores the United States English contacts details template.

```
Restore-DetailsTemplate -Identity "en-US\Contact"
```

For detailed syntax and parameter information, see Restore-DetailsTemplate.

Email address policies

Exchange Server 2013 > Email addresses and address books >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

Recipients (which include users, resources, contacts, and groups) are any mail-enabled object in Active Directory to which Microsoft Exchange can deliver or route messages. For a recipient to send or receive email messages, the recipient must have an email address. Email address policies generate the primary and secondary email addresses for your recipients so they can receive and send email.

By default, Exchange contains an email address policy for every mail-enabled user. This default policy specifies the recipient's alias as the local part of the email address and uses the default accepted domain. The local part of an email address is the name that appears before the at sign (@). However, you can change how your recipients' email addresses will display. For example, you can specify that the addresses display as *firstname.lastname*@contoso.com.

Furthermore, if you want to specify additional email addresses for all recipients or just a subset, you can modify the default policy or create additional policies. For example, the user mailbox for David Hamilton can receive email messages addressed to hdavid@mail.contoso.com and hamilton.david@mail.contoso.com.

Looking for management tasks related to email address policies? See Email address policy procedures.

Behaviors of recipient policies

Exchange applies a policy to all recipients that match the recipient filtering criteria:

- The recipient policy functionality is divided into two features: email address policies and accepted domains. For more information about accepted domains, see Accepted domains.
- When you run the **Update-EmailAddressPolicy** cmdlet in the Exchange Management Shell, the recipient object is updated with the email address policy.
- Each time a recipient object is modified and saved, Exchange enforces the correct application of the email address criteria and settings. When an email address policy is modified and saved, all associated recipients are updated with the change. In addition, if a recipient object is modified, that recipient's email address policy membership is reevaluated and enforced.

Creating email address policies

When creating an email address policy, you can use the following email address types:

- **Precanned SMTP email address**. *Precanned* SMTP email addresses are commonly used email address types that are provided for you.
- **Custom SMTP email address**. If you don't want to use one of the precanned SMTP email addresses, you can specify a custom SMTP email address.

When creating a custom SMTP email address, you can use the variables in the following table to specify alternate values for the local part of the email address.

Variable	Value

%g	Given name (first name)
%i	Middle initial
%s	Surname (last name)
%d	Display name
%m	Exchange alias
%*x*s	Uses the first *x* letters of the surname. For example, if *x* = 2, the first two letters of the surname are used.
%*x*g	Uses the first *x* letters of the given name. For example, if *x* = 2, the first two letters of the given name are used.

- **Non-SMTP email address**. The following types of non-SMTP email addresses are supported:
  - EX (Legacy DN Proxy Address Prefix DisplayName)
  - X.500
  - X.400
  - MSMail
  - CcMail
  - Lotus Notes
  - Novell GroupWise
  - Exchange Unified Messaging proxy address (EUM proxy address)

Important:
In Exchange, all non-SMTP email addresses are considered custom addresses. Exchange doesn't provide unique dialog boxes or property pages for X.400, GroupWise, or Lotus Notes email address types. If you add a non-SMTP custom email address, you must have the appropriate dynamic-link library (DLL) files. If you don't provide the appropriate DLL files, you won't be able to create a customized email address policy. The following error will be logged in Event Viewer: "The email address description object in the Microsoft Exchange directory for the 'SADF' address type on 'i386' machines are missing."

For detailed instructions about how to create an email address policy, see the following topics:

Create an Email Address Policy

Create an email address policy by using recipient filters

Email address policy procedures

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-13*

Create an Email Address Policy

Create an email address policy by using recipient filters

Edit an email address policy

Remove an email address policy

Create an Email Address Policy

Email addresses and address books > Email address policies > Email address policy procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-10*

For a recipient to receive or send email messages, the recipient must have an email address. Email address policies generate the primary and secondary email addresses for your recipients (which include users, contacts, and groups) so they can receive and send email.

When creating an email address policy, you can use the following email address types:

- **Precanned SMTP email address**. *Precanned* SMTP email addresses are commonly used email address types that are provided for you.
- **Custom SMTP email address**. If you don't want to use one of the precanned SMTP email addresses, you can specify a custom SMTP email address.

When creating a custom SMTP email address, you can use the variables in the following table to specify alternate values for the local part of the email address.

Variable	Value
%g	Given name (first name)
%i	Middle initial
%s	Surname (last name)
%d	Display name
%m	Exchange alias
%*x*s	Uses the first *x* letters of the surname. For example, if *x* = 2, the first two letters of the surname are used.
%*x*g	Uses the first *x* letters of the given name. For

	example, if *x* = 2, the first two letters of the given name are used.

- **Non-SMTP email address**. The following types of non-SMTP email addresses are supported:
  - EX (Legacy DN Proxy Address Prefix DisplayName)
  - X.500
  - X.400
  - MSMail
  - CcMail
  - Lotus Notes
  - Novell GroupWise
  - Exchange Unified Messaging proxy address (EUM proxy address)

Important:

In Exchange, all non-SMTP email addresses are considered custom addresses. Exchange doesn't provide unique dialog boxes or property pages for X.400, GroupWise, or Lotus Notes email address types. If you add a non-SMTP custom email address, you must have the appropriate dynamic-link library (DLL) files. If you don't provide the appropriate DLL files, you won't be able to create a customized email address policy. The following error will be logged in Event Viewer: "The email address description object in the Microsoft Exchange directory for the 'SADF' address type on 'i386' machines are missing."

For detailed instructions about how to create an email address policy, see the following topics:

Create an Email Address Policy

Create an email address policy by using recipient filters

What do you need to know before begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Email address policies" entry in the Email addresses and address books topic.
- Before an SMTP address domain can be used in an email address policy, you must configure an accepted domain. To learn more, see Accepted domains.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Warning:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to create an email address policy

1. Navigate to **Mail flow** > **Email address policies**, and then click **Add +**.
2. In **Email Address Policy**, complete the following fields:
   - **Policy name**
   - **Email address format**
   - **Specify the types of recipients this email address will apply to**
3. Click **Add a rule** to further restrict the recipients that this policy will apply to. This creates a Boolean **And** statement.

Caution:
If you apply too many rules, it's possible to restrict the email address policy to the point that it doesn't contain any users.

4. Click **Preview recipients the policy applies to** to view the recipients that policy will apply to.
5. Click **Save** to save your changes and create the policy.
6. You'll get a warning that the email address policy won't be applied until you update it. After it's created, select it, and then, in the details pane, click **Apply**.

Use the Shell to create an email address policy

This example creates an email address policy that includes mailbox users in the Southeast offices who will have email addresses that include their last name combined with the first two letters of their first name.

```
New-EmailAddressPolicy -Name "southeast offices" -
IncludedRecipients MailboxUsers -ConditionalStateorProvince
"Georgia","Alabama","Louisiana" -
EnabledEmailAddressTemplates "SMTP:%s%
2g@southeast.contoso.com"
```

For detailed syntax and parameter information, see New-EmailAddressPolicy.

Create an email address policy by using recipient filters

Email addresses and address books > Email address policies > Email address policy procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

You can use the Shell to create an email address policy by using recipient filters. To learn more about email address policies, see Email address policies.

For additional management tasks related to email address policies, see Email address policy procedures.

What do you need to know before begin?

- Estimated time to complete: 5 minutes.
- To use the *RecipientFilter* parameter to create a custom filter, you must specify a string for the filter. The Shell uses OPath for the filtering syntax. OPath is a querying language designed to query object data sources.

Important:

If you use a recipient filter to create or edit an email address policy, you can't use the Exchange Administration Center (EAC) to edit the email address policy. You must use the Shell. For detailed syntax and parameter information, see Set-EmailAddressPolicy.

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Email address policies" entry in the Email addresses and address books topic.
- Before an SMTP address domain can be used in an email address policy, you must configure an accepted domain. For more, see Accepted domains.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Warning:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to create an email address policy by using recipient filters

To create an email address policy by using recipient filters, use the following syntax.

```
New-EmailAddressPolicy -Name <String> -RecipientFilter
<String>
```

This example creates an email address policy that applies to all executives and for which the local part of the email address consists of the first two letters of their first name and their entire last name.

```
New-EmailAddressPolicy -Name 'Execs' -
```

```
EnabledEmailAddressTemplates 'SMTP:%2g%s@contoso.com' -
RecipientFilter {((RecipientType -eq 'UserMailbox') -and
(Title -like 'executive'))}
```

For detailed syntax and parameter information, see New-EmailAddressPolicy.

Edit an email address policy

Email addresses and address books > Email address policies > Email address policy procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-10*

Email address policies generate the primary and secondary email addresses for your recipients (which include users, contacts, and groups) so they can receive and send email.

For additional management tasks related to email address policies, see Email address policy procedures.

What do you need to know before begin?

- Estimated time to complete: 5 minutes.
- You can't use the Exchange Administration Center (EAC) to edit an email address policy if the policy was created by using the Shell.
- If the email address policy was created using a recipient filter, you must use the Shell to edit the email address policy. For more information, see Create an email address policy by using recipient filters.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Email address policies" entry in the Email addresses and address books topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

⚠**Warning:**
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to change the recipients that the policy

applies to

1. Navigate to **Mail flow** > **Email address policies**.
2. In the list view, select the email address policy you want to change, and then click **Edit** ✎.
3. In **Email Address Policy**, click **Apply to** and modify the settings.

Use the EAC to change the email address policy's priority

A user can have multiple proxy email addresses for the same email account (for example, ayla@exchange.mail.contoso.com or ayla@contoso.com). These email addresses can then be applied by priority. For example, consider this scenario: you have two email address policies, and you assign them priorities of 1 and 2. If you create another policy, it will automatically be assigned a priority of 3. However, let's say you have two policies, and you specify that one of them is priority 1, but the other policy was assigned a default priority of 2 when it was created. In this case, the next policy you create will, by default, become the priority 2 policy. The previous priority 2 policy will be assigned a priority of 3.

1. Navigate to **Mail flow** > **Email address policies**.
2. Select the email address policy for which you want to change the priority, and then click **Increase priority** ↑ or **Decrease priority** ↓.

Use the Shell to edit an email address policy

This example edits the email address policy South East Offices that currently includes recipients in Georgia, Alabama, and Louisiana to also include recipients in Texas.

```
Set-EmailAddressPolicy -Identity "South East Offices" -
ConditionalStateorProvince
"Georgia","Alabama","Louisiana","Texas"
```

Note:

Although the email address policy is already applied to recipients in Georgia, Alabama, and Louisiana, you must include them in the parameter because the parameter overwrites values; it doesn't append values to existing ones.

For detailed syntax and parameter information, see Set-EmailAddressPolicy.

Remove an email address policy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-13*

By default, Exchange contains an email address policy that specifies the recipient's alias as the local part of the email address and uses the default accepted domain. The local part of an email address is the name that appears before the "at" sign (@).This email address policy applies to all users in the organization. You can't remove this email address policy.

For additional management tasks related to e-mail address policies, see Email address policy procedures.

What do you need to know before begin?

- Estimated time to complete: 5 minutes.
- If you remove an email address policy that's used by recipients as the primary policy and no other policies have been configured for recipients, the default policy will be used.
- You can’t delete the default policy. If you want to delete the default policy, you must first assign a different policy as the default.
- If the email address policy you’re deleting contains more than 3,000 recipients, you should use the Shell to perform this procedure.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Email address policies" entry in the Email addresses and address books topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

⚠**Warning:**
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to remove an email address policy

1. Navigate to **Mail flow** > **Email address policies**.
2. In the list view, select the email address policy that you want to delete and then click **Delete** 🗑.
3. In the warning, click **Yes** to remove the policy.

Use the Shell to remove an email address policy

This example removes the e-mail address policy South East Offices.

```
Remove-EmailAddressPolicy -Identity "South East Offices"
```

Type **Y** to confirm that you want to remove the policy, and then press ENTER.

For detailed syntax and parameter information, see Remove-EmailAddressPolicy.

Offline address books

Exchange Server 2013 > Email addresses and address books >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-28*

An offline address book (OAB) is a copy of an address list collection that's been downloaded so a Microsoft Outlook user can access the address book while disconnected from the server. Microsoft Exchange generates the new OAB files and then compresses the files and places them on a local share. You can decide which address lists are made available to users who work offline, and you can also configure the method by which the address books are distributed.

To learn more about address lists, see Address lists.

Important:
OAB data is produced by the Microsoft Exchange OABGen service, which is a mailbox assistant. If you use the security descriptor to prevent users from viewing certain recipients in Active Directory, users who download the OAB will be able to view those hidden recipients. Therefore, to hide a recipient from an address list, set the *HiddenFromAddressListsEnabled* parameter on the Set-PublicFolder, Set-MailContact, Set-MailUser, Set-DynamicDistributionGroup, Set-Mailbox, and Set-DistributionGroup cmdlets. Alternatively, you can create a new default OAB that doesn't contain the hidden recipients. For details about how to add or remove address lists from an OAB, see Add an address list to or remove an address list from an offline address book.

Looking for management tasks related to OABs? See Offline address book procedures.

Contents

Moving OABs between Exchange versions

In Exchange 2007 and Exchange 2010, you use the **Move-OfflineAddressBook** cmdlet to move the OAB generation to another Mailbox server. Exchange 2013 supports only OAB (version 4). This is the same version that was the default in Exchange 2010. You can't configure Exchange 2013 to generate

other OAB versions, and the OAB generation occurs on the Mailbox server on which the organization mailbox resides. Therefore, to move OAB generation in Exchange 2013, you must move the organization mailbox. You can only move the OAB generation to another Exchange 2013 mailbox database. You can't move OAB generation to a previous version of Exchange. To find the Exchange 2013 OAB organization mailbox, run the following Shell command:

```
Get-Mailbox -Arbitration | where {$_.PersistedCapabilities
-like "*oab*"}
```

You can then use the **MoveRequest** cmdlets to move the mailbox.

OAB version 4 and Outlook clients

Exchange 2013 only supports OAB version 4. OAB version 4 was introduced in Exchange 2003 Service Pack 2 (SP2) and is supported by Outlook 2007, Outlook 2010, and Outlook 2013. This Unicode OAB allows client computers to receive differential updates rather than full OAB downloads and a reduced file size.

Outlook clients that use OAB version 4

For Outlook 2013, Outlook 2010, Outlook 2007, and clients that use OAB version 4, if the size of the changes.oab files is half the size (or more) of the entire OAB files, Outlook initiates a full OAB download.

Web-based distribution

Web-based distribution is the distribution method by which Outlook 2013, Outlook 2010, or Outlook 2007 clients that are working offline can access the OAB.

There are several advantages to using Web-based distribution, including:

- Support of more concurrent client computers.
- Reduction in bandwidth usage.
- More control over the OAB distribution points. With Web-based distribution, the distribution point is the HTTPS web address where client computers can download the OAB.

To benefit most from Web-based distribution, client computers must be running Outlook 2013, Outlook 2010, or Outlook 2007.

To function properly, Web-based distribution depends on the following components:

- **OAB generation process** This is the process by which Exchange creates and updates the OAB. To create and update the OAB, the OABGen service runs on the Mailbox server on which the organization mailbox is located. To support OAB distribution, this server must be an Exchange Mailbox server.
- **OAB distribution** If a client initiates the OAB distribution request, the request be directed

through a Client Access server. The Client Access server then routes the request to the Mailbox server that's hosting the OAB files. The OAB files are then distributed directly from the Mailbox server to the client.

- **OAB virtual directory** The OAB virtual directory is the distribution point used by the Web-based distribution method. By default, when Exchange is installed, a new virtual directory named **OAB** is created in the default internal website in Internet Information Services (IIS). If you have client-side users that connect to Outlook from outside your organization's firewall, you can add an external website. Alternatively, when you run the **New-OABVirtualDirectory** cmdlet in the Shell, a new virtual directory named OAB is created in the default IIS website on the local Exchange Client Access server. For information, see Create an offline address book virtual directory.
- **Autodiscover service** This is a feature available in Outlook 2013, Outlook 2010, Outlook 2007, and in some mobile devices that automatically configure the clients for access to Exchange. The service runs on a Client Access server and returns the correct OAB URL for a specific client connection.

OAB considerations

As a best practice, whether you use a single OAB or multiple OABs, consider the following factors as you plan and implement your OAB strategy:

- Size of each OAB in your organization. For more information, see OAB size considerations later in this topic.
- Number of OAB downloads.
- Number and frequency of parent distinguished name changes.
- SMTP address mismatches.
- Overall number of changes made to the directory.

OAB size considerations

For some organizations, the OAB is a small file that remote users occasionally download. For these organizations, downloading the OAB is usually not a concern. However, for some large organizations that have large directories, or for organizations that have deployed Outlook 2003 in Cached Exchange Mode, it may be a concern, especially if the organizations have consolidated Exchange servers into a regional datacenter.

OAB sizes can vary from a few megabytes to a few hundred megabytes. The following factors can affect the size of the OAB:

- Usage of certificates in a company. The more public key infrastructure (PKI) certificates, the larger the OAB. PKI certificates range from 1 kilobyte (KB) to 3 KB. They're the single largest contributor to the OAB size.
- Number of mail recipients in Active Directory.
- Number of distribution groups in Active Directory.
- Information that a company adds to Active Directory for each mailbox-enabled or mail-enabled object. For example, some organizations populate the address properties on each user; others

don't.

Offline address book procedures

Exchange Server 2013 > Email addresses and address books > Offline address books >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-04*

Create an offline address book

Add an address list to or remove an address list from an offline address book

Change the default offline address book

Provision recipients for offline address book downloads

Remove an offline address book

Update an offline address book

Create an offline address book virtual directory

Change the offline address book generation schedule

Configure offline address book distribution properties

Create an offline address book

Email addresses and address books > Offline address books > Offline address book procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-19*

An offline address book (OAB) in Exchange Server 2013 is a downloaded copy of an address book that allows an Outlook user to access the information while disconnected from the server. Exchange administrators can decide which address books are made available to users who work offline, and they can also configure the method by which the address books are distributed (web-based distribution or public folder distribution).

For additional management tasks related to OABs, see Offline address book procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.
- By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to create an OAB with web-based distribution

This example creates an OAB named OAB_Contoso that uses web-based distribution for Outlook 2007 or later clients by using the default virtual directory.

```
New-OfflineAddressBook -Name "OAB_Contoso" -AddressLists
"\Default Global Address List" -VirtualDirectories
"SERVER01\OAB (Default Web Site)"
```

For detailed syntax and parameter information, see New-OfflineAddressBook.

Add an address list to or remove an address list from an offline address book

Email addresses and address books > Offline address books > Offline address book procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-19*

You can use the Shell to add or remove an address list from an offline address book (OAB). By

default, there is an OAB named the Default Offline Address Book that contains the global address list (GAL). OABs are generated based on the address lists that they contain. To create custom OABs that users can download, you can add or remove address lists from OABs.

For additional management tasks related to OABs, see Offline address book procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Offline address books" entry in the Recipients Permissions topic.
- By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.
- Changes to the address list aren't available for client download until after the OAB in which the address list resides has been generated. For more information, see Update an offline address book.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to add an address list to an OAB

When using the *AddressLists* parameter, any address lists that currently exist will be overwritten. You must include existing address lists when you use the *AddressLists* parameter to continue to generate those address lists in your OAB. This example, in which you have AddressList1 and AddressList2, adds AddressList3.

```
Set-OfflineAddressBook -Identity "My OAB" -AddressLists
AddressList1,AddressList2,AddressList3
```

For detailed syntax and parameter information, see Set-OfflineAddressBook.

Use the Shell to remove an address list from an OAB

To remove an address list from an OAB, simply omit that address list from the list of address lists. This example, in which you have AddressList1, AddressList2, and AddressList3, removes AddressList3.

```
Set-OfflineAddressBook -Identity "My OAB" -AddressLists
AddressList1,AddressList2
```

For detailed syntax and parameter information, see Set-OfflineAddressBook.

Change the default offline address book

Email addresses and address books > Offline address books > Offline address book procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-19*

By default, when you install the Mailbox server role, a Web-based default offline address book (OAB) named Default Offline Address Book is created. You can set any OAB in your Exchange organization as the default OAB. This new default OAB is associated with all newly created mailbox databases. You can have only one default OAB in your organization. If you delete the default OAB, Microsoft Exchange doesn't automatically assign another OAB as the default. You must manually designate another OAB as the default.

For additional management tasks related to OABs, see Offline address book procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Offline address books" entry in the Recipients Permissions topic.
- By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to change the default OAB

This example sets the OAB named My OAB as the default OAB.

```
Set-OfflineAddressBook -Identity "My OAB" -IsDefault $true
```

For detailed syntax and parameter information, see Set-OfflineAddressBook.

Provision recipients for offline address book downloads

Email addresses and address books > Offline address books > Offline address book procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-15*

If you use multiple offline address books (OABs) in your organization, there are several ways to specify which recipients download which OABs:

- **Per mailbox database** You can use the EAC or the Shell to provision recipients for OAB downloads by linking a mailbox database to a default OAB for Office Outlook 2007, Outlook 2010 and Outlook 2013 clients.
- **Per recipient** You can use the **Set-Mailbox** cmdlet in the Shell to specify which OAB is downloaded by linking the OAB directly to a recipient's mailbox.
- **Per multiple recipients** You can use a pipelined command in the Shell to specify the OAB that multiple recipients download, based on common attributes.
- **Per address book policy** You can assign an address book policy (ABP) to a mailbox user's account to specify which OAB is downloaded to a recipient's mailbox. If you assign an ABP to a user account that already has an OAB assigned, the OAB that's explicitly assigned to the mailbox will take precedence. For more information, see Assign an address book policy to mail users.

For additional management tasks related to OABs, see Offline address book procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You can't use the Exchange admin center (EAC) to perform these procedures. You must use the

Shell.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to provision recipients for OAB downloads by linking their mailbox database to a public folder database or to a default OAB

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox databases" entry in the Recipients Permissions topic.

This example sets up the web-based distribution of My OAB for the default mailbox database.

```
Set-MailboxDatabase -Identity "Mailbox Database" -
OfflineAddressBook "My OAB"
```

For detailed syntax and parameter information, see Set-MailboxDatabase.

Use the Shell to specify which OAB will be downloaded by linking the OAB directly to a recipient's mailbox

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

To specify which OAB is downloaded by linking the OAB directly to a recipient's mailbox, use the following syntax.

```
Set-Mailbox -Identity <MailboxIDParameter> -
OfflineAddressBook <OfflineAddressBookIdParameter>
```

Note:
The *Identity* parameter identifies the mailbox and can take the following values: GUID, ADObjectID, distinguished name (DN), *domain**account*, user principal name (UPN), LegacyExchangeDN, SmtpAddress, and alias.

This example specifies that the user Kim will download the OAB My OAB.

```
Set-Mailbox -Identity Kim -OfflineAddressBook "My OAB"
```

For detailed syntax and parameter information, see Set-Mailbox.

Use the Shell to specify the OAB that multiple recipients will download

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

This example specifies that all user mailboxes in the United States for Contoso will download the OAB Contoso United States.

```
Get-User -ResultSize Unlimited -Filter { Company -eq
"Contoso" -and RecipientType -eq "UserMailbox" } | Where
{ $_.CountryOrRegion -eq "United States"} | Set-Mailbox -
OfflineAddressBook "Contoso United States"
```

For detailed syntax and parameter information, see Get-User and Set-Mailbox.

Remove an offline address book

Email addresses and address books > Offline address books > Offline address book procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-19*

This topic explains how to remove an offline address book (OAB).

For additional management tasks related to OABs, see Offline address book procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.
- By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any

cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

- After you remove an OAB that's linked to a user or to a mailbox database, the recipient will download the default OAB until you assign a new OAB for that user. If you remove the default OAB, you must assign a different OAB as the default OAB. For instructions about how to change the default OAB, see Change the default offline address book.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to remove an OAB

This example removes an OAB named My OAB.

```
Remove-OfflineAddressBook -Identity "My OAB"
```

Type **Y** to confirm that you want to remove the OAB, and then press ENTER.

For detailed syntax and parameter information, see Remove-OfflineAddressBook.

Update an offline address book

Email addresses and address books > Offline address books > Offline address book procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-11-15*

After you create an OAB or modify OAB settings, the changes aren't available to users until the OAB generation (OABGen) process has completed.

For additional management tasks related to OABs, see Offline address book procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.

- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to update an OAB

This example updates the OAB named My OAB.

```
Update-OfflineAddressBook -Identity "My OAB"
```

For detailed syntax and parameter information, see Update-OfflineAddressBook.

Create an offline address book virtual directory

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

The OAB virtual directory is the distribution for the OAB. By default, when Microsoft Exchange Server 2013 is installed, a new virtual directory named OAB is created in the default internal website in Internet Information Services (IIS). If you have client-side users that connect to Microsoft Outlook from outside your organization's firewall, you can add an external website. Alternatively, when you run the **New-OABVirtualDirectory** cmdlet in the Shell, a new virtual directory named OAB is created in the default IIS website on the local Exchange server.

Creating an OAB virtual directory isn't a common task. Exchange allows for one OAB virtual directory named OAB, and you should create an OAB virtual directory only if there is a problem with the existing OAB virtual directory, and the previous OAB virtual directory was removed.

For additional management tasks related to OABs, see Offline address book procedures.

Important:

Before you create an OAB virtual directory, make sure that your users are aware of the changes you are making. This procedure may interrupt the OAB downloading process for your users.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.
- The local Exchange server must have the Client Access server role installed.
- A default IIS website must exist (for example, /w3svc/1/root).
- A virtual directory named OAB doesn't already exist.
- Although Web-based distribution is enabled by default and doesn't require further configuration, we recommend that you enable Secure Sockets Layer (SSL) for the OAB distribution point.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to create an OAB virtual directory

To create an OAB virtual directory with all of the default settings, you can run the **New-OABVirtualDirectory** cmdlet without any parameters. Use the following procedure to create an OAB virtual directory with custom settings.

Note:

When creating an OAB virtual directory, we recommend that you have SSL enabled.

This example creates an OAB virtual directory on the Client Access server named CASServer01 that has SSL enabled and an external URL.

```
New-OABVirtualDirectory -Server CASServer01 -RequireSSL
$true -ExternalURL "https://www.contoso.com/OAB"
```

After you create a new OAB virtual directory, you must edit the settings on each OAB that uses Web-based distribution to reconnect to the OAB virtual directory. For more information, see Change the offline address book generation schedule.

For detailed syntax and parameter information, see New-OabVirtualDirectory.

Change the offline address book

generation schedule

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-05*

An offline address book (OAB) is a copy of an address book that's been downloaded so that an Outlook user can access the information it contains while disconnected from the server. You can configure how often the OAB is generated by using the *OABGeneratorWorkCycle* and *OABGeneratorWorkCycleCheckpoint* parameters on the Set-MailboxServer cmdlet.

For additional management tasks related to OABs, see Offline address book procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.
- You can't use the Exchange Administration center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to configure OAB properties

In this example, the offline address book is generated every six hours each day on the Mailbox server, MBXServer01.

```
Set-MailboxServer -Identity MBXServer01 -
OABGeneratorWorkCycle 01.00:00:00 -
OABGeneratorWorkCycleCheckpoint 06:00:00
```

For detailed syntax and parameter information, see Set-OfflineAddressBook.

Configure offline address book distribution properties

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-14*

For each offline address book (OAB) distribution point in Exchange, you can configure two URLs—an internal URL that can be accessed only from your internal corporate network and an external URL that can be accessed from the Internet.

For additional management tasks related to OABs, see Offline address book procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to configure OAB distribution properties

This example sets the polling interval for OAB distribution on the OAB virtual directory OAB (Default Web Site) to six hours.

```
Set-OABVirtualDirectory "OAB (Default Web Site)" -
PollInterval 360
```

This example sets the external distribution point to https://contoso.com/OAB for the default OAB virtual directory OAB (Default Web Site).

```
Set-OABVirtualDirectory "OAB (Default Web Site)" -
```

```
ExternalUrl https://contoso.com/OAB
```

For detailed syntax and parameter information, see Set-OabVirtualDirectory.

For More Information

Offline address books

Address lists

Exchange Server 2013 > Email addresses and address books >

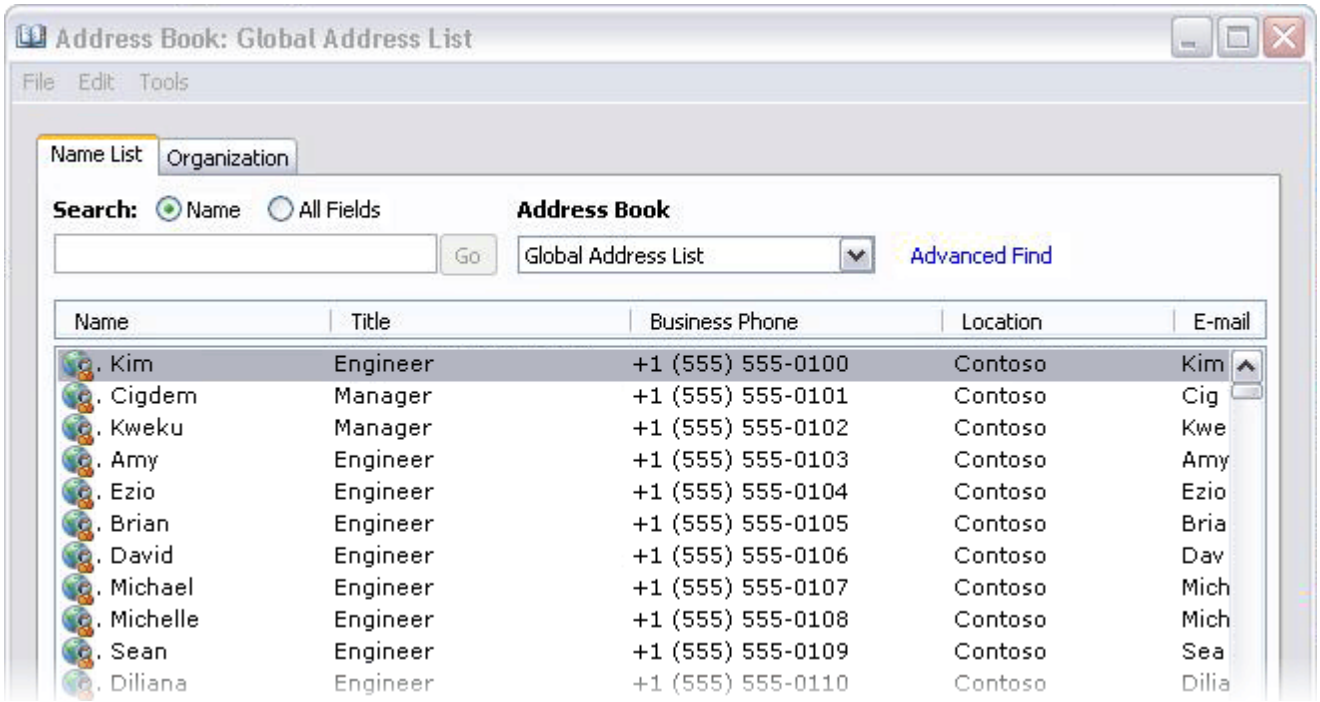
Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-23*

An *address list* is a collection of recipient and other Active Directory objects. Each address list can contain one or more types of objects (for example, users, contacts, groups, public folders, and room and equipment resources). You can use address lists to organize recipients and resources, making it easier to find the recipients and resources you want. Address lists are updated dynamically. Therefore, when new recipients are added to your organization, they're automatically added to the appropriate address lists.

As shown in the following figure, client applications, such as Microsoft Outlook, display the available address lists that Exchange provides.

Global address list as displayed in Microsoft Office Outlook 2007



Address lists reside in Active Directory. Therefore, mobile users who are disconnected from the

network are also disconnected from these server-side address lists. However, you can create offline address books (OABs) for users who are disconnected from the network. These OABs can be downloaded to a user's hard disk. Frequently, to conserve resources, OABs are subsets of the information in the actual address lists that reside on your servers. For more information, see Offline address books.

Contents

Default address lists

When users want to use their client application to find recipient information, they can select from available address lists. Several address lists, such as the global address list (GAL), are created by default. Exchange contains the following default address lists, which are then automatically populated with new users, contacts, groups, or rooms as they're added to your organization:

- **All Contacts** This address list contains all mail-enabled contacts in your organization. Mail-enabled contacts are those recipients who have an external email address. If you want mail-enabled contact information to be available to all users in your organization, you must include the contact in the GAL. To learn more about mail contacts, see Recipients.
- **All Distribution Lists** This address list contains mail-enabled groups, such as mail-enabled security groups, distribution groups and dynamic distribution groups in your organization. Mail-enabled groups are lists of recipients that are created to expedite the mass sending of email messages and other information. When an email message is sent to a mail-enabled group, all members of that list receive a copy of the message. To learn more about mail-enabled groups, see Recipients.
- **All Rooms** This address list contains all resources that have been designated as a room in your organization. Rooms are resources in your organization that can be scheduled by sending a meeting request from a client application. The user account that's associated with a room is disabled. To learn more about resource mailboxes, see Recipients.
- **All Users** This address list contains all mail-enabled users in your organization. A mail-enabled user represents a user outside your Exchange organization. Each mail-enabled user has an external email address. All messages sent to mail-enabled users are routed to this external email address. A mail-enabled user is similar to a mail contact, except that a mail-enabled user has Active Directory logon credentials and can access resources. To learn more about mail-enabled users, see Recipients.
- **Default Global Address List** This address list contains all mail-enabled users, contacts, groups, or rooms in the organization. During setup, Exchange creates various default address lists. The most familiar address list is the GAL. By default, the GAL contains all recipients in an Exchange organization. In other words, any mailbox-enabled or mail-enabled object in an Active Directory

forest that has Exchange installed is listed in the GAL. For ease of use, the GAL is organized by name, not by email address. For more information, see Create a global address list.

- **Public Folders** This address list contains all public folders in your organization. Access permissions determine who can view and use the folders. Public folders are stored on computers running Exchange. For more information about public folders in Exchange 2013, see Public folders. For more information about public folders in Exchange Online, see **Public folders in Office 365 and Exchange Online**.

Custom Address Lists

An Exchange organization can contain thousands of recipients. If you compile all your recipients in the default address lists, those lists could become quite large. To prevent this, you can create custom address lists to help users in your organization find what they are looking for more easily.

For example, consider a company that has two large divisions and one Exchange organization. One division, named Fourth Coffee, imports and sells coffee beans. The other division, Contoso, Ltd, underwrites insurance policies. For most day-to-day activities, the employees at Fourth Coffee don't communicate with the employees at Contoso, Ltd. Therefore, to make it easier for employees to find recipients who exist only in their division, you can create two new custom address lists—one for Fourth Coffee and one for Contoso, Ltd. When searching for recipients in their division, these custom address lists allow employees to select only the address list that's specific to their division. However, if an employee is unsure about the division in which the recipient exists, the employee can search within the GAL, which contains all recipients in both divisions.

You can also create subcategories of address lists called hierarchical address lists. For example, you can create an address list that contains all recipients in Manchester and another that contains all recipients in Stuttgart.

Best Practices for Creating Address Lists

Although address lists are useful tools for users, poorly planned address lists can cause frustration. To make sure that your address lists are practical for users, consider the following best practices:

- Avoid creating so many address lists that users won't be sure which list to search for recipients.
- Address lists should make it easier for users to find addresses in the GAL.
- Name your address lists in such a way that, when users glance at them, they will know immediately which recipient types are contained in the list. If you have difficulty naming your address lists, create fewer lists and remind users that they can find anyone in your organization by using the GAL.

Note:

By default in Exchange Online, the Address List role isn’t assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the “Add a role to a role assignment policy” section of Manage role assignment policies.

For detailed instructions about creating an address list in Exchange 2013, see Create an address list. For detailed instructions about creating an address list in Exchange Online, see **Manage address lists**.

Address list procedures

Exchange Server 2013 > Email addresses and address books > Address lists >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-12*

Create an address list

Update an address list

Create an address list by using recipient filters

Move an address list

Remove an address list

Create a global address list

Configure global address list properties

Remove a global address list

Update a global address list

Create an address list

Email addresses and address books > Address lists > Address list procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-12*

Address lists are a collection of recipient and other Active Directory objects. Each address list can contain one or more types of objects (for example, users, contacts, groups, public folders, conferencing, and other resources). Address lists also provide a mechanism to partition mail-enabled objects in Active Directory for the benefit of specific groups of users.

For other management tasks related to address lists, see Address list procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to create an address list

1. Navigate to **Organization** > **Address lists**, and then click **Add +**.
2. In **Address List**, type a name and specify the types of recipients to include in the list.
3. By default, Exchange creates address lists that contain all members of your organization. To create a unique custom address list, click **Add a rule**.

Important:

If you don't add a rule, you'll create an address list that's redundant with one of the default address lists.

4. In the list, select a filtering option (for example, **Custom attribute 1**).
5. In **Specify words or phrases**, type words or phrases to filter by, click **Add +**, and then click **OK**.

You can continue to add several phrases or words by repeating Step 4. The filter is a Boolean **OR** statement. For example, you can create a filter that will apply the address list to users whose Custom 1 attribute equals **Oregon**, **Idaho**, or **Washington**.

6. (Optional) Click **Add a rule** again to add additional filters. Additional filters create a Boolean **And** statement. The more filters you add, the fewer number of users the address list will apply to.
7. Click **Preview recipients the address lists includes** to see the recipients that this address list is going to apply to.
8. Click **Save**.
9. You'll get a warning that the address list won't be applied until you update it. Depending on the size of your organization and the filters that you added to the address list, some address lists can contain thousands or tens of thousands of recipients. Updating address lists can impact your resources, so you may want to update the address during off-peak hours.

For details about updating an address list, see Update an address list.

Use the Shell to create an address list

This example creates the address list MyAddressList by using the *RecipientFilter* parameter and includes recipients that are mailbox users and have `StateOrProvince` set to `Washington` or `Oregon`.

```
New-AddressList -Name MyAddressList -RecipientFilter
{((RecipientType -eq 'UserMailbox') -and ((StateOrProvince
-eq 'Washington') -or (StateOrProvince -eq 'Oregon')))}
```

This example creates the child address list Building 34 Meeting Rooms in the All Rooms parent container, using built-in conditions.

```
New-AddressList -Name "Building 34 Meeting Rooms" -
Container "\All Rooms" -IncludedRecipients Resources -
ConditionalCustomAttribute1 "Building 34"
```

For detailed syntax and parameter information, see New-AddressList.

Update an address list

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-14*

Address lists are a collection of recipient and other Active Directory objects. You apply an address list when the address list filter rule has been edited. To update the membership of the address list to include new recipients and remove those who no longer meet the filtering criteria, you must apply the address list.

For additional management tasks related to address lists, see Address list procedures.

What do you need to know before you begin?

- Estimated time to complete: This process may take a long time to finish depending on the number of recipients in the address list.
- Some address lists contain thousands or tens of thousands of recipients depending on the size of your organization and the filters that you added to the address list. Updating address lists can take up a lot of computer resources. So, you may want to update the address list during off-peak hours.
- If the address list contains more than 3,000 recipients, we recommend that you use the Exchange Management Shell to update the address list.

For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server,

Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to update an address list

1. Navigate to **Organization** > **Address lists**.
2. In the list view, select the address list that you want to update.
3. In the details pane, click **Update**.

Use the Shell to update an address list

This example updates the address list Washington State.

```
Update-AddressList "Washington State"
```

If you have more than one address list with the same name, you must specify the full path to the address list you want to update. For example, if you want to update the address list Sales under North America but there is also a Sales address list under Europe, use the following command:

```
Update-AddressList "North America\Sales"
```

For detailed syntax and parameter information, see Update-AddressList.

Create an address list by using recipient filters

Email addresses and address books > Address lists > Address list procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-18*

This topic explains how to create an address list by using recipient filters. To learn more about address lists, see Address lists.

For additional management tasks related to address lists, see Address list procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

- By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.
- To use the *RecipientFilter* parameter to create a custom filter, you must specify a string for the filter. The Shell uses OPATH for the filtering syntax. OPATH is a querying language designed to query object data sources.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to create an address list by using recipient filters

This example creates an address list for all users with Exchange mailboxes who reside in Washington or Oregon.

```
New-AddressList -Name "Pacific Northwest Mailboxes" -
RecipientFilter {((RecipientType -eq 'UserMailbox') -and
((StateOrProvince -eq 'Washington') -or (StateOrProvince -
eq 'Oregon')))}
```

This example creates an address list for all users with Exchange mailboxes who have AgencyB as the value for the *CustomAttribute15* parameter.

```
New-AddressList -Name "AgencyB" -RecipientFilter
{(RecipientType -eq 'UserMailbox') -and (CustomAttribute15
-like *AgencyB*)}
```

For detailed syntax and parameter information, see New-AddressList.

Move an address list

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-14*

This topic explains how to move an existing address list to a new container under the root address list.

For additional management tasks related to address lists, see Address list procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address Lists" entry in the Email address and address book permissions topic.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to move an address list

This example uses the address list's GUID to move the address list to the Building 4 container, which is located in the All Users\Sales container.

```
Move-AddressList -Identity c3fffd8e-026b-41b9-88c4-
8c21697ac8ac -Target "\All Users\Sales\Building4"
```

Type **Y** to confirm that you want to move this address list, and then press ENTER.

For detailed syntax and parameter information, see Move-AddressList.

Remove an address list

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-14*

This topic explains how to remove an address list. You can't remove the default global address list

(GAL).

For additional management tasks related to address lists, see Address list procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address list" entry in the Email address and address book permissions topic.
- You can't remove a parent address list that contains child address lists. However, you can remove both the child and parent address lists by pressing the CTRL key on the keyboard, and then selecting the parent and child address lists. If you attempt to remove a parent address list without removing the child address lists, you'll receive an error.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to remove an address list

1. Navigate to **Organization** > **Address lists**.
2. In the list view, select the address list you want to remove, and then click **Delete** .
3. In the warning, click **Yes** to remove the address list.

Use the Shell to remove an address list

This example removes the address list Sales Department, which doesn't contain child address lists.

```
Remove-AddressList -Identity "Sales Department"
```

Type **Y** to confirm that you want to remove this address list, and then press ENTER.

For detailed syntax and parameter information, see Remove-AddressList.

Use the Shell to remove an address list that contains child address lists

This example removes the parent address list Departments and all its child address lists.

```
Remove-AddressList -Identity Departments -Recursive
```

Type **Y** to confirm that you want to remove the parent address list and its child address lists, and then press ENTER.

For detailed syntax and parameter information, see Remove-AddressList.

Create a global address list

Email addresses and address books > Address lists > Address list procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-18*

The global address list (GAL) is a directory that contains entries for every group, user, and contact within an organization's implementation of Microsoft Exchange. If your organization uses address book policies, you may want to create additional GALs. To learn more, see Address book policies.

For additional management tasks related to address lists, see Address list procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.
- By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to create a GAL using conditional filter

properties

This example creates a GAL named GAL_Contoso that includes recipients who are mailbox users and have their company listed as Contoso.

```
New-GlobalAddressList -Name "GAL_Contoso" -
IncludedRecipients MailboxUsers -ConditionalCompany Contoso
```

Note:

If you're using precanned conditional filter properties, the *IncludedRecipients* parameter can't be blank.

For detailed syntax and parameter information, see New-GlobalAddressList.

Use the Shell create a GAL using a recipient filter

This example creates a GAL named GAL_AgencyA that includes recipients for which the *CustomAttribute15* parameter has a value of AgencyA.

```
New-GlobalAddressList -Name "GAL_AgencyA" -RecipientFilter
{CustomAttribute15 -like "AgencyA"}
```

For detailed syntax and parameter information, see New-GlobalAddressList.

Configure global address list properties

Email addresses and address books > Address lists > Address list procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-18*

This topic explains how to modify the settings of a global address list (GAL).

For additional management tasks related to address lists, see Address list procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.
- By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any

cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

- You can't edit the settings of the default GAL.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to configure GAL properties

This example assigns a new name, FourthCoffee, to the GAL that has the GUID 96d0c505-eba8-4103-ad4f-577a1bf4ad7b.

```
Set-GlobalAddressList -Identity 96d0c505-eba8-4103-ad4f-577a1bf4ad7b -Name FourthCoffee
```

Note:

If you're using precanned conditional filter properties, the value for the *IncludedRecipients* parameter can't be blank.

This example changes the recipients who will be included in the Fourth Coffee global GAL to those whose company is set to Fourth Coffee.

```
Set-GlobalAddressList -Identity Fourth Coffee -RecipientFilter {Company -eq "Fourth Coffee"}
```

For detailed syntax and parameter information, see Set-GlobalAddressList.

Remove a global address list

Email addresses and address books > Address lists > Address list procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-18*

The global address list (GAL) is a directory that contains entries for every group, user, and contact within an Exchange organization.

For additional management tasks related to address lists, see Address list procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.
- By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.
- You can't remove the default GAL.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to remove a GAL

This example removes the GAL Fourth Coffee from the domain controller ad-server.fourthcoffee.com.

```
Remove-GlobalAddressList -Identity "Fourth Coffee" -
DomainController ad-server.fourthcoffee.com
```

To confirm that you want to remove the GAL, type **Y**, and then press ENTER.

For detailed syntax and parameter information, see Remove-GlobalAddressList.

Update a global address list

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-18*

You can use the Shell to update a global address list (GAL). A GAL is a directory that contains entries for every group, user, and contact within an organization's implementation of Microsoft Exchange.

For additional management tasks related to address lists, see Address list procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.
- By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to update a GAL

This example updates a GAL for the Fourth Coffee company.

Note:

Running this command only starts the update process. It may take several hours for the GAL to be updated.

```
Update-GlobalAddressList -Identity "Fourth Coffee"
```

For detailed syntax and parameter information, see Update-GlobalAddressList.

Hierarchical address books

Exchange Server 2013 > Email addresses and address books >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-26*

The hierarchical address book (HAB) allows end users to look for recipients in their address book using an organizational hierarchy. Normally, users are limited to the default global address list (GAL) and its recipient properties and the structure of the GAL often doesn't reflect the

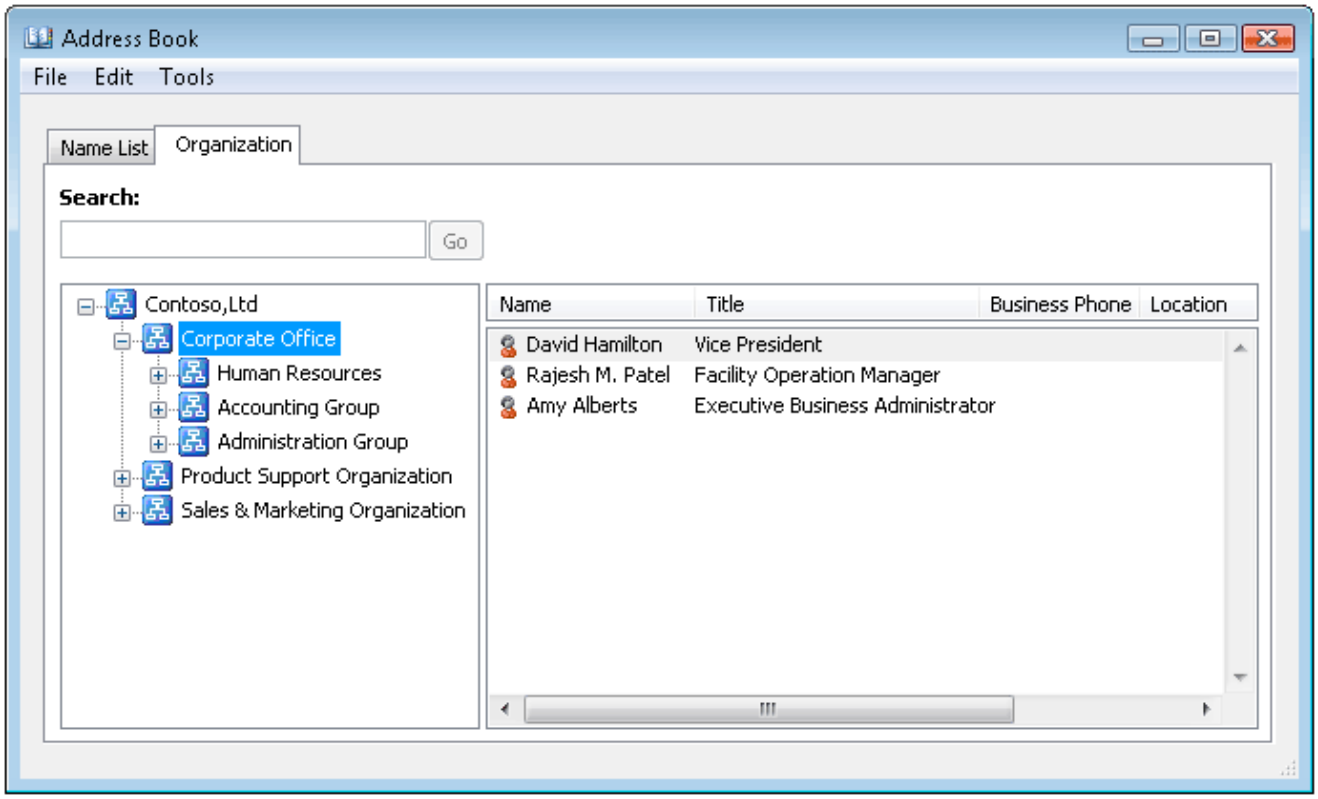
management or seniority relationships of recipients in your organization. Being able to customize an HAB that maps to your organization's unique business structure provides your users with an efficient method for locating internal recipients.

Using hierarchical address books

In an HAB, your root organization (for example, Contoso, Ltd) is used as the top-level tier. Under this top-level tier, you can add several child tiers to create a customized HAB that's segmented by division, department, or any other organizational tier you want to specify. The following figure illustrates an HAB for Contoso, Ltd with the following structure:

- The top-level tier represents the root organization Contoso, Ltd.
- The second-level child tiers represent the business divisions within Contoso, Ltd: Corporate Office, Product Support Organization, and Sales & Marketing Organization.
- The third-level child tiers represent departments within the Corporate Office division: Human Resources, Accounting Group, and Administration Group.

Example HAB for Contoso, Ltd



You can provide an additional level of hierarchical structure by using the *SeniorityIndex* parameter. When creating an HAB, use the *SeniorityIndex* parameter to rank individual recipients or organizational groups by seniority within these organizational tiers. This ranking specifies the order in which the recipients or groups are displayed in the HAB. For example, in the preceding example, the *SeniorityIndex* parameter for the recipients in the Corporate Office division is set to the following:

- 100 for David Hamilton

- 50 for Rajesh M. Patel
- 25 for Amy Alberts

Note:
If the *SeniorityIndex* parameter isn't set or is equal for two or more users, the HAB sorting order uses the *PhoneticDisplayName* parameter value to list the users in ascending alphabetical order. If the *PhoneticDisplayName* parameter value isn't set, the HAB sorting order defaults to the *DisplayName* parameter value and lists the users in ascending alphabetical order.

Configuring hierarchical address books

Detailed instructions for creating HABs are included in the topic Enable or disable hierarchical address books. The general steps are as follows:

1. Create a distribution group that will be used for the root organization (top-level tier). If desired, you can use an existing organizational unit in your Exchange forest for the distribution group.
2. Create distribution groups for the child tiers and designate them as members of the HAB. Modify the *SeniorityIndex* parameter of these groups so they're listed in the proper hierarchical order within the root organization.
3. Add organization members. Modify the *SeniorityIndex* parameter of the members so they're listed in the proper hierarchical order within the child tiers.
4. For accessibility purposes, you can use the *PhoneticDisplayName* parameter, which specifies a phonetic pronunciation of the *DisplayName* parameter.

Enable or disable hierarchical address books

Exchange Server 2013 > Email addresses and address books > Hierarchical address books >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-26*

You can configure a hierarchical address book (HAB), which is a feature available to end users in Microsoft Outlook 2010 or later. With an HAB, users can look for recipients in their Exchange organization by using an organizational hierarchy based on seniority or management structure. To learn more about HABs, see Hierarchical address books.

What do you need to know before you begin?

- Estimated time to complete: 1 hour.

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.
- You can't use the Exchange Administration Center (EAC) to perform this procedure. You must use the Shell.
- Before you get started, read the topic Hierarchical address books. You should understand if a HAB is appropriate for your Exchange organization.
- Understand how organizational units (OUs), groups, users, and contacts are currently configured in your Exchange organization.
- Understand the cmdlets and associated parameters in the following table, which are required to configure a HAB.

Cmdlet	Parameter
Set-OrganizationConfig	*HierarchicalAddressBookRoot*
Set-Group	*IsHierarchicalGroup* *SeniorityIndex* *PhoneticDisplayName*
Set-User	*SeniorityIndex* *PhoneticDisplayName*
Set-Contact	*SeniorityIndex* *PhoneticDisplayName*

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable a HAB

Note:
Although you can't use the EAC to enable a HAB, after it's enabled you can use the EAC to manage the membership of the groups in the organizational hierarchy.

For this example, an OU called HAB will be created for the HAB. The name of the domain for the organization is Contoso-dom, and Contoso,Ltd will be the name of the top-level organization in the hierarchy (the *root organization*). Subordinate groups named Corporate Office, Product Support Organization, and Sales & Marketing Organization will be created as child organizations under Contoso,Ltd. Additionally, the groups Human Resources, Accounting Group, and Administration

Group will be created as child organizations under Corporate Office.

For detailed information about creating distribution groups, see Manage Distribution Groups.

1. Create an OU named HAB in the Contoso organization. You can use Active Directory Users and Computers or type the following at a command prompt.

Note:
Alternatively, you can use an existing OU in your Exchange forest.

```
dsadd ou "OU=HAB,DC=Contoso-dom,DC=Contoso,DC=com"
```

Note:
For details, see Create a New Organizational Unit.

2. Create the root distribution group Contoso,Ltd for the HAB.

Note:
For the purposes of this topic, the Shell example is provided. However, you can also use the EAC to create a distribution group. For details, see Manage Distribution Groups.

```
New-DistributionGroup -Name "Contoso,Ltd" -DisplayName
"Contoso,Ltd" -Alias "ContosoRoot" -OrganizationalUnit
"Contoso-dom.Contoso.com/HAB" -SamAccountName "ContosoRoot"
-Type "Distribution"
```

3. Designate Contoso,Ltd as the root organization for the HAB.

```
Set-OrganizationConfig -HierarchicalAddressBookRoot
"Contoso,Ltd"
```

4. Create distribution groups for the other tiers in the HAB. For this example, you would create the following groups: Corporate Office, Product Support Organization, Sales & Marketing Organization, Human Resources, Accounting Group, and Administration Group. This example creates the distribution group Corporate Office.

Note:
For the purposes of this topic, the Shell example is provided. However, you can also use the EAC to create distribution groups. For details, see Manage Distribution Groups.

```
New-DistributionGroup -Name "Corporate Office" -DisplayName
"Corporate Office" -Alias "CorporateOffice" -
OrganizationalUnit "Contoso-dom.Contoso.com/HAB" -
SamAccountName "CorporateOffice" -Type "Distribution"
```

5. Designate each of the groups as members of the HAB. For this example, you would designate the following groups as being hierarchical groups: Contoso,Ltd, Corporate Office, Product Support Organization, Sales & Marketing Organization, Human Resources, Accounting Group, and

Administration Group. This example designates the distribution group Contoso,Ltd as a member of the HAB.

```
Set-Group -Identity "Contoso,Ltd" -IsHierarchicalGroup
$true
```

6. Add each of the subordinate groups as members of the root organization. For this example, distribution groups Corporate Office, Product Support Organization, and Sales & Marketing Organization, are added as members of the root organization Contoso,Ltd in the HAB. This example adds the Corporate Office distribution group as a member of the Contoso,Ltd root distribution group.

Note:
This example uses the alias of the distribution groups.

```
Add-DistributionGroupMember -Identity "ContosoRoot" -Member
"CorporateOffice"
```

7. Add each of the groups that are subordinate to the distribution group Corporate Office as members of the group. For this example, distribution groups Human Resources, Accounting Group, and Administration Group, are added as members of the distribution group Corporate Office. This example adds the Human Resources distribution group as a member of the Corporate Office distribution group.

Note:
This example uses the alias of the distribution groups and assumes the Human Resources distribution group alias is HumanResources.

```
Add-DistributionGroupMember -Identity "CorporateOffice" -
Member "HumanResources"
```

8. Add users to the groups in the HAB. For this example, David Hamilton (SMTP address DHamilton@contoso.com) is an existing user in the OU Contoso-dom.Contoso.com/Users and will be added to the group Corporate Office. Repeat this step to add other users to groups in the HAB.

```
Add-DistributionGroupMember -Identity "CorporateOffice" -
Member "DHamilton"
```

9. Set the *SeniorityIndex* parameter for groups in the HAB. For this example, the Corporate Office group contains three child groups: Human Resources, Accounting Group, and Administration Group. Instead of having the groups listed in ascending alphabetical order, which is the default, the preferred sorting will be Human Resources (*SeniorityIndex* = 100), Accounting Group (*SeniorityIndex* = 50), and then Administration Group (*SeniorityIndex* = 25). This example sets the *SeniorityIndex* parameter for the Human Resources group to 100.

```
Set-Group -Identity "Human Resources" -SeniorityIndex 100
```

Note:

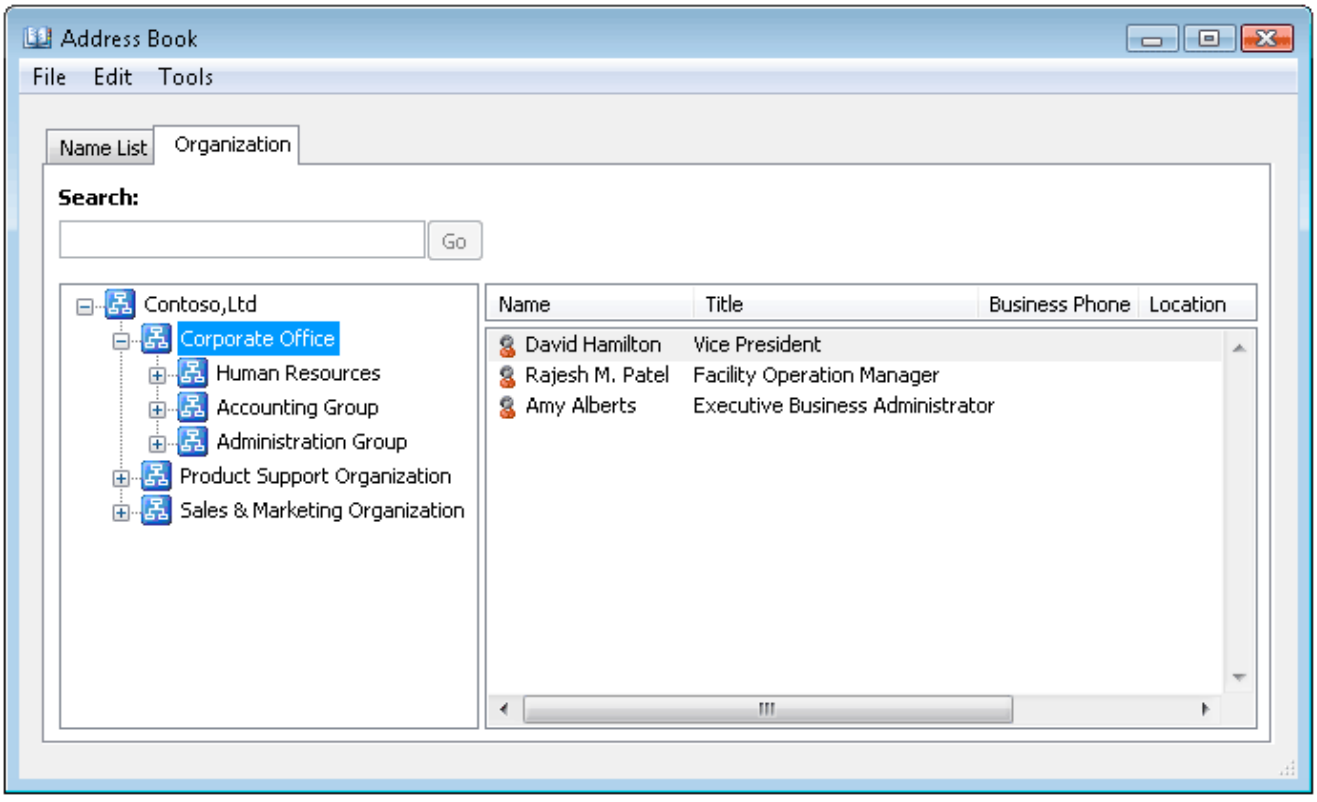
The *SeniorityIndex* parameter is a numerical value used to sort groups or users in descending numerical order in a HAB. If the *SeniorityIndex* parameter isn't set or is equal for two or more users, the HAB sorting order uses the *PhoneticDisplayName* parameter value to list the users in ascending alphabetical order. If the *PhoneticDisplayName* value isn't set, the HAB sorting order defaults to the *DisplayName* parameter value and lists the users in ascending alphabetical order.

10. Set the *SeniorityIndex* parameter for users in the HAB groups. For this example, the Corporate Office group contains three users: Amy Alberts, David Hamilton, and Rajesh M. Patel. Instead of having the users listed in ascending alphabetical order by default, the preferred sorting will be David Hamilton (*SeniorityIndex* = 100), Rajesh M. Patel (*SeniorityIndex* = 50), and then Amy Alberts (*SeniorityIndex* = 25). This example sets the *SeniorityIndex* parameter for the user David Hamilton to 100.

```
Set-User -Identity "DHamilton@contoso.com" -SeniorityIndex
100
```

After completing the preceding steps, the HAB will be visible in Outlook. To view the HAB, open Outlook and click **Address Book**. The HAB is displayed on the **Organization** tab, similar to the following figure.

Example HAB for Contoso,Ltd



After the HAB is created, you can use the EAC to manage the membership of the groups in the organizational hierarchy. However, you must use the Shell to modify the *SeniorityIndex* parameter for any new groups or users.

For detailed syntax and parameter information, see the following:

- New-DistributionGroup
- Set-OrganizationConfig
- Set-Group
- Add-DistributionGroupMember
- Set-User

Use the Shell to disable a hierarchical address book

This example disables the root organization used for the HAB.

```
Set-OrganizationConfig -HierarchicalAddressBookRoot $null
```

Note:

This command doesn't delete the root organization or child groups used in the HAB structure or reset the *SeniorityIndex* values for groups or users. It only prevents the HAB from being displayed in Outlook. To enable the HAB with the same configuration settings again, you only need to enable the root organization again.

For detailed syntax and parameter information, see Set-OrganizationConfig.

Sharing

Exchange Server 2013 >

Applies to: *Outlook 2013*

Topic Last Modified: *2014-02-14*

You may need to coordinate schedules with people in different organizations or with friends and family members so that you can work together on projects or plan social events. With Exchange 2013, administrators can set up different levels of calendar access to allow businesses to collaborate with other businesses and to let users share their schedules with others. Business-to-business calendar sharing is set up by creating *organization relationships*. User-to-user calendar sharing is set up by applying *sharing policies*.

Important:

This feature of Exchange Server 2013 isn't fully compatible with Office 365 operated by 21Vianet in China and some feature limitations may apply. For more information, see Learn about Office 365 operated by 21Vianet.

Contents

Limitations of free/busy sharing

Firewall considerations for federated sharing

Coexistence with Exchange 2010

Coexistence with Exchange 2007

Sharing Documentation

Sharing Scenarios in Exchange 2013

The following sharing scenarios are supported in Exchange 2013:

Sharing goal	Setting to use	Requirements
Share calendars with an Office 365 organization	Organization relationships	The Office 365 organization is ready to configure. The on-premises Exchange administrator has to set up an authentication relationship with the cloud (also known as "federation") and must meet minimum software requirements. To learn more about setting up federation, see Federation.
Share calendars with another on-premises Exchange organization	Organization relationships	Both on-premises Exchange organizations have to set up federation and must meet minimum software requirements
Share an Exchange user's calendar with an Internet user	Sharing policies	None, ready to configure
Share an Exchange user's calendar with another Exchange on-premises user	Sharing policies	Both on-premises Exchange organizations have to set up federation and must meet minimum software

		requirements.

The following table lists the differences between organization relationships and sharing policies.

Organization relationships vs. sharing policies

Functionality	Organization relationship	Sharing policy
Requires a federation trust for your organization	Yes	Yes when sharing with other federated domain organizations. Not required for Internet sharing policies.
Recommends that the external domain be federated	Yes	Yes when sharing with other federated domain organizations. Not required for Internet sharing policies.
Allows sharing of free/busy information (including subject and location) with external organizations for a set of many users.	Yes	No
Allows sharing of Calendar folders with free/busy information	No	Yes
Allows sharing of Calendar folders with free/busy information, including subject and body	No	Yes
Requires users to send a sharing invitation to external recipients	No	Yes
Provides an access method	Your Client Access server accesses the Client Access	Your Client Access server accesses the Client Access

	server of the external organization and retrieves free/busy information for the external user when requested.	server of the external organization and subscribes to the external user's Calendar folder. For Internet sharing policies, external users access either a restricted or public URL on the Client Access server.
Can be applied to all external domains	No (a one-to-one relationship between two Exchange 2013 organizations)	Yes
Provides users with different sharing experiences with external recipients	No	Yes, based on the sharing policy that's applied
Disables sharing for some users	Yes, by specifying a security distribution group for the organization relationship	Yes, by disabling the sharing policy that's applied
Requires that the mailbox reside on an Exchange 2013 Mailbox server	No	Yes

Return to top

Limitations of free/busy sharing

The following limitations apply when sharing free/busy information between Exchange organizations:

1. **Outlook Web Access 2003** When a user in an Exchange 2003 organization uses Outlook Web Access to access free/busy for users in a remote Exchange 2013 organization, the request will fail. Outlook Web Access connections from Exchange 2003 can't make WebDAV (Web-based Distributed Authoring and Versioning) connections to a free/busy system folder to retrieve the free/busy information for remote users. Because Exchange 2013 doesn't support WebDAV connections, the Exchange 2003 server can't connect to External (FYDIBOHF25SPDLT) on the Exchange 2013 CAS server for Outlook Web Access requests. Outlook clients don't experience this

limitation because they use MAPI instead of WebDAV when connecting to External (FYDIBOHF25SPDLT).

2. **Wide Area Network (WAN) latency** In Exchange 2003 organizations, the replicas for all free/busy folders must reside on Exchange 2010 SP2 or higher Mailbox servers. In environments where Exchange 2003 public folder databases are located in multiple physical sites, there may be excessive latency and performance issues if internal free/busy queries have to traverse WAN links to access Exchange 2010 public folder databases not located in the same physical site.
3. **Free/busy information period** Free/busy information requests to an Exchange 2007 organization from an Exchange 2013 organization may fail due to a mismatch in the requested free/busy information period. By default, Exchange 2007 accepts availability requests for 42 days of free/busy information and Exchange 2013 may request 62 days of free/busy information. If the request exceeds the default 42 limit imposed by Exchange 2007, the request will fail.

Follow the steps below to configure your Exchange 2007 CAS servers to accept longer period free/busy information requests:

a. On all your Exchange 2007 CAS servers, open the following file with a text editor such as Notepad: <Exchange Installation Path>\V14\ClientAccess\ExchWeb\EWS\web.config

Caution:

Before you make any changes to the web.config file, make a copy of the file and store it in a safe location.

b. Locate the **appSettings** section in the web.config file.

c. Add a new key "<add key="maximumQueryIntervalDays" value="62" />" and save the web.config file.

Note:

The maximumQueryIntervalDays value isn't present by default. When this value isn't present, Exchange 2007 uses the default interval of 42 days.

d. Stop and restart the Microsoft Internet Information Services (IIS) on all the Exchange 2007 CAS servers.

4. **Exchange organizations that have both on-premises and cloud users** If you set up calendar sharing with another Exchange organization that is configured in a hybrid deployment with Microsoft Office 365, free/busy availability lookups for Office 365-based or remote users that have been moved to the cloud will fail. Because the organization relationship for your Exchange organization is with the remote on-premises Exchange organization, not the Office 365-based Exchange Online organization, the free/busy request can't query the Office 365-based users. Exchange 2013 doesn't support functionality to proxy these availability requests through the on-premises organization to the Office 365 service.

For details about how to configure free/busy sharing between common Exchange deployments, see Configuring federated sharing between Exchange organizations.

Firewall considerations for federated sharing

Federated sharing features require that the Client Access servers in your organization have

outbound access to the Internet by using HTTPS. You must allow outbound HTTPS access (port 443 for TCP) to all Exchange 2013 Client Access servers in the organization.

For an external organization to access your organization's free/busy information, you must publish at least one Client Access server to the Internet. This requires inbound HTTPS access from the Internet to the Client Access server. Client Access servers in Active Directory sites that don't have a Client Access server published to the Internet can use Client Access servers in other Active Directory sites that are accessible from the Internet. The Client Access servers that aren't published to the Internet must have the external URL of the Web services virtual directory set with the URL that's visible to external organizations.

Return to top

Coexistence with Exchange 2010

In organizations that contain both Exchange 2010 and Exchange 2013 servers, users who have a mailbox on an Exchange 2010 Mailbox server can use organization relationships to share free/busy information with recipients in external Exchange 2013 federated domain organizations. The Exchange 2010 Client Access and Mailbox servers must be running SP2 or higher, and you must have at least one Exchange 2013 Client Access server in the Exchange 2010 organization.

Coexistence with Exchange 2007

In organizations that contain both Exchange 2013 and Exchange 2007 servers, users who have a mailbox on an Exchange 2007 Mailbox server can use organization relationships to share free/busy information with recipients in external federated domain organizations. The Mailbox server must be running Exchange 2007 SP2 or higher, and you must have at least one Exchange 2013 Client Access server in the Exchange organization. You can use organization relationships by introducing a single Exchange 2013 Client Access server in the organization, providing a more robust solution than solutions that synchronize free/busy information and require GAL synchronization.

When using Outlook 2010 or Outlook Web App to scheduling a meeting on an Exchange 2007 server, a user who has a mailbox on an Exchange 2007 server can see free/busy information for a user in the external organization. Free/busy information for Exchange 2007 mailboxes is visible to recipients in the external organization.

Sharing policies are assigned to Exchange 2013 mailbox users. To use sharing policies, a mailbox must be located on an Exchange 2013 Mailbox server. Only Outlook 2010 and Outlook Web App clients can be used to generate or respond to sharing invitations.

Return to top

Sharing documentation

The following table contains links to topics that will help you learn about and manage sharing in Exchange 2013.

Topic	Description
Federation	Learn more about the underlying trust infrastructure that supports sharing, an easy method for users to share calendar information with external recipients.
Organization relationships	Learn more about the one-to-one relationships between Exchange organizations that enable calendar free/busy sharing.
Sharing policies	Learn more about the person-to-person policies that enable sharing.

Federation

Exchange Server 2013 > Sharing >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-17*

Information workers frequently need to collaborate with external recipients, vendors, partners, and customers and share their free/busy (also known as calendar availability) information. Federation in Microsoft Exchange Server 2013 helps with these collaboration efforts. *Federation* refers to the underlying trust infrastructure that supports *federated sharing*, an easy method for users to share calendar information with recipients in other external federated organizations. To learn more about federated sharing, see Sharing.

Important:
This feature of Exchange Server 2013 isn't fully compatible with Office 365 operated by 21Vianet in China and some feature limitations may apply. For more information, see Learn about Office 365 operated by 21Vianet.

Contents

Federated organization identifier

Federation example

Certificate requirements for Federation

Transitioning to a new certificate

Firewall Considerations for Federation

Key terminology

The following table defines the core components associated with federation in Exchange 2013.

application identifier (AppID)

A unique number generated by the Windows Azure AD authentication system to identify Exchange organizations. The AppID is automatically generated when you create a federation trust with the Windows Azure AD authentication system.

delegation token

A Security Assertion Markup Language (SAML) token issued by the Windows Azure AD authentication system that allows users from one federated organization to be trusted by another federated organization. A delegation token contains the user's email address, an immutable identifier, and information associated with the offer for which the token is issued for action.

external federated organization

An external Exchange organization that's established a federation trust with the Windows Azure AD authentication system.

federated sharing

A group of Exchange features that leverage a federation trust with the Windows Azure AD authentication system to work across Exchange organizations, including cross-premises Exchange deployments. Together, these features are used to make authenticated requests between servers on behalf of users across multiple Exchange organizations.

federated domain

An accepted authoritative domain that's added to the organization identifier (OrgID) for an Exchange organization.

domain proof encryption string

A cryptographically secure string used by an Exchange organization to provide proof that the organization owns the domain used with the Windows Azure AD authentication system. The string is generated automatically when using the **Enable federation trust** wizard or can be generated by using the **Get-FederatedDomainProof** cmdlet.

federated sharing policy

An organization-level policy that enables and controls user-established, person-to-person sharing of calendar information.

federation

A trust-based agreement between two Exchange organizations to achieve a common purpose.

With federation, both organizations want authentication assertions from one organization to be recognized by the other.

federation trust

A relationship with the Windows Azure AD authentication system that defines the following components for your Exchange organization:

- Account namespace
- Application identifier (AppID)
- Organization identifier (OrgID)
- Federated domains

To configure federated sharing with other federated Exchange organizations, a federation trust must be established with the Windows Azure AD authentication system.

non-federated organization

Organizations that don't have a federation trust established with the Windows Azure AD authentication system.

organization identifier (OrgID)

Defines which of the authoritative accepted domains configured in an organization are enabled for federation. Only recipients that have e-mail addresses with federated domains configured in the OrgID are recognized by the Windows Azure AD authentication system and are able to use federated sharing features. The OrgID is a combination of a pre-defined string and the first accepted domain selected for federation in the **Enable federation trust** wizard. For example, if you specify the federated domain contoso.com as your organization's primary SMTP domain, the account namespace FYDIBOHF25SPDLT.contoso.com will be automatically created as the OrgID for the federation trust.

organization relationship

A one-to-one relationship between two federated Exchange organizations that allows recipients to share free/busy (calendar availability) information. An organization relationship requires a federation trust with the Windows Azure AD authentication system and replaces the need to use Active Directory forest or domain trusts between Exchange organizations.

Windows Azure AD authentication system

A free, cloud-based identity service that acts as the trust broker between federated Microsoft Exchange organizations. It's responsible for issuing delegation tokens to Exchange recipients when they request information from recipients in other federated Exchange organizations. To learn more, see Windows Azure Active Directory.

Windows Azure AD authentication system

The Windows Azure AD authentication system, a free cloud-based service offered by Microsoft, acts as the trust broker between your on-premises Exchange 2013 organization and other federated Exchange 2010 and Exchange 2013 organizations. If you want to configure federation in your Exchange organization, you must establish a one-time federation trust with the Windows Azure AD authentication system, so that it can become a federation partner with your organization. With this

trust in place, users authenticated by Active Directory (known as *identity providers*) are issued Security Assertion Markup Language (SAML) delegation tokens by the Windows Azure AD authentication system. These delegation tokens allow users from one federated Exchange organization to be trusted by another federated Exchange organization. With the Windows Azure AD authentication system acting as the trust broker, organizations aren't required to establish multiple individual trust relationships with other organizations, and users can access external resources using a single sign-on (SSO) experience. For more information, see Windows Azure Active Directory.

Return to top

Federation trust

To use Exchange 2013 federated sharing features, you must establish a federation trust between your Exchange 2013 organization and the Windows Azure AD authentication system. Establishing a federation trust with the Windows Azure AD authentication system exchanges your organization's digital security certificate with the Windows Azure AD authentication system and retrieves the Windows Azure AD authentication system certificate and federation metadata. You can establish a federation trust by using the **Enable federation trust** wizard in the Exchange Administration Center (EAC) or the **New-FederationTrust** cmdlet in the Exchange Management Shell. A self-signed certificate is automatically created by the **Enable federation trust** wizard and is used for signing and encrypting delegation tokens from the Windows Azure AD authentication system that allow users to be trusted by external federated organizations. For details about certificate requirements, see Certificate Requirements for Federation later in this topic.

When you create a federation trust with the Windows Azure AD authentication system, an *application identifier* (AppID) is automatically generated for your Exchange organization and provided in the output of the **Get-FederationTrust** cmdlet. The AppID is used by the Windows Azure AD authentication system to uniquely identify your Exchange organization. It's also used by the Exchange organization to provide proof that your organization owns the domain for use with the Windows Azure AD authentication system. This is done by creating a text (TXT) record in the public Domain Name System (DNS) zone for each federated domain.

Return to top

Federated organization identifier

The *federated organization identifier* (OrgID) defines which of the authoritative accepted domains configured in your organization are enabled for federation. Only recipients that have e-mail addresses with accepted domains configured in the OrgID are recognized by the Windows Azure AD authentication system and are able to use federated sharing features. When you create a new federation trust, an OrgID is automatically created with the Windows Azure AD authentication system. This OrgID is a combination of a pre-defined string and the accepted domain selected as

the primary shared domain in the wizard. For example, in the Edit Sharing-Enabled Domains wizard, if you specify the federated domain **contoso.com** as the primary shared domain in your organization, the **FYDIBOHF25SPDLT.contoso.com** account namespace will be automatically created as the OrgID for the federation trust for your Exchange organization.

Although typically the primary SMTP domain for the Exchange organization, this domain doesn't have to be an accepted domain in your Exchange organization and doesn't require a domain name system (DNS) proof of ownership TXT record. The only requirement is that accepted domains selected to be federated are limited to a maximum of 32 characters. The only purpose of this subdomain is to serve as the federated namespace for the Windows Azure AD authentication system to maintain unique identifiers for recipients that request SAML delegation tokens. For more information about SAML tokens, see SAML Tokens and Claims

You can add or remove accepted domains from the federation trust at any time. If you want to enable or disable all federation sharing features in your organization, all you have to do is enable or disable the OrgID for the federation trust.

Important:
If you change the OrgID, accepted domains, or the AppID used for the federation trust, all federation sharing features are affected in your organization. This also affects any external federated Exchange organizations, including Exchange Online and hybrid deployment configurations. We recommend that you notify all external federated partners of any changes to these federation trust configuration settings.

Return to top

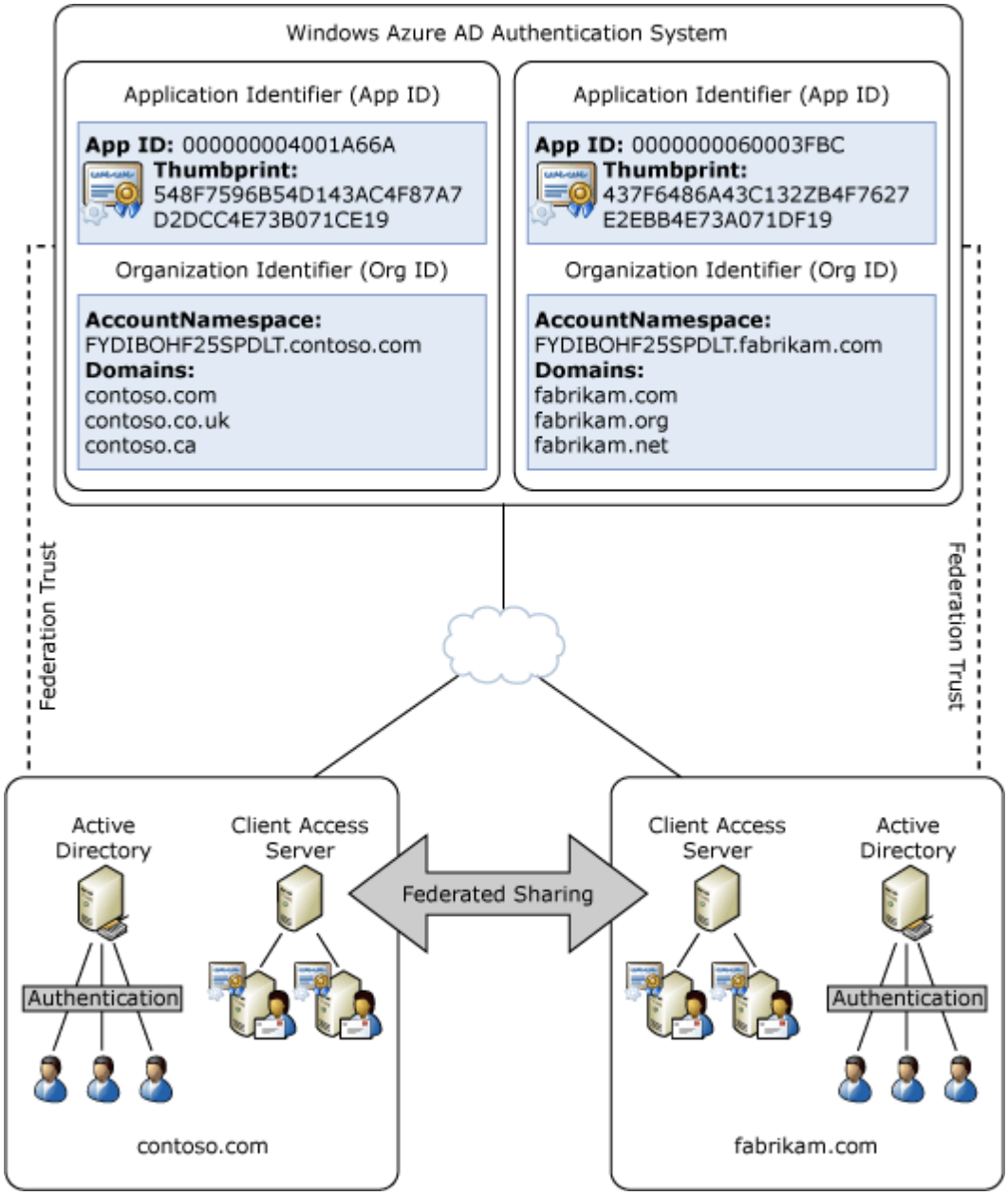
Federation example

Two Exchange organizations, Contoso, Ltd. and Fabrikam, Inc., want their users to be able to share calendar free/busy information with each other. Each organization creates a federation trust with the Windows Azure AD authentication system and configures its account namespace to include the domain used for its user's e-mail address domain.

Contoso employees use one of the following e-mail address domains: contoso.com, contoso.co.uk, or contoso.ca. Fabrikam employees use one of the following e-mail address domains: fabrikam.com, fabrikam.org, or fabrikam.net. Both organizations make sure that all accepted e-mail domains are included in the account namespace for their federation trust with the Windows Azure AD authentication system. Rather than requiring a complex Active Directory forest or domain trust configuration between the two organizations, both organizations configure an organization relationship with each other to enable calendar free/busy sharing.

The following figure illustrates the federation configuration between Contoso, Ltd. and Fabrikam, Inc.

Federated sharing example



Certificate requirements for Federation

To establish a federation trust with the Windows Azure AD authentication system, either a self-signed certificate or an X.509 certificate signed by a certification authority (CA) must be created and installed on the Exchange 2013 server used to create the trust. We strongly recommend using a self-signed certificate, which is automatically created and installed using the **Enable federation trust** wizard in the EAC. This certificate is used only to sign and encrypt delegation tokens used for federated sharing and only one certificate is required for the federation trust. Exchange 2013 automatically distributes the certificate to all other Exchange 2013 servers in the organization.

If you want to use an X.509 certificate signed by an external CA, the certificate must meet the following requirements:

- **Trusted CA** If possible, the X.509 Secure Sockets Layer (SSL) certificate should be issued from a CA trusted by Windows Live. However, you can use certificates issued by CAs that aren't currently certified by Microsoft. For a current list of trusted CAs, see Trusted root certification authorities for federation trusts.

- **Subject key identifier** The certificate must have a subject key identifier field. Most X.509 certificates issued by commercial CAs have this identifier.
- **CryptoAPI cryptographic service provider (CSP)** The certificate must use a CryptoAPI CSP. Certificates that use Cryptography API: Next Generation (CNG) providers aren't supported for federation. If you use Exchange to create a certificate request, a CryptoAPI provider is used. For more information, see Cryptography API: Next Generation.
- **RSA signature algorithm** The certificate must use RSA as the signature algorithm.
- **Exportable private key** The private key used to generate the certificate must be exportable. You can specify that the private key be exportable when you create the certificate request using the **New Exchange certificate** wizard in the EAC or the New-ExchangeCertificate cmdlet in the Shell.
- **Current certificate** The certificate must be current. You can't use an expired or revoked certificate to create a federation trust.
- **Enhanced key usage** The certificate must include the enhanced key usage (EKU) type **Client Authentication (1.3.6.1.5.5.7.3.2)**. This usage type is used to prove your identity to a remote computer. If you use the EAC or the Shell to generate a certificate request, this usage type is included by default.

Note:
Because the certificate isn't used for authentication, it doesn't have any subject name or subject alternative name requirements. You can use a certificate with a subject name that's the same as the host name, the domain name, or any other name.

Return to top

Transitioning to a new certificate

The certificate used to create the federation trust is designated as the current certificate. However, you may need to install and use a new certificate for the federation trust periodically. For example, you may need to use a new certificate if the current certificate expires or to meet a new business or security requirement. To ensure a seamless transition to a new certificate, you must install the new certificate on your Exchange 2013 server and configure the federation trust to designate it as the new certificate. Exchange 2013 automatically distributes the new certificate to all other Exchange 2013 servers in the organization. Depending on your Active Directory topology, distribution of the certificate may take a while. You can verify the certificate status using the Test-FederationTrustCertificate cmdlet in the Shell.

After you verify the certificate's distribution status, you can configure the trust to use the new certificate. After switching certificates, the current certificate is designated as the previous certificate, and the new certificate is designated as the current certificate. The new certificate is published to the Windows Azure AD authentication system, and all new tokens exchanged with the Windows Azure AD authentication system are encrypted using the new certificate.

Note:
This certificate transition process is used only by federation. If you use the same certificate for other Exchange 2013 features that require certificates, you must take the feature requirements

into consideration when planning to procure, install, or transition to a new certificate.

Return to top

Firewall Considerations for Federation

Federation features require that the Mailbox and Client Access servers in your organization have outbound access to the Internet by using HTTPS. You must allow outbound HTTPS access (port 443 for TCP) from all Exchange 2013 Mailbox and Client Access servers in the organization.

For an external organization to access your organization's free/busy information, you must publish one Client Access server to the Internet. This requires inbound HTTPS access from the Internet to the Client Access server. Client Access servers in Active Directory sites that don't have a Client Access server published to the Internet can use Client Access servers in other Active Directory sites that are accessible from the Internet. The Client Access servers that aren't published to the Internet must have the external URL of the Web services virtual directory set with the URL that's visible to external organizations.

Return to top

Trusted root certification authorities for federation trusts

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-09*

To establish a federation trust between your Microsoft Exchange Server 2013 organization and the Windows Azure Active Directory authentication system, you need a digital certificate installed on the Exchange server used to create the trust. We strongly recommend using a self-signed certificate. A self-signed certificate is created and installed automatically when using the **Enable federation trust** wizard in the Exchange Administration Center (EAC).

If you don't want to use the recommended self-signed certificate, you should request and install an X.509 Secure Sockets Layer (SSL) certificate from a certification authority (CA) trusted by Microsoft. Although certificates issued by other CAs may also be used to establish a federation trust with the Windows Azure AD authentication system, they aren't certified by Microsoft to date.

The following table lists CAs currently trusted Microsoft. These CAs have been tested for use with Exchange 2013.

CA friendly name	Issued by	Intended purposes

Comodo	Comodo Certification Authority	Server authentication, client authentication
Digicert	Digicert Global Root Certification Authority	Server authentication, client authentication
Digicert High Assurance EV	Digicert Global Root Certification Authority	Server authentication, client authentication
Entrust	Entrust.net Secure Server Certification Authority	Server authentication, client authentication
Entrust (2048)	Entrust.net Secure Server Certification Authority	Server authentication, client authentication
Equifax	Equifax Secure Certification Authority	Server authentication, client authentication
GlobalSign	GlobalSign Certification Authority	Server authentication, client authentication
Go Daddy	Go Daddy Class 2 Certification Authority	Server authentication, client authentication
Network Solutions	Network Solutions Certification Authority	Server authentication, client authentication
PositiveSSL	Comodo Certification Authority	Server authentication, client authentication
SECOM	SECOM Trust Systems Certification Authority	Server authentication, client authentication
UTN-UserFirst-Hardware	Comodo Certification Authority	Server authentication, client authentication
VeriSign	Class 3 Public Primary Certification Authority	Server authentication, client authentication

VeriSign	VeriSign Trust Network	Server authentication, client authentication

For more information about certificate requirements for Federation, see Federation.

Federation procedures

Exchange Server 2013 > Sharing > Federation >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-07-10*

Configure federated sharing

Configure a federation trust

Manage a federation trust

Remove a federation trust

Configuring federated sharing between Exchange organizations

Disable or Re-enable federated sharing for your Exchange organization

Important:

This feature of Exchange Server 2013 isn't fully compatible with Office 365 operated by 21Vianet in China and some feature limitations may apply. For more information, see Learn about Office 365 operated by 21Vianet.

Configure federated sharing

Sharing > Federation > Federation procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-15*

With federated sharing in Exchange Server 2013, users can share information with recipients in external federated organizations. This includes sharing their free/busy (also known as calendar availability) information for scheduling purposes or, depending on the nature of the business relationship, sharing more detailed calendar information. To learn more about federation sharing, see Sharing.

Important:

This feature of Exchange Server 2013 isn't fully compatible with Office 365 operated by 21Vianet in China and some feature limitations may apply. For more information, see Learn about Office 365 operated by 21Vianet.

What do you need to know before you begin?

- Estimated time to complete this task: 1 hour.
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Create and configure a federation trust

A federation trust establishes a trust relationship between an Exchange 2013 organization and the Windows Azure AD authentication system and is a requirement for federated sharing.

For detailed instructions, see Configure a federation trust.

Step 2: Create an organization relationship

An organization relationship enables users in your Exchange organization to share calendar free/busy information as part of federated sharing with other federated Exchange organizations. Federated sharing can be configured between two federated Exchange 2013 organizations or between a federated Exchange 2013 organization and federated Exchange 2010 organizations.

For detailed instructions, see Create an organization relationship.

Step 3: Create a sharing policy

Sharing policies enable user-established, people-to-people sharing of calendar information with different types of external users. They support the sharing of calendar and contact information with external federated organizations, external non-federated organizations, and individuals with Internet access. If you don't need to configure people-to-people or contact sharing (organization-level sharing only), you don't need to configure a sharing policy.

For detailed instructions, see Create a sharing policy.

Step 4: Configure an Autodiscover public DNS record

You need to add an alias canonical name (CNAME) resource record to your public-facing DNS. The new CNAME record should point to an Internet-facing Exchange 2013 Client Access server that's running the Autodiscover service.

For detailed instructions about how to add CNAME records, see the host service for your public DNS records. Typically this is an Internet-based service that may also host your domain website.

Configure a federation trust

Sharing > Federation > Federation procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-21*

A federation trust establishes a trust relationship between a Microsoft Exchange 2013 organization and the Windows Azure Active Directory authentication system. By configuring a federation trust, you can configure federated sharing with other federated Exchange organizations to share calendar free/busy information among recipients. Federated sharing can be configured between two federated Exchange 2013 organizations or between a federated Exchange 2013 organization and federated Exchange 2010 organizations. You can also set up sharing with an Office 365 organization.

Note:

Creating a federation trust is one of several steps in setting up federated sharing in your Exchange organization. To review all the steps, see Configure federated sharing.

For additional management tasks related to federation, see Federation procedures.

Important:

This feature of Exchange Server 2013 isn't fully compatible with Office 365 operated by 21Vianet in China and some feature limitations may apply. For more information, see Learn about Office 365 operated by 21Vianet.

What do you need to know before you begin?

- Estimated time to complete: 30 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Federation and certificates" permissions entry in the Exchange and Shell infrastructure permissions topic.
- The domain used for establishing a federation trust should be resolvable from the Internet. This requires that the domain be registered with a domain registrar and the Domain Name System

(DNS) zone for the domain to be hosted on a DNS server accessible from the Internet. If the organization receives Internet email for the domain, these requirements are already met.

- You will need to add a TXT record to your public DNS. Review the requirements for adding a TXT record with the organization that hosts your public DNS records.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.
- Both Exchange organizations in a federated sharing relationship must use the same Windows Azure AD authentication system for their federation trusts. This requirement applies when configuring federated sharing between two on-premises Exchange organizations or between an on-premises Exchange organization and an Exchange organization hosted by Office 365.
- When you create a federation trust with the Windows Azure AD authentication system for your Exchange 2013 organization, the federation trust will use the business instance of the Windows Azure AD authentication system. However, other federated Exchange organizations with previous versions of Exchange and existing federation trusts may be using either the business or consumer instance of the Windows Azure AD authentication system.

The following Exchange organizations use the business instance of the Windows Azure AD authentication system by default:

- Exchange 2013 organizations by using the **Enable federation trust** wizard and self-signed certificates for a federation trust.
- Exchange 2010 SP1 or later organizations by using the **New Federation Trust** wizard and self-signed certificates for a federation trust.
- Exchange organizations hosted by Office 365, such as the Exchange Online.

The following Exchange organizations use the consumer instance of the Windows Azure AD authentication system by default:

- Release to manufacturing (RTM) version of Exchange 2010 organizations using certificates issued by third-party certification authorities.

We recommend that all Exchange organizations use the business instance of the Windows Azure AD authentication system for federation trusts. Before configuring federated sharing between the two Exchange organizations, you need to verify which Windows Azure AD authentication system instance each Exchange organization is using for any existing federation trusts. To determine which Windows Azure AD authentication system instance an Exchange organization is using for an existing federation trust, run the following Shell command.

```
Get-FederationInformation -DomainName <hosted Exchange domain namespace>
```

The business instance returns a value of `<uri:federation:MicrosoftOnline>` for the *TokenIssuerURIs* parameter.

The consumer instance returns a value of `<uri:WindowsLiveID>` for the *TokenIssuerURIs* parameter.

To configure federated sharing with an Exchange organization that has an existing federation trust that's using the business instance of the Windows Azure AD authentication system, follow the steps in this topic. These steps are all you need to perform to create federation trusts that can be used to enable federated sharing between two Exchange 2013 organizations or between an Exchange 2013

organization and an Exchange 2010 organization that's already using the business instance of the Windows Azure AD authentication system.

To configure federated sharing between your Exchange 2013 organization and an Exchange organization that has an existing federation trust that's using the consumer instance of the Windows Azure AD authentication system , the Exchange organization using the consumer instance should install Exchange 2010 SP2 or later, or upgrade to Exchange 2013. If you decide to install Exchange 2010 SP2 or later, use the **New Federation Trust** wizard to remove and re-create the existing federated domains and federation trusts. When the federation trusts are re-created, the business instance of the Windows Azure AD authentication system will be used.

What do you want to do?

Use the EAC to create and configure a federation trust

1. On an Exchange 2013 server in your on-premises organization, navigate to **Organization** > **Sharing**.
2. Click **Enable** to start the **Enable federation trust** wizard.
3. After the wizard completes, click **Close**.
4. In the **Federation Trust** section of the **Sharing** tab, click **Modify**.
5. In **Sharing-Enabled Domains**, next to **Step 1**, click **Browse**.
6. In **Select Accepted Domains**, select the primary shared domain from the list, and then click **OK**.

Note:

The domain you select will be used to configure the OrgID for the federation trust. For more information about the OrgID, see Federation.

7. Make a note of the federated domain proof that's generated for the primary shared domain. You'll use this string to create a TXT record on your public DNS server.

Important:

The federated domain proof is a string of alphanumeric characters. To avoid input errors, we recommend that you copy the string from the EAC, paste it into a text editor such as Notepad. You can then copy it from the text editor to the Clipboard, and then paste it into the **Text** field when creating the TXT record. If the TXT record is created by using an incorrect federated domain proof string, the Windows Azure AD authentication system won't be able to verify proof of domain ownership, and you won't be able to add it to the federated organization identifier.

8. In **Step 2**, click **Add +** to add additional domains to the federated trust for email addresses that will used by users in your organization that require federated sharing features. For example, if you have users that use a subdomain in their email address such as sales.contoso.com, you' would add the sales.contoso.com domain to the federation trust.

Note:

A federated domain proof string will be created for each additional domain selected. You must create separate TXT records on your public DNS for each additional domain.

9. Using the federated domain proof strings created for each domain, create TXT records for each of these domains on your public DNS server. Depending on the update schedule of your public DNS host, replication of DNS changes may take 15 minutes or longer.
10. After the TXT records are created and replicated, click **Update**.

Use the Shell to create and configure a federation trust

1. This example creates a unique subject key identifier to be used with the certificate.

```
$ski = [System.Guid]::NewGuid().ToString("N")
```

2. This example creates a self-signed certificate for the federation trust with the Windows Azure AD authentication system.

```
New-ExchangeCertificate -FriendlyName "Exchange Federated
Sharing" -DomainName $env:USERDNSDOMAIN -Services
Federation -KeySize 2048 -PrivateKeyExportable $true -
SubjectKeyIdentifier $ski
```

3. This example retrieves the self-signed certificate and creates the federation trust "Azure AD authentication". This automatically deploys the self-signed certificate to the Exchange servers in your organization.

```
Get-ExchangeCertificate | ?{$_.friendlyname -eq "Exchange
Federated Sharing"} | New-FederationTrust -Name "Azure AD
authentication"
```

For detailed syntax and parameter information, see New-ExchangeCertificate and New-FederationTrust.

How do you know this worked?

The successful completion of the **Enable federation trust** and **Sharing-Enabled Domains** wizards will be your first indication that the federation trust was configured as expected.

To further verify that you have successfully created and configured the federation trust, do the following:

1. Run the following Shell command to verify the federation trust information.

```
Get-FederationTrust | format-list
```

2. Run the following Shell command to verify that federation information can be retrieved from your organization.

```
Get-FederationInformation -DomainName <your primary sharing
```

domain>

For detailed syntax and parameter information, see Get-FederationTrust and Get-FederationInformation.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Manage a federation trust

Sharing > Federation > Federation procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-15*

A federation trust establishes a trust relationship between a Microsoft Exchange 2013 organization and the Windows Azure Active Directory authentication system and supports federated sharing with other federated Exchange organizations. Normally, you shouldn't have to manage or modify the federation trust after it's created. However, there may be circumstances that require adding or removing federated domains or resetting the domain used to configure the organization identifier (OrgID) for the federation trust.

Note:
Modifying an existing federation trust, especially the primary shared domain used to define the OrgID, can disrupt federated sharing between federated Exchange organizations or for hybrid deployments with Office 365 organizations.

For additional management tasks related to Federation, see Federation procedures.

Important:
This feature of Exchange Server 2013 isn't fully compatible with Office 365 operated by 21Vianet in China and some feature limitations may apply. For more information, see Learn about Office 365 operated by 21Vianet.

What do you need to know before you begin?

- Estimated time to complete: 30 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the *Federation and certificates* permissions entry in the Exchange and Shell infrastructure permissions topic.
- You will need to add a TXT record to your public DNS for each new federated domain added to the federation trust. Review the requirements for adding a TXT record with the organization that hosts your public DNS records.

- For the purposes of this topic, an existing federation trust was configured with the following settings:
  - **Contoso.com** is the primary shared domain for the federation trust. (This domain will not be changed.)
  - The federated domains **service.contoso.com** and **sales.contoso.com** are included in the existing federation trust.
  - **Marketing.contoso.com** is an accepted domain in the Exchange organization.
- This topic also covers other federation management tasks, such as viewing and managing certificates used for the federation trust and viewing federation trust parameter information in the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to manage a federation trust

1. On an Exchange 2013 server in your on-premises organization, navigate to the **Organization** > **Sharing**.
2. In the **Federation Trust** section, click **Modify**.
3. In **Sharing-Enabled Domains**, skip **Step 1** because the primary sharing domain isn't changing.
4. In **Step 2**, select the **service.contoso.com** domain and then click **Remove —** to remove the domain from the federated trust.
5. In **Step 2**, click **Add +**.
6. In **Select Accepted Domains**, select **marketing.contoso.com** from the list of accepted domains, and then click **OK** to add the domain to the federated trust.

Important:
A federated domain proof string will be created for the **marketing.contoso.com** domain. You must create separate TXT record on your public DNS for this domain.

7. Using the federated domain proof string created for the **marketing.contoso.com** domain, create a TXT record on your public DNS server. Depending on the update schedule of your public DNS host, replication of DNS changes may take 15 minutes or longer.
8. After the TXT record is created and replicated, click **Update**.

Use the Shell to manage a federation trust

1. This example removes the service.contoso.com domain from the federation trust.

```
Remove-FederatedDomain -DomainName service.contoso.com
```

2. This example adds the marketing.contoso.com domain to the federation trust.

```
Add-FederatedDomain -DomainName marketing.contoso.com
```

For detailed syntax and parameter information, see Remove-FederatedDomain and Add-FederatedDomain.

Run the following Shell commands to manage other aspects of a federation trust:

1. View the federated OrgID and federated domains

This example displays the Exchange organization's federated OrgID and related information, including federated domains and status.

```
Get-FederatedOrganizationIdentifier
```

2. View federation trust certificates

This example displays the previous, current, and next certificates used by the federation trust "Azure AD authentication".

```
Get-FederationTrust "Azure AD authentication" | Select Org*certificate
```

3. Check federation certificates status

This example displays the state of federation certificates on all Mailbox and Client Access servers in the organization.

```
Test-FederationTrustCertificate
```

4. Configure the federation trust to use a certificate as the next certificate

This example configures the federation trust "Azure AD authentication" to use the certificate with the provided thumbprint as the next certificate. After the certificate is deployed to all Exchange servers in the organization, you can use the *PublishCertificate* switch to configure the federation trust to use this certificate as the current certificate.

```
Set-FederationTrust "Azure AD authentication" -Thumbprint AC00F35CBA8359953F4126E0984B5CCAFA2F4F17
```

5. Configure the federation trust to use the next certificate as the current certificate

This example configures the federation trust Azure AD authentication to use the next certificate as the current certificate and publishes it to the Windows Azure AD authentication system.

```
Set-FederationTrust "Azure AD authentication" -PublishFederationCertificate
```

Caution:

Before configuring the federation trust to use the next certificate as the current federation certificate, make sure that the certificate is deployed on all Exchange servers in your organization. Use the Test-FederationTrustCertificate cmdlet to check the deployment status of the certificate.

6. Refresh federation metadata and certificate from the Windows Azure AD authentication system

This example refreshes the federation metadata and certificate of the Windows Azure AD authentication system for the federation trust Azure AD authentication.

```
Set-FederationTrust "Azure AD authentication" -RefreshMetadata
```

For detailed syntax and parameter information, see the following topics:

- Get-FederatedOrganizationIdentifier
- Get-FederationTrust
- Test-FederationTrustCertificate
- Set-FederationTrust

How do you know this worked?

The successful completion of the **Sharing-enabled domains** wizard is your first indication that you configured the federation trust as expected.

To further verify success, do the following:

1. Run the following Shell command to verify the federation trust information.

```
Get-FederationTrust | format-list
```

2. Run the following Shell command to verify that federation information can be retrieved from your organization. For example, verify that the sales.contoso.com and marketing.contoso.com domains are returned in the *DomainNames* parameter.

```
Get-FederationInformation -DomainName <your primary sharing domain>
```

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Remove a federation trust

Sharing > Federation > Federation procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-17*

A federation trust establishes a trust relationship between a Microsoft Exchange 2013 organization and the Windows Azure Active Directory authentication system and supports sharing with other federated Exchange organizations. Removing a federation trust from your on-premises Exchange

organization will disable federated sharing with other federated Exchange organizations and with Office 365 organizations connected to your organization as part of a hybrid deployment. You should carefully consider the overall impact to your organization before removing a federation trust.

For additional management tasks related to federation trusts, see Federation procedures.

Important:
This feature of Exchange Server 2013 isn't fully compatible with Office 365 operated by 21Vianet in China and some feature limitations may apply. For more information, see Learn about Office 365 operated by 21Vianet.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Federation and certificates" permissions entry in the Exchange and Shell infrastructure permissions topic.
- After removing the federation trust, you can remove the TXT records from your public DNS server for each federated domain. Review the requirements for removing a TXT record with the organization that hosts your public DNS records.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to remove a federation trust

1. On an Exchange 2013 server in your on-premises organization, navigate to **organization** > **sharing**.
2. In the **Federation Trust** section, click **Remove**.
3. In the warning, click **yes** to confirm that you want to remove the federation trust.
4. After the federation trust is removed, click **Close**.

Use the Shell to remove a federation trust

This example removes the federation trust.

```
Remove-FederationTrust
```

For detailed syntax and parameter information, see Remove-FederationTrust.

How do you know this worked?

To verify that you have successfully removed the federation trust, do one of the following:

- In the EAC, navigate to **organization** > **sharing**. If you successfully removed the federation trust, only the **Enable** button will be available under **Federation Trust**.
- In the Shell, run the following command to verify that federation trust information isn't returned for your Exchange organization.

```
Get-FederationTrust
```

For detailed syntax and parameter information, see Get-FederationTrust.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Disable or Re-enable federated sharing for your Exchange organization

Sharing > Federation > Federation procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-17*

There may be situations when you need to temporarily disable federated sharing for your organization. Instead of deleting the existing federation trust or deleting organization relationships and sharing policies that you may need in the future, you can simply disable the organization identifier (OrgID) for the federation trust.

Caution:
For hybrid deployments with Office 365, disabling the federation trust for your on-premises servers will also disable hybrid features such as shared calendar free/busy information, MailTips, and message tracking. However, secure mail transport won't be disabled in the hybrid deployment if the federation trust for the on-premises organization is disabled.

To learn more about federation trusts, see Federation. To learn more about federated sharing, see Sharing.

For additional management tasks related to federated sharing, see Federation procedures.

Important:
This feature of Exchange Server 2013 isn't fully compatible with Office 365 operated by 21Vianet in China and some feature limitations may apply. For more information, see Learn about Office 365 operated by 21Vianet.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the *Federation and certificates* permissions entry in the Exchange and Shell infrastructure permissions topic.
- Any existing organization relationships and sharing policies for other federated Exchange organizations won't be modified and won't be functional. Sharing policies that are configured to provide Internet recipients with access to calendar information won't be affected.
- You can't use the Exchange Administration Center (EAC) to disable or enable the OrgID for a federation trust. You must use the Shell.

Use the Shell to disable or re-enable federated sharing

This example disables the OrgID and disables federation and federated sharing for the Exchange organization.

```
Set-FederatedOrganizationIdentifier -Enabled $false
```

This example enables the OrgID and re-enables federation and federated sharing for the Exchange organization.

```
Set-FederatedOrganizationIdentifier -Enabled $true
```

For detailed syntax and parameter information, see Set-FederatedOrganizationIdentifier.

How do you know this worked?

Successful completion of the **Set-OrganizationIdentifier** cmdlet will be the first indication that the OrgID has been disabled or enabled.

To further verify success, run the following Shell command and verify the value returned for the *Enabled* parameter

```
Get-FederatedOrganizationIdentifier
```

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Configuring federated sharing between Exchange organizations

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-07-10*

With federated sharing, users in your on-premises Exchange organization can share free/busy calendar information with recipients in other Exchange organizations that are also configured for federated sharing. Free/busy sharing can be enabled between two organizations running Exchange 2013 and also between organizations with a mixed Exchange deployment. To learn more about federated sharing, see Sharing.

This topic provides a summary of the requirements and configuration steps necessary to enable free/busy sharing between different types of the following common Exchange deployments:

- Two Exchange 2013 organizations.
- An Exchange 2013 organization and an Exchange 2010 SP2 organization.
- An Exchange 2007 organization (or mixed Exchange 2007 and Exchange 2010 SP2 organization) and an Exchange 2013 organization.
- An Exchange 2003 organization (or mixed Exchange 2003 and Exchange 2010 SP2 organization) and an Exchange 2013 organization.

For additional management tasks related to federated sharing, see Federation procedures.

Important:
This feature of Exchange Server 2013 isn't fully compatible with Office 365 operated by 21Vianet in China and some feature limitations may apply. For more information, see Learn about Office 365 operated by 21Vianet.

What do you need to know before you begin?

- Estimated time to complete each procedure: 2 hours.
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- Before you perform the procedures in this topic, make sure you understand the limitations associated with sharing free/busy information across Exchange organizations. For details, see Limitations of free/busy sharing.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Configure free/busy sharing between Exchange 2013 organizations

Complete the steps in Configure federated sharing for both organizations.

Configure free/busy sharing between Exchange 2013 and Exchange 2010 SP2 organizations

- Configure federated sharing for the Exchange 2013 organization. Complete the steps in Configure federated sharing.
- Configure federated delegation (previous name for federated sharing) for the Exchange 2010 SP2 organization. Complete the steps in Configure federated delegation.

Configure free/busy sharing between Exchange 2013 and Exchange 2007 organizations

- Configure federated sharing for the Exchange 2013 organization. Complete the steps in Configure federated sharing.
- Complete the following steps in the Exchange 2007 organization:

1. **Add an Exchange 2010 SP2 server**

An Exchange 2010 SP2 server with the Client Access server role must be installed in the Exchange 2007 organization. If you have existing Exchange 2010 servers, they should also be updated to Exchange 2010 SP2. For information about installing Exchange 2010 in an Exchange 2007 organization, see Exchange 2007 - Planning Roadmap for Upgrade and Coexistence.

2. **Configure federated delegation**

Configure federated delegation for the Exchange 2007 organization. On an Exchange 2010 SP2 server in the Exchange 2007 organization, complete the steps in Configure federated delegation.

3. **Configure Active Directory synchronization**

Active Directory synchronization must be configured for all users that need to share free/busy information between the organizations. You can either configure the Active Directory synchronization manually or use an automated Active Directory synchronization service. To configure Active Directory synchronization, see the steps below:

- **Prerequisites** Make sure your organization meets the requirements for installing Active Directory synchronization.

To learn more, see Prepare for Active Directory Synchronization

- **Plan** Understand the Microsoft Online Services Directory Synchronization tool and installation roadmap.

To learn more, see Active Directory Synchronization: Roadmap

- **Install and Configure** Configure Active Directory synchronization between your on-premises organization and the Office 365 tenant service organization.

To learn more, see Install and Upgrade the Microsoft Online Services Directory Synchronization tool

4. **Create an availability address space**

Create an availability address space for the remote Exchange 2013 organization that directs availability requests from Exchange 2007 mailbox users to the Exchange 2010 SP2 Client Access

server in the Exchange 2007 organization. This setting enables user availability requests from Exchange 2007 users for users in the remote Exchange 2013 organization to be proxied through the Exchange 2010 Client Access server in the Exchange 2007 organization. The Exchange 2010 Client Access server in the Exchange 2007 organization uses the federation trust and organization relationship to send the availability requests to the remote Exchange 2013 organization forest availability endpoint.

To configure the availability address space, on the Exchange 2010 Client Access server in the Exchange 2007 organization, run the following command in the Exchange Management Shell:

```
Add-AvailabilityAddressSpace -AccessMethod InternalProxy -
ProxyUrl https://<Exchange 2010 CAS server name>/ews/
exchange.asmx -ForestName <SMTP domain of the remote
Exchange organization> -UseServiceAccount $True
```

For detailed syntax and parameter information, see Add-AvailabilityAddressSpace

Configure Free/Busy Sharing Between Exchange 2013 and Exchange 2003 Organizations

- Configure federated sharing for the Exchange 2013 organization. Complete the steps in Configure federated sharing.
- Complete the following steps in the Exchange 2003 organization:

1. **Add Exchange 2010 SP2 server**.

An Exchange 2010 SP2 server with the Client Access server role must be installed in the Exchange 2003 organization. If you have existing Exchange 2010 servers, they should also be updated to Exchange 2010 SP2. For information about installing Exchange 2010 in an Exchange 2003 organization, see Exchange 2003 - Planning Roadmap for Upgrade and Coexistence.

⚠**Warning:**
For free/busy sharing to work properly between Exchange 2013 and Exchange 2003 organizations, the **OU=EXTERNAL (FYDIBOHF25SPDLT)** public folder must exist in the public folder hierarchy. This folder is automatically created on the Exchange 2010 Mailbox server in the Exchange 2003 organization only if you select the option to create public folders as part of configuring client settings for Outlook 2003 support during Exchange 2010 Setup. Additionally, this option is presented during the setup process only if the Exchange 2010 Mailbox server is the first Mailbox server installed in the organization. If the **OU=EXTERNAL (FYDIBOHF25SPDLT)** public folder wasn't created during Setup, you must manually create it. For details about how to create this public folder, see How to troubleshoot Free/Busy issues when you use Exchange Federation in the Microsoft Office 365 for enterprises environment.

2. **Configure federated delegation**.

Configure federated delegation for the Exchange 2003 organization. On an Exchange 2010 SP2 server in the Exchange 2003 organization, complete the steps in Configure federated delegation.

3. **Configure Active Directory synchronization**.

Active Directory synchronization must be configured for all users that need to share free/busy information between the organizations. You can either configure the Active Directory synchronization manually or use an automated Active Directory synchronization service. To learn more about Active Directory synchronization, see Forefront Identity Management.

- **Prerequisites** Make sure your organization meets the requirements for installing Active Directory synchronization.

To learn more, see Prepare for Active Directory Synchronization

- **Plan** Understand the Microsoft Online Services Directory Synchronization tool and installation roadmap.

To learn more, see Active Directory Synchronization: Roadmap

- **Install and Configure** Configure Active Directory synchronization between your on-premises organization and the Office 365 tenant service organization.

To learn more, see Install and Upgrade the Microsoft Online Services Directory Synchronization tool

4. **Configure public folders for free/busy sharing in your Exchange 2003 organization.**

Complete the following steps on an Exchange 2003 server:

- In Exchange System Manager, in the console tree, navigate to **Administrative Groups** > **First Administrative Group** > **Servers**.
- Select your Exchange 2003 server, and then navigate to **First Storage Group** > **Public Folder Store** > **Public Folders** > **Schedule+ FREE BUSY**.
- In the action pane, select the **OU=EXTERNAL (FYDIBOHF25SPDLT)** folder for the **First Administrative Group**.
- Right-click the **OU=EXTERNAL (FYDIBOHF25SPDLT)** folder, and then click **Properties**.
- In **OU=EXTERNAL (FYDIBOHF25SPDLT) Properties**, select the **Replication** tab.
- To replicate the **OU=EXTERNAL (FYDIBOHF25SPDLT)** folder to the Exchange 2010 Client Access/Mailbox server, click **Add**.
- In **Select a Public Folder Store**, select the **Public Folder Database** for the Exchange 2010 Client Access/Mailbox server, and then click **OK**.

Note:

By default, Exchange uses the replication schedule set on the public folder database.

- Click **OK** to close **OU=EXTERNAL (FYDIBOHF25SPDLT) Properties** and save your changes.
- Complete the same steps above for the **OU=Exchange Administrative Group (FYDIBOHF23SPDLT)** folder.

Warning:

Depending on the size of your public folders, this replication could take several hours to complete.

- After the **OU=EXTERNAL (FYDIBOHF25SPDLT)** and **OU=Exchange Administrative Group (FYDIBOHF23SPDLT)** public folders have replicated to the Exchange 2010 Client Access/Mailbox server, you must remove the replicas for these public folders on the Exchange 2003 server.

5. **Modify the LegacyExchangeDN parameter**

Modify the *LegacyExchangeDN* parameter on all mail-enabled objects in the Exchange 2003

organization that reference the remote Exchange 2013 organization. Change the existing organizational unit (OU) value for the mail-enabled object to **External (FYDIBOHF25SPDLT)**. For example, **LegacyExchangeDN=/o=First Organization/ou=External (FYDIBOHF25SPDLT)/cn=Recipients/cn=User Name**.

To modify mail-enabled objects in the Exchange 2003 organization, you can use either the Active Directory Service Interfaces Editor (ADSI Edit) tool or the Microsoft Exchange Server LegacyDN Utility.

Configure OAuth authentication between Exchange and Exchange Online organizations

Sharing > Federation > Federation procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-07*

Exchange 2013-only hybrid deployments configure OAuth authentication when using the Hybrid Configuration Wizard. For mixed Exchange 2013/2010 and Exchange 2013/2007 hybrid deployments, the new hybrid deployment OAuth-based authentication connection between Office 365 and on-premises Exchange organizations isn't configured by the Hybrid Configuration wizard. These deployments continue to use the federation trust process by default. However, certain Exchange 2013 features are only fully available across your organization by using the new Exchange OAuth authentication protocol.

The new Exchange OAuth authentication process currently enables the following Exchange features:

- Message Rights Management (MRM)
- Exchange In-place eDiscovery
- Exchange In-place Archiving

We recommend that all mixed Exchange organizations that implement a hybrid deployment with Exchange 2013 and Exchange Online configure Exchange OAuth authentication after configuring their hybrid deployment with the Hybrid Configuration Wizard.

Important:
This feature of Exchange Server 2013 isn't fully compatible with Office 365 operated by 21Vianet in China and some feature limitations may apply. For more information, see Learn about Office 365 operated by 21Vianet.

What do you need to know before you begin?

- Estimated time to complete this task: 15 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Federation and certificates" permissions entry in the Exchange and Shell infrastructure permissions topic.
- Completed configuration of your hybrid deployment using the Hybrid Deployment Wizard. For more information, see **Hybrid deployments with Exchange 2013**.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you configure OAuth authentication between your on-premises Exchange and Exchange Online organizations?

Step 1: Create an authorization server object for your Exchange Online organization

For this procedure, you have to specify a verified domain for your Exchange Online organization. This domain should be the same domain used as the primary SMTP domain used for the cloud-based email accounts. This domain is referred as <*your verified domain*> in the following procedure.

Run the following command in the Exchange Management Shell (the Exchange PowerShell) in your on-premises Exchange organization.

```
New-AuthServer -Name "WindowsAzureACS" -AuthMetadataUrl
https://accounts.accesscontrol.windows.net/<your verified
domain>/metadata/json/1
```

Step 2: Enable the partner application for your Exchange Online organization

Run the following command in the Exchange PowerShell in your on-premises Exchange organization.

```
Get-PartnerApplication |  ?{$_.ApplicationIdentifier -eq
```

```
"00000002-0000-0ff1-ce00-000000000000" -and $_.Realm -eq
""} | Set-PartnerApplication -Enabled $true
```

Step 3: Export the on-premises authorization certificate

In this step, you have to run a PowerShell script to export the on-premises authorization certificate, which is then imported to your Exchange Online organization in the next step.

1. Save the following text to a PowerShell script file named, for example, **ExportAuthCert.ps1**.

```
$thumbprint = (Get-AuthConfig).CurrentCertificateThumbprint
if((test-path $env:SYSTEMDRIVE\OAuthConfig) -eq $false)
{
    md $env:SYSTEMDRIVE\OAuthConfig
}
cd $env:SYSTEMDRIVE\OAuthConfig
$oAuthCert = (dir Cert:\LocalMachine\My) | where
{$_.Thumbprint -match $thumbprint}
$certType =
[System.Security.Cryptography.X509Certificates.X509ContentT
ype]::Cert
$certBytes = $oAuthCert.Export($certType)
$CertFile = "$env:SYSTEMDRIVE\OAuthConfig\OAuthCert.cer"
[System.IO.File]::WriteAllBytes($CertFile, $certBytes)
```

2. In Exchange PowerShell in your on-premises Exchange organization, run the PowerShell script that you created in the previous step. For example:

```
.\ExportAuthCert.ps1
```

Step 4: Upload the on-premises authorization certificate to Windows Azure Active Directory ACS

Next, you have to use Windows PowerShell to upload the on-premises authorization certificate that you exported in the previous step to Windows Azure Active Directory Access Control Services (ACS). To do this, the Windows Azure Active Directory (AD) Module for Windows PowerShell cmdlets has to be installed. If it's not installed, go to http://aka.ms/aadposh to install the Windows Azure AD Module. Complete the following steps after the Windows Azure AD Module is installed.

1. Click the **Windows Azure Active Directory Module for Windows PowerShell** shortcut to open a Windows PowerShell workspace that has the Windows Azure AD cmdlets installed. All commands in this step will be run using the Windows PowerShell for Windows Azure Active Directory console.

2. Save the following text to a PowerShell script file named, for example, **UploadAuthCert.ps1**.

```
Connect-MsolService;
Import-Module msonlineextended;
$CertFile = "$env:SYSTEMDRIVE\OAuthConfig\OAuthCert.cer"
$objFSO = New-Object -ComObject Scripting.FileSystemObject;
$CertFile = $objFSO.GetAbsolutePathName($CertFile);
$cer = New-Object
System.Security.Cryptography.X509Certificates.X509Certifica
te
$cer.Import($CertFile);
$binCert = $cer.GetRawCertData();
$credValue = [System.Convert]::ToBase64String($binCert);
$ServiceName = "00000002-0000-0ff1-ce00-000000000000";
$p = Get-MsolServicePrincipal -ServicePrincipalName
$ServiceName
New-MsolServicePrincipalCredential -AppPrincipalId
$p.AppPrincipalId -Type asymmetric -Usage Verify -Value
$credValue
```

3. Run the PowerShell script that you created in the previous step. For example:

```
.\UploadAuthCert.ps1
```

4. After you start the script, a credentials dialog box is displayed. Enter the credentials for the tenant administrator account in your Microsoft Online Windows Azure AD organization. After running the script, leave the Windows PowerShell for Windows Azure Active Directory session open. You will use this to run a PowerShell script in the next step.

Step 5: Register all hostname authorities for your external on-premises Exchange HTTP endpoints with Windows Azure Active Directory

You have to run the script in this step for each endpoint in your on-premises Exchange organization that is publically accessible. We recommended that you use wild cards, if possible. For example, assume that Exchange is externally available on **https://mail.contoso.com/ews/exchange.asmx**. In this case a single wildcard could be used: ***.contoso.com**. This would cover autodiscover.contoso.com and mail.contoso.com endpoints. However, it doesn't cover the top-level domain, **contoso.com**. In cases where your Exchange 2013 Client Access servers are externally accessible with the top-level hostname authority, this hostname authority must also be registered as **contoso.com**. There isn't a limit for registering additional external hostname authorities.

If you are not sure of the external Exchange endpoints in your on-premises Exchange organization, you can get a list of the external configured Web services endpoints by running the following command in Exchange PowerShell in your on-premises Exchange organization:

```
Get-WebServicesVirtualDirectory | FL ExternalUrl
```

Note:

Successfully running the following script requires that the Windows PowerShell for Windows Azure Active Directory is connected to your Microsoft Online Windows Azure AD tenant, as explained in step 4 in the previous section.

1. Save the following text to a PowerShell script file named, for example, **RegisterEndpoints.ps1**. This example uses a wildcard to register all endpoints for contoso.com. Replace **contoso.com** with a hostname authority for your on-premises Exchange organization.

```
$externalAuthority="*.contoso.com"
$ServiceName = "00000002-0000-0ff1-ce00-000000000000";
$p = Get-MsolServicePrincipal -ServicePrincipalName
$ServiceName;
$spn = [string]::Format("{0}/{1}", $ServiceName,
$externalAuthority);
$p.ServicePrincipalNames.Add($spn);
Set-MsolServicePrincipal -ObjectID $p.ObjectId -
ServicePrincipalNames $p.ServicePrincipalNames;
```

2. In Windows PowerShell for Windows Azure Active Directory, run the PowerShell script that you created in the previous step. For example:

```
.\RegisterEndpoints.ps1
```

Step 6: Create an IntraOrganizationConnector from your on-premises organization to Office 365

You must define a target address for your mailboxes that are hosted in Exchange Online. This target address is created automatically when your Office 365 tenant is created. For example, if your organization's domain hosted in the Office 365 tenant is "contoso.com", your target service address would be "contoso.mail.onmicrosoft.com".

Using Exchange PowerShell, run the following cmdlet in your on-premises organization:

```
New-IntraOrganizationConnector -name
ExchangeHybridOnPremisesToOnline -DiscoveryEndpoint
https://outlook.office365.com/autodiscover/autodiscover.svc
```

```
-TargetAddressDomains <your service target address>
```

Step 7: Create an IntraOrganizationConnector from your Office 365 tenant to your on-premises Exchange organization

You must define a target address for your mailboxes that are hosted in your on-premises organization. If you organization's primary SMTP adddress is "contoso.com", this would be "contoso.com".

You must also define the external Autodiscover endpoint for your on-premises organization. If your company is "contoso.com" this is usually either of the following:

- https://autodiscover.<your primary SMTP domain>/autodiscover/autodiscover.svc
- https://<your primary SMTP domain>/autodiscover/autodiscover.svc

Note:
You can use the Get-IntraOrganizationConfiguration cmdlet in both your on-premises and Office 365 tenants to determine the endpoint values needed by New-IntraOrganizationConnector cmdlet.

Using Windows PowerShell, run the following cmdlet:

```
$UserCredential = Get-Credential
$Session = New-PSSession -ConfigurationName
Microsoft.Exchange -ConnectionUri https://
outlook.office365.com/powershell-liveid/ -Credential
$UserCredential -Authentication Basic -AllowRedirection
Import-PSSession $Session
New-IntraOrganizationConnector -name
ExchangeHybridOnlineToOnPremises -DiscoveryEndpoint <your
on-premises Autodiscover endpoint> -TargetAddressDomains
<your on-premises SMTP domain>
```

Step 8: Configure an AvailabilityAddressSpace for any pre-Exchange 2013 SP1 servers

When you configure a hybrid deployment in a pre-Exchange 2013 organization, you have to install at least one Exchange 2013 SP1 or greater server with the Client Access and Mailbox server roles in your existing Exchange organization. The Exchange 2013 Client Access and Mailbox servers serve as frontend servers and coordinate communications between your existing Exchange on-premises

organization and the Exchange Online organization. This communication includes message transport and messaging features between the on-premises and Exchange Online organizations. We highly recommend installing more than one Exchange 2013 server in your on-premises organization to help increase reliability and availability of hybrid deployment features.

In a mixed deployment with Exchange 2013/2010 or Exchange 2013/2007, it is recommended that all the Internet-facing frontend servers for your on-premises organization are Client Access servers running Exchange 2013 SP1 or greater. All Exchange Web Services (EWS) requests originating from Office 365 and Exchange Online must connect to an Exchange 2013 Client Access server(s) in your on-premises deployment. Additionally, all EWS requests originating in your on-premises Exchange organizations for Exchange Online must be proxied through a Client Access server running Exchange 2013 SP1 or greater. Since these Exchange 2013 Client Access servers have to handle this additional incoming and outgoing EWS requests, it is important to have a sufficient number of Exchange 2013 Client Access servers available to handle the processing load and provide connection redundancy. The number of Client Access servers needed will depend on the average amount of EWS requests and will vary by organization.

Before you complete the following step, make sure:

- The frontend hybrid servers are Exchange 2013 SP1 or greater
- You have a unique external EWS URL for the Exchange 2013 server(s). The Office 365 tenant must connect to these servers in order for cloud-based requests for hybrid features to work correctly.
- The servers have both the Mailbox and Client Access server roles
- Any existing Exchange 2010/2007 Mailbox and Client Access servers have the latest Cumulative Update (CU) or Service Pack (SP) applied.

Note:

Existing Exchange 2010/2007 Mailbox servers can continue to use Exchange 2010/2007 Client Access servers for frontend servers for non-hybrid feature connections. Only hybrid deployment feature requests from the Office 365 tenant need to connect to Exchange 2013 servers.

An *AvailabilityAddressSpace* must be configured on pre-Exchange 2013 Client Access servers that points to the Exchange Web Services endpoint of your on-premises Exchange 2013 SP1 Client Access server(s). This endpoint is the same endpoint as previously outlined in Step 5 or can be determined by running the following cmdlet on your on-premises Exchange 2013 SP1 Client Access server:

```
Get-WebServicesVirtualDirectory | FL
AdminDisplayVersion,ExternalUrl
```

Note:

If virtual directory information is returned from multiple servers, make sure you use the endpoint returned for an Exchange 2013 SP1 Client Access server. It will display 15.0 (Build 847.32) or higher for the *AdminDisplayVersion* parameter.

To configure the *AvailabilityAddressSpace*, use Exchange PowerShell and run the following cmdlet in

your on-premises organization:

```
Add-AvailabilityAddressSpace -AccessMethod InternalProxy -
ProxyUrl <your on-premises External Web Services URL> -
ForestName <your Office 365 service target address> -
UseServiceAccount $True
```

How do you know this worked?

You can verify that the OAuth configuration is correct by using the Test-OAuthConnectivity cmdlet. This cmdlet verifies that the on-premises Exchange and Exchange Online endpoints can successful authenticate requests from each other.

Important:

When connecting to your Exchange Online organization using Remote PowerShell, you may have to use the *AllowClobber* parameter with the **Import-PSSession** cmdlet to import the latest commands in to the local PowerShell session.

To verify that your on-premises Exchange organization can successfully connect to Exchange Online, run the following command in Exchange PowerShell in your on-premises organization:

```
Test-OAuthConnectivity -Service EWS -TargetUri https://
outlook.office365.com/ews/exchange.asmx -Mailbox <On-
Premises Mailbox> -Verbose | fl
```

To verify that your Exchange Online organization can successfully connect to your on-premises Exchange organization, use the **Remote PowerShell** to connect to your Exchange Online organization and run the following command:

```
Test-OAuthConnectivity -Service EWS -TargetUri <external
hostname authority of your Exchange On-Premises deployment>
-Mailbox <Exchange Online Mailbox> -Verbose | fl
```

Important:

You can ignore the "The SMTP address has no mailbox associated with it." error. It's only important that the *ResultTask* parameter returns a value of **Success**. For example, the last section of the test output should read:

```
ResultType: Success
Identity: Microsoft.Exchange.Security.OAuth.ValidationResultNodeId
IsValid: True
ObjectState: New
```

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Organization relationships

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-20*

Set up an organization relationship to share calendar information with an external business partner. Exchange admins can set up an organization relationship with an Office 365 organization or with another Exchange on-premises organization. If you want to share calendars with another on-premises Exchange organization, both on-premises Exchange administrators have to set up an authentication relationship with the cloud (also known as "federation") and must meet minimum software requirements.

An organization relationship is a one-to-one relationship between businesses to allow users in each organization to view calendar availability information. When you set up the organization relationship, you are setting up your side of the relationship and specifying the level of information that the users in the external organization can view. The external organization may set up the same or different settings on their side. For example, if Contoso creates an organization relationship with Tailspin Toys, the users at Tailspin Toys will be able to schedule meetings with the users at Contoso by adding their email address to the meeting invitation. The availability of the invited Contoso user would display to the Tailspin Toys user. However, before Contoso can also see availability for users at Tailspin Toys, their administrator needs to set up an organization relationship with Contoso.

There are three of levels of access that you can specify:

- No access
- Access to availability (free/busy) time only
- Access to free/busy, including time, subject, and location

Note:

If users don't want to share their free/busy information with others, they can change the Default permission entry in Outlook. To do this, users go to the **Calendar Properties** > **Permissions** tab, select the **Default** permission, and select **None** from the **Permission Level** list. Their free/busy information won't be seen by internal or external users, even if an organization relationship exists. The permissions set by the user will apply.

The following topics will help you configure and manage organization relationships as a part of sharing for your organization:

Create an organization relationship

Modify an organization relationship

Remove an organization relationship

Looking for more information about federated sharing? See Sharing.

Create an organization relationship

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-20*

Set up an organization relationship to share calendar information with an external business partner. You can configure an organization relationship between two federated Exchange 2013 organizations or between a federated Exchange 2013 organization and federated Exchange 2010 organizations. You can also set up an organization relationship between your on-premises Exchange organization and an Office 365 organization.

Important:
Creating an organization relationship is one of several steps in setting up federated sharing in your Exchange organization and requires the configuration of a federation trust for your on-premises Exchange organization.

To learn more about federated sharing, see Sharing.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Calendar and Sharing Permissions" section in the Recipients Permissions topic.
- An active federation trust for the on-premises Exchange organization must be configured. For details, see Configure a federation trust.
- The external organization you want to configure in the organization relationship must also have a federation trust established with the Windows Azure AD authentication system. You'll use the primary federated domain for the external Exchange organization when configuring the organization relationship.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to create an organization relationship

1. On an Exchange 2013 server in your on-premises organization, navigate to **organization** > **sharing**.
2. Under **Organization Sharing**, click **New +**.
3. In **new organization relationship**, in the **Relationship name** box, type a friendly name for the

organization relationship.

4. In the **Domains to share with** box, type the federated domain or federated subdomain for the Office 365 or Exchange on-premises organization you want to let see your calendars. If you need to enter multiple domains for the external organization, separate the domains with a comma. For example, **contoso.com, service.contoso.com**.
5. Select the **Enable calendar free/busy information sharing** check box to turn on calendar sharing with the domains you listed. Set the sharing level for calendar free/busy information and set which users can share calendar free/busy information.

To set the free/busy access level, select one of the following:

- **Calendar free/busy information with time only**
- **Calendar free/busy with time, subject, and location**

To set which users will share calendar free/busy information, select one of the following:

- **Everyone in your organization**
- **A specified security group**

To specify a security group, click **browse**.

6. Click **save** to create the organization relationship.

Use the Shell to create an organization relationship

This example creates an organization relationship with Contoso, Ltd with the following conditions:

- The organization relationship is enabled for contoso.com, northamerica.contoso.com, and europe.contoso.com.
- Free/busy access is enabled.
- The requesting organization receives free/busy time, subject, and location information from the target organization.

```
New-OrganizationRelationship -Name "Contoso" -DomainNames
"contoso.com","northamerica.contoso.com","europe.contoso.co
m" -FreeBusyAccessEnabled $true -FreeBusyAccessLevel
LimitedDetails
```

This example attempts to automatically discover configuration information from the external Exchange organization Contoso.com by using the domain names provided in the **Get-FederationInformation** cmdlet. If you use this method to create your organization relationship, you must first make sure that you've created an organization identifier by using the **Set-FederatedOrganizationIdentifier** cmdlet.

```
Get-FederationInformation -DomainName Contoso.com | New-
OrganizationRelationship -Name "Contoso" -
FreeBusyAccessEnabled $true -FreeBusyAccessLevel -
LimitedDetails
```

For detailed syntax and parameter information, see Get-FederationInformation and New-

OrganizationRelationship.

This example creates an organization relationship with Fourth Coffee. In this example, the connection settings with the external Exchange organization are provided. The following conditions apply:

- The organization relationship is established with the domain fourthcoffee.com, a federated domain of Fourth Coffee.
- The Exchange Web Services application URL is mail.fourthcoffee.com.
- The Autodiscover URL is https://mail.fourthcoffee.com/autodiscover/autodiscover.svc/wssecurity.
- Free/busy access is enabled.
- The requesting organization receives only free/busy information with the time.

```
New-OrganizationRelationship -Name "Fourth Coffee" -
DomainNames "fourthcoffee.com" -FreeBusyAccessEnabled $true
-FreeBusyAccessLevel -AvailabilityOnly -
TargetAutodiscoverEpr "https://mail.fourthcoffee.com/
autodiscover/autodiscover.svc/wssecurity" -
TargetApplicationUri "mail.fourthcoffee.com"
```

For detailed syntax and parameter information, see New-OrganizationRelationship.

How do you know this worked?

The successful completion of the **New organization relationship** wizard will be your first indication that the creation of the organization relationship worked as expected.

To further verify that you have successfully created the organization relationship, run the following Shell command to verify the organization relationship information:

```
Get-OrganizationRelationship | format-list
```

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Modify an organization relationship

Exchange Server 2013 > Sharing > Organization relationships >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-15*

An organization relationship lets users in your Exchange organization share calendar free/busy information with an Office 365 organization or another on-premises Exchange organization. You may want to change the settings of an organization relationship, such as changing the name, temporarily disabling calendar sharing, changing the access level, or changing which security groups will share calendars.

Before you can share calendars with another organization, you have to set up an authentication relationship with the cloud (also known as "federation") and must meet minimum software requirements. To learn more about federated sharing, see Sharing.

For additional management tasks related to federation, see Federation procedures.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the *Calendar and Sharing Permissions* entry in the Recipients Permissions topic.
- An active federation trust for the on-premises Exchange organization must be configured.
- The external organization you want to configure in the organization relationship must also have a federation trust established with the Windows Azure AD authentication system.
- The procedures in this topic make changes to an organization relationship named Contoso. The examples show how to:
  - Add a domain named service.contoso.com to the external organization.
  - Disable free/busy sharing for the organization relationship.
  - Change the free/busy access level from *Calendar free/busy information with time, subject, and location* to *Calendar free/busy information with time only*.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to add a domain to an organization relationship

1. On an Exchange 2013 server in your on-premises organization, navigate to **organization** > **sharing**.
2. In list view, under **Organization Sharing**, select the organization relationship Contoso, and then click **Edit** ✎.
3. In **organization relationship**, **general** don't change the **Name** for the organization relationship
4. In the **Domains to share with** box, enter the domain **service.contoso.com**, then click **Add +**.
5. Click **save** to update the organization relationship.

Use the EAC to disable free/busy sharing for the organization relationship

1. Go to **organization** > **sharing**.
2. In list view, under **Organization Sharing**, select the organization relationship Contoso, and then click **Edit** .
3. In **organization relationship**, click **sharing**
4. Select **Calendar free/busy information with time only**.
5. Click **save** to update the organization relationship.

Use the EAC to change the free/busy access level for the organization relationship

1. Go to **organization** > **sharing**.
2. In list view, under **Organization Sharing**, select the organization relationship Contoso, and then click **Edit** .
3. In **organization relationship**, click **sharing**
4. Select **Calendar free/busy information with time only**.
5. Click **save** to update the organization relationship.

Use the Shell to modify the organization relationship

- This example adds the domain name service.contoso.com to the organization relationship Contoso.

```
$domains = (Get-OrganizationRelationship
Contoso).DomainNames
$domains += 'service.contoso.com'
Set-OrganizationRelationship -Identity Contoso -DomainNames
$domains
```

- This example disables the organization relationship Contoso.

```
Set-OrganizationRelationship -Identity Contoso -Enabled
$false
```

- This example enables calendar availability information access for the organization relationship WoodgroveBank and sets the access level to `Availabilityonly` (calendar free/busy information with time only).

```
Set-OrganizationRelationship -Identity Contoso -
FreeBusyAccessEnabled $true -FreeBusyAccessLevel
```

```
AvailabilityOnly
```

For detailed syntax and parameter information, see Get-OrganizationRelationship and Set-OrganizationRelationship.

How do you know this worked?

To verify that you have successfully updated the organization relationship, run the following Shell command and verify the organization relationship information.

```
Get-OrganizationRelationship | format-list
```

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Remove an organization relationship

Exchange Server 2013 > Sharing > Organization relationships >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-15*

An organization relationship lets users in your Exchange organization share calendar free/busy information with an Office 365 organization or with another Exchange on-premises organization. You can remove an organization relationship to disable calendar sharing with the other organization.

Before you can share calendars with another organization, you have to set up an authentication relationship with the Windows Azure AD authentication system (also known as “federation”) and must meet minimum software requirements. To learn more about federated sharing, see Sharing.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the “Calendar and Sharing Permissions” section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to remove an organization relationship

1. On an Exchange 2013 server in your on-premises organization, navigate to **organization** > **sharing**.
2. Under **Organization Sharing**, select an organization relationship, and then click **Delete** to remove organization relationship.
3. In the warning that appears, click **yes**.

Use the Shell to remove an organization relationship

This example removes the organization relationship Contoso from the Exchange organization

```
Remove-OrganizationRelationship -Identity "Contoso"
```

For detailed syntax and parameter information, see Remove-OrganizationRelationship.

How do you know this worked?

To verify that you have successfully removed the organization relationship, do one of the following:

- In the EAC, navigate to **Organization** > **Sharing** and verify that the organization relationship isn't displayed in the list view under **Organization Sharing**.
- Run the following Shell command to verify the organization relationship information is removed.

```
Get-OrganizationRelationship | Format-List
```

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Sharing policies

Exchange Server 2013 > Sharing >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-15*

A part of federated sharing, sharing policies enable user-established, people-to-people sharing of calendar information with different types of external users. Sharing polices are assigned to user mailboxes and allow your users to self-manage and share their free/busy information (including the Calendar folder) with recipients in external Office 365 organizations or other federated Exchange on-premises organizations. If you want to share calendars with recipients that aren't in one of these

types of organizations, sharing policies allow people-to-people sharing of calendar information with any email recipient through the use of Internet Calendar Publishing.

The following topics will help you configure and manage sharing policies as a part of federated sharing for your organization:

Create a sharing policy

Apply a sharing policy to mailboxes

Modify, disable, or remove a sharing policy

Looking for more information about federated sharing? See Sharing

Create a sharing policy

Exchange Server 2013 > Sharing > Sharing policies >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-15*

You can use sharing policies to control how users in your Exchange organization share calendar information with users outside your organization. Sharing policies provide user-established, people-to-people sharing of calendar information with different types of external users. They support the sharing of calendar information with external federated organizations (such as Office 365 or another on-premises Exchange organization), external non-federated organizations, and individuals with Internet access. To apply a specific sharing policy to users, see Apply a sharing policy to mailboxes.

Important:
Creating a sharing policy is one of several steps in setting up federated sharing in your Exchange organization. You have to set up a federation trust with the Windows Azure AD authentication system for your on-premises Exchange organization before you can share calendar information with other federated Exchange organizations. A federation trust isn't required for Internet sharing policies.

To learn more about federated sharing, see Sharing.

For additional management tasks related to federation, see Federation procedures.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Calendar and Sharing Permissions" section in the Recipients Permissions topic.

- The following are required for sharing policies between federated Exchange organizations:
  - An Exchange 2013 Client Access server exists in each Exchange organization. Sharing policies are also supported between Exchange organizations where one organization has Exchange 2013 Client Access servers and the other one organization has Exchange 2010 SP3 or later Client Access servers.
  - Each Exchange organization has created a federation trust with the Windows Azure AD authentication system. For details, see Configure a federation trust.
  - Each Exchange organization has configured a federated organization identifier. Domains used for generating users' e-mail addresses have been added to the organization identifiers.
  - User mailboxes are located on Exchange 2013 Mailbox servers or Exchange 2010 Mailbox servers in each Exchange organization.
  - Only Outlook 2010 or later and Outlook Web App users can create sharing invitations.
- The following are required for sharing policies with non-federated Exchange organizations or individuals:
  - An Exchange 2013 Client Access server exists in the Exchange organization that's sharing user's calendar information.
  - User mailboxes are located on Exchange 2013 Mailbox servers in the Exchange organization that's sharing user's calendar information.
  - The Exchange 2013 Client Access server must be enabled for Outlook Web App access.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to create a sharing policy

1. Navigate to **organization**> **sharing**.
2. In the list view, under **Individual Sharing**, click **New +**.
3. In **new sharing policy**, type a friendly name for the sharing policy in the **Policy name** box.
4. Click **Add +** to specify the sharing rules for the sharing policy.
5. In **sharing rule**, select one of the following options to specify the domains you want to share with:
   - **Sharing with all domains**
   - **Sharing with a specific domain**
6. If you select **Sharing with a specific domain**, type the name of the domain you want to share with. If you need to enter more than one domain for this sharing policy, save the settings for the first domain, then edit the sharing rules to add more domains.
7. To define the calendar sharing levels you want to enforce for the policy, select the **Share your calendar folder** check box, and then select one of the following options:
   - **Calendar free/busy information with time only**
   - **Calendar free/busy information with time, subject, and location**
   - **All calendar appointment information, including time, subject, location and title**

8. Click **save** to set the rules for the sharing policy.
9. If you want to make this sharing policy the default sharing policy for users in your Exchange organization, select the **Make this policy my default sharing policy** check box.
10. Click **save** to create the sharing policy.

Use the EAC to allow all users to share full calendar details

You can edit the default sharing policy to allow all of your users to share full calendar details with people outside of your organization.

1. Navigate to **organization**> **sharing**.
2. In the list view, under **Individual Sharing**, select **the Default Sharing Policy**, and then click **Edit** .
3. In the **Sharing Policy** dialog box, select **Sharing with all domains**, and then click **Edit** .
4. In the **Sharing Rule** dialog box, under **Specify what information you want to share**, select **All calendar appointment information, including time, subject, location and title**, and then click **save**.
5. In the **Sharing Policy** dialog box, click **save** to set the rules for the sharing policy.

Use the Shell to create a sharing policy

- This example creates the sharing policy Contoso for the external federated domain contoso.com. This policy allows users in the contoso.com domain to see your user's detailed calendar availability (free/busy) information. By default, this policy is enabled.

```
New-SharingPolicy -Name "Contoso" -Domains contoso.com:
CalendarSharingFreeBusyDetail
```

- This example creates the sharing policy ContosoWoodgrove for two different federated domains (contoso.com and woodgrovebank.com) with different sharing actions configured for each domain. The policy is disabled.

```
New-SharingPolicy -Name "ContosoWoodgrove" -Domains
'contoso.com: CalendarSharingFreeBusySimple',
'woodgrovebank.com: CalendarSharingFreeBusyDetail -Enabled
$false
```

- This example creates the sharing policy Anonymous for an Exchange organization with the Client Access server CAS01 and the Mailbox server MAIL01 with the sharing action configured for limited calendar availability information. This policy allows users in your Exchange organization to invite users with Internet access to view their calendar availability information by sending them a link. The policy is enabled.
  1. Set the Web proxy URL for MAIL01.

```
Set-ExchangeServer -Identity "Mail01" -InternetWebProxy
```

```
"<Webproxy URL>"
```

2. Enable the publishing virtual directory on CAS01.

```
Set-OwaVirtualDirectory -Identity "CAS01" -ExternalURL
"<URL for CAS01>" -CalendarPublishingEnabled $true
```

3. Create the sharing policy Anonymous and configure limited calendar information sharing.

```
New-SharingPolicy -Name "Anonymous" -Domains 'Anonymous:
CalendarSharingFreeBusySimple' -Enabled $true
```

For detailed syntax and parameter information, see the following topics:

- New-SharingPolicy
- Set-ExchangeServer
- Set-OwaVirtualDirectory

How do you know this worked?

To verify that you have successfully created the sharing policy, run the following Shell command to verify the sharing policy information.

```
Get-SharingPolicy <policy name> | format-list
```

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Apply a sharing policy to mailboxes

Exchange Server 2013 > Sharing > Sharing policies >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-15*

Sharing policies are a part of federated sharing and enable user-established, people-to-people sharing of calendar information with different types of external users. The sharing policy that an admin applies to the user's mailbox determines what level of access a user can share and with whom. If you don't change anything, then the default sharing policy applies to all users. If you create a new sharing policy, you have to apply that policy to mailboxes before it takes effect. A sharing policy can be applied to a single user mailbox or to multiple user mailboxes simultaneously. An admin can also disable a user's sharing policy to prevent external access to calendars.

To learn more about federated sharing, see Sharing.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the *Recipient Provisioning Permissions* entry in the Recipients Permissions topic.
- A sharing policy must exist. For details, see Create a sharing policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

What do you want to do?

Use the EAC to apply a sharing policy to a single mailbox

1. Navigate to **recipients** > **mailboxes**.
2. In the list view, select the mailbox you want, and then click **Edit** .
3. In **User Mailbox**, click **mailbox features**.
4. In the **Sharing policy** list, select the sharing policy you want to apply to this mailbox.
5. Click **save** to apply the sharing policy.

Use the EAC to apply a sharing policy to multiple mailboxes

1. Navigate to **recipients** > **mailboxes**.
2. In the list view, hold the Ctrl key while you select multiple mailboxes..
3. In the details pane, the mailbox properties will be configured for bulk editing. Click **More options**.
4. Under **Sharing policy**, click **Update**.
5. In **bulk assign sharing policy**, use the **Select the sharing policy** list to select a sharing policy to assign to the mailboxes.
6. Click **save** to apply the sharing policy to the selected mailboxes.

Use the Shell to apply a sharing policy to one or more mailboxes

This example applies the sharing policy Contoso to a single mailbox for the user Barbara.

```
Set-Mailbox -Identity Barbara -SharingPolicy "Contoso"
```

This example specifies that all user mailboxes in the Marketing department use the sharing policy Contoso Marketing.

```
Get-Mailbox -Filter {Department -eq "Marketing"} | Set-
Mailbox -SharingPolicy "Contoso Marketing"
```

This example returns all mailboxes that have the sharing policy Contoso applied, and it sorts the users into a table that displays only their aliases and email addresses.

```
Get-Mailbox -ResultSize unlimited | Where {$_.SharingPolicy
-eq "Contoso" } | format-table Alias, EmailAddresses
```

For detailed syntax and parameter information, see Set-Mailbox and Get-Mailbox.

How do you know this worked?

To verify that you have successfully applied the sharing policy to a user mailbox, do one of the following:

- In the EAC, navigate to **Recipients** > **Mailboxes**, and then select the mailbox to which you applied the sharing policy. Click **Edit**, click **mailbox features**, and then confirm that the correct sharing policy appears in the **Sharing policy** list.
- Run the following Shell command to verify the sharing policy was assigned to a user mailbox. Verify that the correct sharing policy is listed in the *SharingPolicy* parameter.

```
Get-Mailbox <user name> | format-list
```

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Modify, disable, or remove a sharing policy

Exchange Server 2013 > Sharing > Sharing policies >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-15*

Sharing policies allow individual users in your Exchange organization to share calendar free/busy information with other federated Exchange organizations, non-federated Exchange organizations, and individual Internet users. During the course of normal operations, you may want to change

some sharing policy properties, such as modifying sharing rules, changing the free/busy access level, temporarily disabling a sharing policy, or removing a sharing policy entirely.

To learn more about federated sharing, see Sharing

For details about how to create a sharing policy, see Create a sharing policy

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Calendar and Sharing Permissions" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the EAC to modify a sharing policy

1. Navigate to **organization** > **sharing**.
2. Under **Individual Sharing**, select a sharing a policy, and then click **Edit** .
3. In **sharing policy**, click **Edit** .
4. In **sharing rule**, modify the sharing rules accordingly. You can change settings such as the domain you want to share information with and the sharing level for calendar appointments. When finished, click **rave** to close the **sharing rules** dialog box.
5. In **sharing policy**, click **save** to update the sharing policy.

Use the EAC to set a sharing policy as the default sharing policy

1. Navigate to **organization** > **sharing**.
2. Under **Individual Sharing**, select a sharing a policy, and then click **Edit** .
3. In **sharing policy**, select the **Make this policy my default sharing policy** check box.
4. Click **save** to update the sharing policy.

Use the EAC to disable a sharing policy

1. Navigate to **Organization** > **Sharing**.
2. Under **Individual Sharing**, select a sharing a policy.

3. In the **On** column, clear the check box for the sharing policy you want to disable.

Use the EAC to remove a sharing policy

Important:
Before you remove a sharing policy, the sharing policy must be removed from all user mailboxes.

1. Navigate to **organization** > **sharing**.
2. Under **Individual Sharing**, select a sharing a policy, and then click **Delete** 🗑.
3. In the warning, click **yes** to delete the sharing policy.

Use the Shell to modify, disable or remove a sharing policy

- This example modifies the sharing policy Contoso for contoso.com, which is a domain outside your organization. This policy allows users in the Contoso domain to see simple free/busy information.

```
Set-SharingPolicy -Identity Contoso -Domains
'sales.contoso.com: CalendarSharingFreeBusySimple'
```

- This example adds a second domain to the sharing policy Contoso. When you're adding a domain to an existing policy, you must include any previously included domains.

```
Set-SharingPolicy -Identity Contoso -Domains 'contoso.com:
CalendarSharingFreeBusySimple', 'atlanta.contoso.com:
CalendarSharingFreeBusyReviewer', 'beijing.contoso.com:
CalendarSharingFreeBusyReviewer'
```

- This example sets the sharing policy Contoso as the default sharing policy.

```
Set-SharingPolicy -Identity Contoso -Default $True
```

- This example disables the sharing policy Contoso.

```
Set-SharingPolicy -Identity "Contoso" -Enabled $False
```

- The first example removes the sharing policy Contoso. The second example removes the sharing policy Contoso and suppresses the confirmation that you want to remove the policy.

```
Remove-SharingPolicy -Identity Contoso

Remove-SharingPolicy -Identity Contoso -Confirm
```

For detailed syntax and parameter information, see Set-SharingPolicy and Remove-SharingPolicy.

Enable Internet calendar publishing

Applies to: *Exchange Online*

Topic Last Modified: *2012-12-10*

Users in Microsoft Exchange Server 2013 organizations can share calendar availability (free/busy) information with users in non-Exchange organizations and other individuals with Internet access. Internet calendar publishing provides increased flexibility and increases the number of users who can share calendar availability information.

Enabling Internet calendar publishing consists of three general steps:

1. Configure the Web proxy URL for the Mailbox server.
2. Enable the publishing virtual directory for the Client Access server.
3. Create a dedicated sharing policy specifically for Internet calendar publishing or update the default sharing policy to support the **Anonymous** domain. Either method allows users in your Exchange organization to invite other users who have Internet access to view limited calendar availability information by accessing a published URL.

For additional management tasks related to sharing policies, see Sharing policies.

What do you need to know before you begin?

- Estimated time to complete this task: 15 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Calendar and Sharing Permissions" entry in the Recipients Permissions topic.
- An Exchange 2013 Client Access server exists in the Exchange organization that's sharing users' calendar information.
- User mailboxes are on Exchange 2013 Mailbox servers in the Exchange organization that's sharing users' calendar information.
- Only Outlook 2010 or later and Outlook Web App users can create sharing invitations.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Use the Shell to configure the Web proxy URL

Note:
You can't use the Exchange Administration Center (EAC) to configure the Web proxy URL.

This example configures a Web proxy URL on Mailbox server MAIL01.

```
Set-ExchangeServer -Identity "MAIL01" -InternetWebProxy
"<Webproxy URL>"
```

For detailed syntax and parameter information, see Set-ExchangeServer.

How do you know this step worked?

To verify that you have successfully configured the Web proxy URL, run the following Shell command and verify the *InternetWebProxy* parameter information.

```
Get-ExchangeServer | format-list
```

Step 2: Use the Shell to enable the publishing virtual directory

Note:
You can't use the EAC to enable the publishing virtual directory.

This example enables the publishing virtual directory on Client Access server CAS01.

```
Set-OwaVirtualDirectory -Identity "CAS01" -ExternalUrl
"<URL for CAS01>" -CalendarEnabled $true
```

For detailed syntax and parameter information, see Set-OwaVirtualDirectory.

How do you know this step worked?

To verify that you have successfully enabled the publishing virtual directory, run the following Shell command and verify the *ExternalURL* parameter information.

```
Get-OwaVirtualDirectory | format-list
```

Step 3, option 1: Use the EAC or the Shell to create a sharing policy specifically for Internet calendar publishing

If you want to create a sharing policy specifically for Internet calendar publishing, complete the

following steps.

Use the EAC

1. Navigate to **Organization**> **Sharing**.
2. In the list view, under **Individual Sharing**, click **New +**.
3. In **Sharing Policy**, type a friendly name for the sharing policy in the **Policy name** field (for example, **Internet**).
4. Click **Add +** to define the sharing rules for the sharing policy.
5. In **Sharing Rule**, click **Sharing with a specific domain**, and then type **Anonymous** in the corresponding box.
6. To specify the calendar sharing levels you want to enforce for the sharing policy, select the **Share your calendar folder** check box, and then select one of the following:
   - **Calendar free/busy information with time only**
   - **Calendar free/busy information with time, subject, and location**
   - **All calendar appointment information, including time, subject, location and title**
7. Click **Save** to set the rules for the sharing policy.
8. In **Sharing Policy**, click **Save** to create the policy.

Use the Shell

This example creates an Internet calendar publishing sharing policy named Internet and configures the policy to share only availability information. The policy is enabled.

```
New-SharingPolicy -Name "Internet" -Domains 'Anonymous:
CalendarSharingFreeBusySimple' -Enabled $true
```

This example adds the sharing policy Internet to a user mailbox.

```
Set-Mailbox -Identity <user name> -SharingPolicy "Internet"
```

This example adds the sharing policy Internet to an organizational unit (OU).

```
Set-Mailbox -OrganizationalUnit <OU name> -SharingPolicy
"Internet"
```

For detailed syntax and parameter information, see New-SharingPolicy and Set-Mailbox.

How do you know this step worked?

To verify that you have successfully created the sharing policy, run the following Shell command to verify the sharing policy information.

```
Get-SharingPolicy <policy name> | format-list
```

Step 3, option 2: Use the EAC or the Shell to configure the

default sharing policy for Internet calendar publishing

If you want to configure the default sharing policy for Internet calendar publishing, complete the following steps.

Use the EAC

1. Navigate to **Organization** > **Sharing**.
2. In the list view, under **Individual Sharing**, select the Default Sharing Policy, and then click **Edit** ✎.
3. In **Sharing Policy**, click **Add +** to add a sharing rule to the policy.
4. In **Sharing Rule**, click **Sharing with a specific domain**, and then and then type **Anonymous** in the corresponding box.
5. To specify the calendar sharing levels you want to enforce for the sharing policy, select the **Share your calendar folder** check box, and then select one of the following:
   - **Calendar free/busy information with time only**
   - **Calendar free/busy information with time, subject, and location**
   - **All calendar appointment information, including time, subject, location and title**
6. Click **Save** to set the rules for the sharing policy.
7. In **Sharing Policy**, click **Save** to save the changes.

Use the Shell

This example updates the Default Sharing Policy and configures the policy to share only availability information. The policy is enabled.

```
Set-SharingPolicy -Name "Default Sharing Policy" -Domains
'Anonymous: CalendarSharingFreeBusySimple' -Enabled $true
```

For detailed syntax and parameter information, see Set-Mailbox.

How do you know this step worked?

To verify that you have successfully updated the Default Sharing Policy, run the following Shell command to verify the sharing policy information.

```
Get-SharingPolicy <policy name> | format-list
```

Disable Internet calendar publishing

Exchange Server 2013 > Sharing > Sharing policies >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-15*

How you disable Internet calendar publishing depends on how you enabled it. If you created a sharing policy dedicated to Internet calendar publishing, you can either disable the policy or delete it altogether. If you configured Internet calendar publishing as a sharing rule in the default sharing policy, you can just remove the sharing rule for the **Anonymous** domain.

When you disable Internet calendar publishing, users who are provisioned to use the sharing policy won't be able to share calendar information with the **Anonymous** Internet domain specified in the policy. However, you can't delete or disable a sharing policy that's dedicated to Internet calendar publishing until all users who are provisioned to use that policy have the policy setting removed from their mailboxes. For details about changing the sharing policy setting for a user, see Manage user mailboxes.

Note:

If you disable or delete a sharing policy, users provisioned to use the policy will continue to share information until the Sharing Policy Assistant runs. To specify how often the Sharing Policy Assistant runs, use the Set-MailboxServer cmdlet with the *SharingPolicySchedule* parameter.

To fully disable Internet calendar publishing, you should also disable the Outlook Web App virtual directory used for calendar publishing. Doing this prohibits access to the published calendar links previously shared by your Exchange organization users with external Internet users. This step is detailed later in this topic.

To learn more about Internet calendar publishing and sharing policies, see Sharing.

What do you need to know before you begin?

- Estimated time to complete this task: 15 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Calendar and Sharing Permissions" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Use the EAC or the Shell to disable or delete the sharing policy for Internet calendar publishing

Use the EAC

1. Navigate to **Organization** > **Sharing**.
2. In the list view, under **Individual Sharing**, perform one of the following steps:
   - If you created a sharing policy specifically for Internet calendar publishing, select that policy, and then either clear the check box in the **On** column to disable the sharing policy or click **Delete** 🗑 to delete it.
   - If you configured Internet calendar publishing as a sharing rule in the default sharing policy, perform the following steps:
     a. Select the default sharing policy, and then click **Edit** ✎.
     b. In **sharing policy**, select the **Anonymous** sharing rule, and then click **Remove** — to remove the sharing rule.
     c. Click **Save**.

Use the Shell

This example disables a dedicated Internet calendar publishing sharing policy named **Internet**.

```
Set-SharingPolicy -Identity "Internet" -Enabled $false
```

This example deletes a dedicated Internet calendar publishing sharing policy named **Internet**.

```
Remove-SharingPolicy -Identity "Internet"
```

For detailed syntax and parameter information, see Set-SharingPolicy.

How do you know this step worked?

To verify that you have successfully removed or updated the sharing policy, run the following Shell command and verify the sharing policy information.

```
Get-SharingPolicy <policy name> | format-list
```

If you've removed the dedicated Internet calendar publishing sharing policy, you won't see the policy in the cmdlet results.

If you've updated the default sharing policy, verify that the `Anonymous` domain has been removed from the *Domains* parameter.

For detailed syntax and parameter information, see Get-SharingPolicy.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Step 2: Use the Shell to disable the Outlook Web App virtual directory Anonymous features

Note:
You can't use the EAC to disable Anonymous features for the Outlook Web App virtual directory.

This example disables Anonymous features for the Outlook Web App virtual directory on Client Access server CAS01.

```
Set-OwaVirtualDirectory -Identity "CAS01" -
AnonymousFeaturesEnabled -$false
```

For detailed syntax and parameter information, see Set-OwaVirtualDirectory.

How do you know this step worked?

To verify that you have successfully disabled the Anonymous features for the Outlook Web App virtual directory on the Client Access server, run the following Shell command and verify that the *AnonymousFeaturesEnabled* parameter is `$false`.

```
Get-OwaVirtualDirectory | format-list
```

For detailed syntax and parameter information, see Get-OwaVirtualDirectory.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Clients and mobile

Exchange Server 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-17*

There are many different clients that can be used to access information in a Microsoft Exchange Server 2013 mailbox. These clients include desktop programs such as Microsoft Outlook, Outlook Web App, and mobile clients such as mobile phones, tablets, and other mobile devices. Each of these clients offers a variety of features.

Looking for a list of all clients and mobile topics? See Clients and mobile documentation.

Clients and mobile documentation

The following table contains links to topics that will help you learn about and manage some of the clients and client access methods that can be used to access an Exchange 2013 mailbox.

Topic	Description
Outlook Anywhere	Learn about Outlook Anywhere, the client access method that provides connectivity to Microsoft Outlook 2007, Outlook 2010, and Outlook 2013. (This feature was formerly known as RPC/HTTP.)
Exchange ActiveSync	Learn about Exchange ActiveSync, the protocol that provides connectivity to a wide variety of mobile phones and tablets. Using Exchange ActiveSync, users can access email, calendar, contact, and task information.
POP3 and IMAP4	Learn about how users can access their Exchange 2013 email by using email programs that use POP3 or IMAP4.
Office Web Apps Server integration	Learn about how the integration of Microsoft Office Web Apps Server helps provide rich attachment preview functionality in Outlook Web App.
Client protocol management	Learn about management of the client protocols of Exchange ActiveSync, Outlook Web App, POP3, IMAP4, the Autodiscover service, Exchange Web Services, and the Availability service.
Outlook Web App	Learn about Outlook Web App, which provides users access to their Exchange 2013 mailbox through a web browser.
MailTips	Learn about MailTips, the informative messages displayed to users while they're composing a message.

Outlook Anywhere

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-26*

In Microsoft Exchange Server 2013, the Outlook Anywhere feature, formerly known as RPC over HTTP, lets clients who use Microsoft Outlook 2013, Outlook 2010, or Outlook 2007 connect to their Exchange servers from outside the corporate network or over the Internet using the RPC over HTTP Windows networking component. This topic describes the Outlook Anywhere feature and lists the benefits of using Outlook Anywhere.

Contents

Outlook Anywhere and Exchange 2013

The Windows RPC over HTTP Proxy component, which Outlook Anywhere clients use to connect, wraps remote procedure calls (RPCs) with an HTTP layer. This allows traffic to traverse network firewalls without requiring RPC ports to be opened. In Exchange 2013, this feature is enabled by default, because Exchange 2013 doesn't allow direct RPC connectivity.

Benefits of using Outlook Anywhere

Outlook Anywhere offers the following benefits to clients that use Outlook 2013, Outlook 2010, or Outlook 2007 to access your Exchange messaging infrastructure:

- Users have remote access to Exchange servers from the Internet.
- You can use the same URL and namespace that you use for Outlook Web App and Microsoft Exchange ActiveSync.
- You can use the same Secure Sockets Layer (SSL) server certificate that you use for both Outlook Web App and Exchange ActiveSync.
- Unauthenticated requests from Outlook can't access Exchange servers.
- You don't have to use a virtual private network (VPN) to access Exchange servers across the

Internet.

- If you already use Outlook Web App with SSL or Exchange ActiveSync with SSL, you don't have to open any additional ports from the Internet.
- You can test end-to-end client connectivity for Outlook Anywhere and TCP-based connections by using the **Test-OutlookConnectivity** cmdlet.

Deploying Outlook Anywhere

In Exchange 2013, Outlook Anywhere is enabled by default, because all Outlook connectivity takes place via Outlook Anywhere. The only post-deployment task you must perform to successfully use Outlook Anywhere is to install a valid SSL certificate on your Client Access server. Mailbox servers in your organization only require the default self-signed SSL certificate.

Managing Outlook Anywhere

You can manage Outlook Anywhere by using the Exchange admin center or the Exchange Management Shell.

Outlook Anywhere coexistence

If you are planning to install Exchange 2013 in a coexistence scenario with previous versions of Exchange Server, you might still have Outlook 2003 clients in your organization. Outlook 2003 is not a supported client for Exchange 2013.

Before you move your namespace to Exchange 2013, you need to ensure that all Outlook clients have been upgraded to the minimum supported version. Outlook 2007 or higher is required for an Outlook Anywhere connection to Exchange 2013, even if the target mailbox is still on Exchange 2007 or Exchange 2010.

In a coexistence scenario that still has 2007 or 2010 Client Access Servers, you need to enable Outlook Anywhere on each legacy Client Access Server. For instructions on enabling Outlook Anywhere for Client Access Servers running on Exchange Server 2007, see How to Enable Outlook Anywhere. For instructions on enabling Outlook Anywhere for Client Access Servers running on Exchange Server 2010, see Enable Outlook Anywhere.

Make sure that when you enable Outlook Anywhere on the Client Access Server, choose NTLM for IIS authentication.

Finally, configure the Outlook Anywhere external host name to point to the Exchange 2013 Outlook Anywhere host name. For instructions for Exchange Server 2007, see How to Configure an External Host Name for Outlook Anywhere. For instructions for Exchange Server 2010, see Configure an External Host Name for Outlook Anywhere.

Testing Outlook Anywhere connectivity

You can test for end-to-end client Outlook connectivity by doing either of the following:

- Running the **Test-OutlookConnectivity** cmdlet. The cmdlet tests for Outlook Anywhere (RPC over HTTP) connections. If the cmdlet test fails, the output notes the step that failed. For detailed syntax and parameters, see Test-OutlookConnectivity.
- Running the Outlook Anywhere connectivity test using the Exchange Remote Connectivity Analyzer (ExRCA). When you run this test, you get a detailed summary showing where the test failed and what steps you can take to fix issues. For more information, see Exchange Remote Connectivity Analyzer.

Both tests try to sign in through Outlook Anywhere after obtaining server settings from the Autodiscover service. End-to-end verification includes the following:

- Testing for Autodiscover connectivity
- Validating DNS
- Validating certificates (whether the certificate name matches the website, whether the certificate has expired, and whether it's trusted)
- Checking that the firewall is set up correctly (ExRCA checks overall firewall setup. The cmdlet tests for Windows firewall configuration.)
- Confirming client connectivity by signing in to the user's mailbox

Test Outlook Anywhere connectivity

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-27*

You can test for end-to-end client Outlook Anywhere connectivity by using either the Shell or the Exchange Remote Connectivity Analyzer (ExRCA). This includes testing for connectivity through the Autodiscover service, creating a user profile, and signing in to the user’s mailbox. All the required values are retrieved from the Autodiscover service.

For additional management tasks related to Outlook Anywhere, see Outlook Anywhere.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Outlook Anywhere" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see

Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to test Outlook Anywhere connectivity

To use the Shell to test Outlook Anywhere connectivity, use the **Test-OutlookConnectivity** cmdlet.

Run the following command.

```
Test-OutlookConnectivity -ProbeIdentity
'OutlookMailboxDeepTestProbe' -MailboxId tony@contoso.com -
Hostname contoso.com
```

Note:

The *OutlookMailboxDeepTestProbe* parameter value tests connectivity from the Mailbox server. To test connectivity from the Client Access server, use *OutlookMailboxCTPProbe* for the *ProbeIdentity* parameter value.

Use the Exchange Remote Connectivity Analyzer to test Outlook Anywhere connectivity

The Exchange Remote Connectivity Analyzer (ExRCA) is a web-based tool designed to test connectivity with a variety of Exchange protocols. You can access the ExRCA here.

1. On the ExRCA website, under **Microsoft Office Outlook Connectivity Tests**, select **Outlook Anywhere**, and then select Next at the bottom of the page.
2. Enter the required information on the next screen, including email address, domain and user name, and password.
3. Choose whether to use Autodiscover to detect server settings or to manually specify server settings.
4. Accept the disclaimer, enter the verification code, and then select **Verify**.
5. Select **Perform Test**.

How do you know this worked?

When the ExRCA tests complete, the output will be displayed on the web page. Any failures will be listed.

MAPI over HTTP

Topic Last Modified: *2014-06-18*

Messaging Application Programming Interface (MAPI) over HTTP is a new transport protocol implemented in Microsoft Exchange Server 2013 Service Pack 1 (SP1) and Microsoft Outlook 2013 SP1. MAPI over HTTP improves the reliability and stability of the Outlook and Exchange connections by moving the transport layer to the industry-standard HTTP model. This allows a higher level of visibility of transport errors and enhanced recoverability. Additional functionality includes support for an explicit pause-and-resume function. This enables supported clients to change networks or resume from hibernation while maintaining the same server context.

Implementing MAPI over HTTP does not mean that it is the only protocol that can be used for Outlook to access Exchange. Outlook clients that are not MAPI over HTTP capable can still use Outlook Anywhere (RPC over HTTP) to access Exchange through a MAPI-enabled Client Access server.

Benefits of MAPI over HTTP

MAPI over HTTP offers the following benefits to clients that use Outlook 2013 SP1:

- Enables future innovation in authentication by using an HTTP based protocol.
- Provides faster reconnection times after a communications break because only TCP connections—not RPC connections—need to be rebuilt. Examples of a communication break include:
  - Device hibernation
  - Changing from a wired network to a wireless or cellular network
- Offers a session context that is not dependent on the connection. The server maintains the session context for a configurable period of time—even if the user changes networks.

Deploy MAPI over HTTP

Consider the following requirements to enable MAPI over HTTP.

- **Supportability** Verify that your intended configuration versions are supported.
- **Prerequisites** Verify that your environment has been upgraded and prepared for MAPI over HTTP.
- **Configuration** Configure the virtual directories, and enable MAPI for your organization.

Supportability

Use the following matrix to verify that your clients and servers support MAPI over HTTP.

Product	Exchange 2013	Exchange 2013	Exchange 2010	Exchange 2007

	SP1	RTM	SP3	SP3
Outlook 2013 SP1	• MAPI over HTTP • Outlook Anywhere	Outlook Anywhere	• RPC • Outlook Anywhere	• RPC • Outlook Anywhere
Outlook 2013 RTM	Outlook Anywhere	Outlook Anywhere	• RPC • Outlook Anywhere	• RPC • Outlook Anywhere
Outlook 2010	Outlook Anywhere	Outlook Anywhere	• RPC • Outlook Anywhere	• RPC • Outlook Anywhere
Outlook 2007	Outlook Anywhere	Outlook Anywhere	• RPC • Outlook Anywhere	• RPC • Outlook Anywhere

Prerequisites

Complete the following steps to prepare the clients and servers to support MAPI over HTTP.

1. Upgrade Outlook clients to Outlook 2013 SP1.
2. Upgrade Client Access and Mailbox servers to Exchange 2013 SP1. For information about how to upgrade, see Upgrade Exchange 2013 to the latest cumulative update or service pack.

Note:
All Client Access servers must be upgraded to Exchange 2013 SP1 before enabling MAPI over HTTP. Otherwise, Outlook can fail to connect to mailboxes. Failure to upgrade the all the Mailbox servers in a Database Availability Group (DAG) can result in email delays and a client requirement to restart Outlook in case of a database failover.

3. For any of your Exchange 2013 SP1 servers that aren't running on Windows Server 2012 R2, you need to upgrade the Microsoft .NET Framework to 4.5.1. For more information see Installing the .NET Framework 4.5, 4.5.1.
4. Install one of the following hotfix rollups for the .NET Framework 4.5.1 on all Exchange 2013 SP1 servers.
   - **Windows Server 2012 R2** KB 2908387
   - **Windows Server 2012** KB 2908385
   - **Windows Server 2008 R2 Service Pack 1** KB 2908383
5. On all Exchange 2013 SP1 Client Access servers, add the following registry key.

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework]
"DisableRetStructPinning"=dword:00000001
```

6. On all Exchange 2013 SP1 Client Access servers, add the **COMPLUS_DisableRetStructPinning**

Windows environment variable by performing the following steps.

- In a Command prompt window, run `systempropertiesadvanced` and click **Environment Variables**.
- In the **System variables** section, click **New** and enter the following information.
  - **Variable name** `COMPLUS_DisableRetStructPinning`
  - **Variable value** 1
- When you are finished, click **OK**.

Configuration

Complete the following steps to configure MAPI over HTTP for your organization.

1. **Virtual directory configuration** By default, Exchange 2013 SP1 creates a virtual directory for MAPI over HTTP. You use the **Set-MapiVirtualDirectory** cmdlet to configure the virtual directory. You must configure an internal URL, an external URL, or both. For more information see, Set-MapiVirtualDirectory.

For example, to configure the default MAPI virtual directory on the local Exchange server by setting the internal URL value to https://contoso.com/mapi, and the authentication method to `Negotiate`, run the following command:

```
Set-MapiVirtualDirectory -Identity "Contoso\mapi (Default Web Site)" -InternalUrl https://Contoso.com/mapi -IISAuthenticationMethods Negotiate
```

2. **Certificate configuration** The digital certificate used by your Exchange environment must include the same *InternalURL* and *ExternalURL* values that are defined on the MAPI virtual directory. For more information on Exchange 2013 certificate management, see Digital certificates and SSL. Make sure the Exchange certificate is trusted on the Outlook client workstation and that there are no certificate errors, especially when you access the URLs configured on the MAPI virtual directory.
3. **Update server rules** Verify that your load balancers, reverse proxies, and firewalls are configured to allow access to the MAPI over HTTP virtual directory.
4. **Enable MAPI over HTTP in your Exchange Organization**

Note:

If you enable MAPI over HTTP in your organization, Outlook 2013 SP1 clients that connect through Exchange 2013 SP1 Client Access servers might not be able to access public folders in the same forest on Exchange 2010 or Exchange 2007 servers. Don't enable MAPI over HTTP in your organization until the public folders have been migrated to Exchange 2013 servers. For more information, see the Release notes for Exchange 2013.
After running the command below, clients using Outlook 2013 SP1 with MAPI over HTTP enabled will see a message to restart Outlook to use MAPI over HTTP.

Run the following command:

```
Set-OrganizationConfig -MapiHttpEnabled $true
```

Test MAPI over HTTP connections

You can test the end-to-end MAPI over HTTP connection by using the **Test-OutlookConnectivity** cmdlet. To use the **Test-OutlookConnectivity** cmdlet, the Microsoft Exchange Health Manager (MSExchangeHM) service must be started.

The following example tests the MAPI over HTTP connection from the Exchange server named ContosoMail.

```
Test-OutlookConnectivity -RunFromServerId ContosoMail -
ProbeIdentity OutlookMapiHttpSelfTestProbe
```

A successful test returns output that's similar to the following example:

```
MonitorIdentity
StartTime                EndTime                  Result
Error     Exception
---------------
---------                -------                  ------
-----     ---------
OutlookMapiHttp.Protocol\OutlookMapiHttpSelfTestProbe
2/14/2014 7:15:00 AM   2/14/2014 7:15:10 AM   Succeeded
```

For more information, see Test-OutlookConnectivity.

Logs for MAPI over HTTP activity are at the following locations:

- %ExchangeInstallPath%Logging\MAPI Address Book Service\
- %ExchangeInstallPath%Logging\MAPI Client Access\
- %ExchangeInstallPath%Logging\HttpProxy\Mapi\

Manage MAPI over HTTP

You can manage the configuration of MAPI over HTTP by using the following cmdlets:

- Set-MapiVirtualDirectory
- Get-MapiVirtualDirectory
- New-MapiVirtualDirectory
- Remove-MapiVirtualDirectory

Exchange ActiveSync

Exchange Server 2013 > Clients and mobile >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-11-19*

Exchange ActiveSync is a client protocol that lets you synchronize a mobile device with your Exchange mailbox. Exchange ActiveSync is enabled by default when you install Microsoft Exchange 2013.

Contents

Overview of Exchange ActiveSync

Exchange ActiveSync is a Microsoft Exchange synchronization protocol that's optimized to work together with high-latency and low-bandwidth networks. The protocol, based on HTTP and XML, lets mobile phones access an organization's information on a server that's running Microsoft Exchange. Exchange ActiveSync enables mobile phone users to access their email, calendar, contacts, and tasks, and to continue to access this information while they're working offline.

Important:
Windows Phone 7 mobile phones support only a subset of all Exchange ActiveSync mailbox policy settings. For a complete list, see Windows Phone 7 Synchronization.

Features in Exchange ActiveSync

Exchange ActiveSync provides the following:

- Support for HTML messages
- Support for follow-up flags
- Conversation grouping of email messages
- Ability to synchronize or not synchronize an entire conversation
- Synchronization of Short Message Service (SMS) messages with a user's Exchange mailbox
- Support for viewing message reply status
- Support for fast message retrieval
- Meeting attendee information
- Enhanced Exchange Search
- PIN reset
- Enhanced device security through password policies
- Autodiscover for over-the-air provisioning
- Support for setting automatic replies when users are away, on vacation, or out of the office
- Support for task synchronization

- Direct Push
- Support for availability information for contacts

Managing Exchange ActiveSync

By default, Exchange ActiveSync is enabled. All users who have an Exchange mailbox can synchronize their mobile device with the Microsoft Exchange server.

You can perform the following Exchange ActiveSync tasks:

- Enable and disable Exchange ActiveSync for users
- Set policies such as minimum password length, device locking, and maximum failed password attempts
- Initiate a remote wipe to clear all data from a lost or stolen mobile phone
- Run a variety of reports for viewing or exporting into a variety of formats
- Control which types of mobile devices can synchronize with your organization through device access rules

Security in Exchange ActiveSync

You can configure Exchange ActiveSync to use Secure Sockets Layer (SSL) encryption for communications between the Exchange server and the mobile device.

Managing mobile device access in Exchange ActiveSync

You can control which mobile devices can synchronize. You do this by monitoring new mobile devices as they connect to your organization or by setting up rules that determine which types of mobile devices are allowed to connect. Regardless of the method you choose to specify which mobile devices can synchronize, you can approve or deny access for any specific mobile device for a specific user at any time

Device security features in Exchange ActiveSync

In addition to the ability to configure security options for communications between the Exchange server and your mobile devices, Exchange ActiveSync offers the following features to enhance the security of mobile devices:

- **Remote wipe** If a mobile device is lost, stolen, or otherwise compromised, you can issue a remote wipe command from the Exchange Server computer or from any Web browser by using Outlook Web App. This command erases all data from the mobile device.
- **Device password policies** Exchange ActiveSync lets you configure several options for device passwords.

⚠**Warning:**
The iOS7 fingerprint reader technology cannot be used as a device password. If you choose to

use the iOS7 fingerprint reader, you'll still need to create and enter a device password if the mobile device mailbox policy for your organization requires a device password.

The device password options include the following:

- **Minimum password length (characters)** This option specifies the length of the password for the mobile device. The default length is 4 characters, but as many as 18 can be included.
- **Minimum number of character sets** Use this text box to specify the complexity of the alphanumeric password and force users to use a number of different sets of characters from among the following: lowercase letters, uppercase letters, symbols, and numbers.
- **Require alphanumeric password** This option determines password strength. You can enforce the usage of a character or symbol in the password in addition to numbers.
- **Inactivity time (seconds)** This option determines how long the mobile device must be inactive before the user is prompted for a password to unlock the mobile device.
- **Enforce password history** Select this check box to force the mobile phone to prevent the user from reusing their previous passwords. The number that you set determines the number of past passwords that the user won't be allowed to reuse.
- **Enable password recovery** Select this check box to enable password recovery for the mobile device. Users can use Outlook Web App to look up their recovery password and unlock their mobile device. Administrators can use the Exchange Administration Center to look up a user's recovery password.
- **Wipe device after failed (attempts)** This option lets you specify whether you want the phone's memory to be wiped after multiple failed password attempts.

- **Device encryption policies** There are a number of mobile device encryption policies that you can enforce for a group of users. These policies include the following:
  - **Require encryption on device** Select this check box to require encryption on the mobile device. This increases security by encrypting all information on the mobile device.
  - **Require encryption on storage cards** Select this check box to require encryption on the mobile device's removable storage card. This increases security by encrypting all information on the storage cards for the mobile device.

Windows Phone 7 synchronization

If you have Windows Phone 7 mobile devices in your organization, these devices will experience synchronization problems if certain Mobile Device mailbox policy properties are configured. To allow Windows Phone 7 mobile phones to synchronize with an Exchange mailbox, either set the **AllowNonProvisionableDevices** property to true or configure only the following Mobile Device mailbox policy properties:

- PasswordRequired
- MinPasswordLength
- IdleTimeoutFrequencyValue
- DeviceWipeThreshold
- AllowSimplePassword
- PasswordExpiration

- PasswordHistory
- DisableRemovableStorage
- DisableIrDA
- DisableDesktopSync
- BlockRemoteDesktop
- BlockInternetSharing

Direct Push

Exchange Server 2013 > Clients and mobile > Exchange ActiveSync >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-07-11*

Direct Push is a feature that's built into Microsoft Exchange Server 2013. Direct Push keeps a mobile device current over a cellular or wireless network connection. It notifies the mobile device when new content is ready to be synchronized.

Contents

Overview

For Direct Push to work, the mobile device must be Direct Push capable. These devices include the following:

- All versions of Windows Phone
- Mobile phones that are produced by Microsoft Exchange ActiveSync licensees and are designed specifically to be Direct Push compatible

By default, Direct Push is enabled in Exchange 2013. Mobile devices that support Direct Push issue a long-lived HTTPS request to the server running Microsoft Exchange. The Exchange server monitors activity on the user's mailbox and sends a response to the mobile device if there are any changes, such as new or changed email, calendar, contact, or task items. If changes occur within the lifespan of the HTTPS request, the Exchange server issues a response to the device that states that changes have occurred and the device should initiate synchronization with the Exchange server. The device then issues this request to the server. When synchronization is complete, a new long-lived HTTPS request is generated to start the process again. This guarantees that email, calendar, contact, and task items are delivered quickly to the mobile device, and that it is always synchronized with the Exchange server.

Direct Push topology

Direct Push operates in the following way:

1. A mobile device that's configured to synchronize with an Exchange 2013 server issues an HTTPS request to the server. This request is known as a PING. The request tells the server to notify the device if any items change in the next 15 minutes in any folder that's configured to synchronize. Otherwise, the server should return an HTTP 200 OK message. The mobile device then stands by. The 15-minute time span is known as a *heartbeat interval*.
2. If no items change in 15 minutes, the server returns a response of HTTP 200 OK. The mobile device receives this response, resumes activity (known as *waking up*), and issues its request again. This restarts the process.
3. If any items change or new items are received within the 15-minute heartbeat interval, the server sends a response that informs the mobile device that there's a new or changed item and provides the name of the folder in which the new or changed item resides. After the mobile device receives this response, it issues a synchronization request for the folder that has the new or changed item. When synchronization is complete, the mobile device issues a new PING request and the whole process starts over.

Direct Push depends on network conditions that support a long-standing HTTPS request. If the carrier network for the mobile device or the firewall doesn't support long-standing HTTPS requests, the HTTPS request is stopped. The following steps describe how Direct Push operates when the carrier network for a mobile device has a time-out value of 13 minutes.

1. A mobile device issues an HTTPS request to the server. The request tells the server to notify the device if any items change in the next 15 minutes in any folder that’s configured to synchronize. Otherwise, the server should return an HTTP 200 OK message. The mobile device then stands by.
2. If the server doesn’t respond after 15 minutes, the mobile device wakes up and concludes that the connection to the server was timed out by the network. The device reissues the HTTPS request, but this time it uses a heartbeat interval of 8 minutes.
3. After 8 minutes, the server sends an HTTP 200 OK message. The device then tries to gain a longer connection by issuing a new HTTPS request to the server that has a heartbeat interval of 12 minutes.
4. After 4 minutes, a new email message is received and the server responds by sending an HTTPS request that tells the device to synchronize. The device synchronizes and reissues the HTTPS request that has a heartbeat of 12 minutes.
5. After 12 minutes, if there are no new or changed items, the server responds by sending an HTTP 200 OK message. The device wakes up and concludes that network conditions support a heartbeat interval of 12 minutes. The device then tries to gain a longer connection by reissuing an HTTPS request that has a heartbeat interval of 16 minutes.
6. After 16 minutes, no response is received from the server. The device wakes up and concludes that network conditions cannot support a heartbeat interval of 16 minutes. Because this failure occurred directly after the device tried to increase the heartbeat interval, it concludes that the heartbeat interval has reached its maximum limit. The device then issues an HTTPS request that

has a heartbeat interval of 12 minutes because this was the last successful heartbeat interval.

The mobile device tries to use the longest heartbeat interval the network supports. This extends battery life on the device and reduces how much data is transferred over the network. Mobile carriers can specify a maximum, minimum, and initial heartbeat value in the registry settings for the mobile device.

Configuring Direct Push to work through your firewall

For Direct Push to work through your firewall, you must open TCP port 443. This port is required for Secure Sockets Layer (SSL) and must be opened between the Internet and the Client Access server.

In addition to opening ports on your firewall, for optimal Direct Push performance, you should increase the time-out value on your firewall from the default of 15 minutes to 30 minutes. The maximum length of the HTTPS request is determined by the following settings:

- The maximum time-out value that's set on the firewalls that control the traffic from the Internet to the Client Access server
- The firewall time-out values that are set by the mobile service provider

Mobile device mailbox policies

Exchange Server 2013 > Clients and mobile > Exchange ActiveSync >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

In Microsoft Exchange Server 2013, you can create mobile device mailbox policies to apply a common set of policies or security settings to a collection of users. After you deploy Exchange ActiveSync in your Exchange 2013 organization, you can create new mobile device mailbox policies or modify existing policies. When you install Exchange 2013, a default mobile device mailbox policy is created. All users are automatically assigned this default mobile device mailbox policy.

Important:

Windows Phone 7 mobile phones only support a subset of all Exchange ActiveSync mailbox policy settings. For a complete list, see Windows Phone 7 Synchronization.

Warning:

The iOS7 fingerprint reader is not supported as a device password. If you enable the fingerprint reader to secure your iOS7 device, you will still need to create and enter a password if your mobile device mailbox policies require a password.

Overview of mobile device mailbox policies

You can use mobile device mailbox policies to manage many different settings. These include the following:

- Require a password
- Specify the minimum password length
- Require a number or special character in the password
- Designate how long a device can be inactive before requiring the user to re-enter a password
- Wipe a device after a specific number of failed password attempts

For more information about all the settings you can configure, see Mobile device policy settings.

Managing Exchange ActiveSync mailbox policies

Mobile device mailbox policies can be created in the Exchange Administration Center (EAC) or the Exchange Management Shell. If you create a policy in the EAC, you can configure only a subset of the available settings. You can configure the rest of the settings using the Shell.

Windows Phone 7 synchronization

If you have Windows Phone 7 mobile phones in your organization, these phones will experience synchronization problems if certain Exchange ActiveSync mailbox policy properties are configured. To allow Windows Phone 7 mobile phones to synchronize with an Exchange mailbox, either set the **AllowNonProvisionableDevices** property to True or only configure the following Exchange ActiveSync mailbox policy properties:

- PasswordRequired
- MinPasswordLength
- IdleTimeoutFrequencyValue
- DeviceWipeThreshold
- AllowSimplePassword
- PasswordExpiration
- PasswordHistory
- DisableRemovableStorage
- DisableIrDA
- DisableDesktopSync
- BlockRemoteDesktop
- BlockInternetSharing

Mobile device mailbox policy settings

The following table summarizes the settings you can specify using mobile device mailbox policies.

Mobile device mailbox policy settings

Setting	Description

Allow Bluetooth	This setting specifies whether a mobile device allows Bluetooth connections. The available options are Disable, HandsFree Only, and Allow. The default value is Allow. The Exchange Enterprise Client Access License is required to change the values of this setting.
Allow Browser	This setting specifies whether Pocket Internet Explorer is allowed on the mobile device. This setting doesn't affect third-party browsers installed on the mobile device. The default value is `$true`. The Exchange Enterprise Client Access License is required to change the values of this setting.
Allow Camera	This setting specifies whether the mobile device camera can be used. The default value is `$true`. The Exchange Enterprise Client Access License is required to change the values of this setting.
Allow Consumer EMail	This setting specifies whether the mobile device user can configure a personal email account (either POP3 or IMAP4) on the mobile device. The default value is `$true`. This setting doesn’t control access to email accounts that are using third-party mobile device email programs. The Exchange Enterprise Client Access License is required to change the values of this setting.
Allow Desktop Sync	This setting specifies whether the mobile device can synchronize with a computer through a cable, Bluetooth, or IrDA connection. The default value is `$true`. The Exchange Enterprise Client Access License is required to change the values of this setting.

Allow External Device Management	This setting specifies whether an external device management program is allowed to manage the mobile device.
Allow HTML Email	This setting specifies whether email synchronized to the mobile device can be in HTML format. If this setting is set to `$false`, all email is converted to plain text.
Allow Internet Sharing	This setting specifies whether the mobile device can be used as a modem for a desktop or a portable computer. The default value is `$true`. The Exchange Enterprise Client Access License is required to change the values of this setting.
AllowIrDA	This setting specifies whether infrared connections are allowed to and from the mobile device. The Exchange Enterprise Client Access License is required to change the values of this setting.
Allow Mobile OTA Update	This setting specifies whether the mobile device mailbox policy settings can be sent to the mobile device over a cellular data connection. The default value is `$true`.
Allow non-provisionable devices	This setting specifies whether mobile devices that may not support application of all policy settings are allowed to connect to Exchange 2013 by using Exchange ActiveSync. Allowing non-provisionable mobile devices has security implications. For example, some non-provisionable devices may not be able to implement an organization's password requirements.
Allow POPIMAPEmail	This setting specifies whether the user can

	configure a POP3 or an IMAP4 email account on the mobile device. The default value is `$true`. This setting doesn't control access by third-party email programs.
Allow Remote Desktop	This setting specifies whether the mobile device can initiate a remote desktop connection. The default value is `$true`. The Exchange Enterprise Client Access License is required to change the values of this setting.
Allow simple password	This setting enables or disables the ability to use a simple password such as 1111 or 1234. The default value is `$true`.
Allow S/MIME encryption algorithm negotiation	This setting specifies whether the messaging application on the mobile device can negotiate the encryption algorithm if a recipient's certificate doesn't support the specified encryption algorithm.
Allow S/MIME software certificates	This setting specifies whether S/MIME software certificates are allowed on the mobile device.
Allow storage card	This setting specifies whether the mobile device can access information that's stored on a storage card. The Exchange Enterprise Client Access License is required to change the values of this setting.
Allow text messaging	This setting specifies whether text messaging is allowed from the mobile device. The default value is `$true`. The Exchange Enterprise Client Access License is required to change the values of this setting.

Allow unsigned applications	This setting specifies whether unsigned applications can be installed on the mobile device. The default value is `$true`. The Exchange Enterprise Client Access License is required to change the values of this setting.
Allow unsigned installation packages	This setting specifies whether an unsigned installation package can be run on the mobile device. The default value is `$true`. The Exchange Enterprise Client Access License is required to change the values of this setting.
Allow Wi-Fi	This setting specifies whether wireless Internet access is allowed on the mobile device. The default value is `$true`. The Exchange Enterprise Client Access License is required to change the values of this setting.
Alphanumeric password required	This setting requires that a password contains numeric and non-numeric characters. The default value is `$true`.
Approved Application List	This setting stores a list of approved applications that can be run on the mobile device. The Exchange Enterprise Client Access License is required to change the values of this setting.
Attachments enabled	This setting enables attachments to be downloaded to the mobile device. The default value is `$true`.
Device encryption enabled	This setting enables encryption on the mobile device. Not all mobile devices can enforce encryption. For more information, see the device and mobile operating system documentation.

Device policy refresh interval	This setting specifies how often the mobile device mailbox policy is sent from the server to the mobile device.
IRM enabled	This setting specifies whether Information Rights Management (IRM) is enabled on the mobile device.
Max attachment size	This setting controls the maximum size of attachments that can be downloaded to the mobile device. The default value is Unlimited.
Max calendar age filter	This setting specifies the maximum range of calendar days that can be synchronized to the mobile device. The following values are accepted: 1. All 2. OneDay 3. ThreeDays 4. OneWeek 5. TwoWeeks 6. OneMonth
Max email age filter	This setting specifies the maximum number of days of email items to synchronize to the mobile device. The following values are accepted: 1. All 2. OneDay 3. ThreeDays 4. OneWeek 5. TwoWeeks 6. OneMonth
Max email body truncation size	This setting specifies the maximum size at which email messages are truncated when synchronized to the mobile device. The value is in kilobytes (KB).

Max email HTML body truncation size	This setting specifies the maximum size at which HTML email messages are truncated when synchronized to the mobile device. The value is in kilobytes (KB).
Max inactivity time lock	This value specifies the length of time that the mobile device can be inactive before a password is required to reactivate it. You can enter any interval between 30 seconds and 1 hour. The default value is 15 minutes.
Max password failed attempts	This setting specifies the number of attempts a user can make to enter the correct password for the mobile device. You can enter any number from 4 through 16. The default value is 8.
Min password complex characters	This setting specifies the minimum number of complex characters required in the mobile device's password. A complex character is a character that is not a letter.
Min password length	This setting specifies the minimum number of characters in the mobile device password. You can enter any number from 1 through 16. The default value is 4.
Password enabled	This setting enables the mobile device password.
Password expiration	This setting enables the administrator to configure a length of time after which a mobile device password must be changed.
Password history	This setting specifies the number of past passwords that can be stored in a user's mailbox. A user can't reuse a stored password.
Password recovery enabled	When this setting is enabled, the mobile device

	generates a recovery password that's sent to the server. If the user forgets their mobile device password, the recovery password can be used to unlock the mobile device and enable the user to create a new mobile device password.
Require device encryption	This setting specifies whether device encryption is required. If set to `$true`, the mobile device must be able to support and implement encryption to synchronize with the server.
Require encrypted S/MIME messages	This setting specifies whether S/MIME messages must be encrypted. The default value is `$false`.
Require encryption S/MIME algorithm	This setting specifies what required algorithm must be used when encrypting S/MIME messages.
Require manual synchronization while roaming	This setting specifies whether the mobile device must synchronize manually while roaming. Allowing automatic synchronization while roaming will frequently lead to larger-than-expected data costs for the mobile device data plan.
Require signed S/MIME algorithm	This setting specifies what required algorithm must be used when signing a message.
Require signed S/MIME messages	This setting specifies whether the mobile device must send signed S/MIME messages.
Require storage card encryption	This setting specifies whether the storage card must be encrypted. Not all mobile device operating systems support storage card encryption. For more information, see your mobile device and mobile operating system

	documentation.
Unapproved InROM application list	This setting specifies a list of applications that cannot be run in ROM. The Exchange Enterprise Client Access License is required to change the values of this setting.

Add or remove users from a mobile mailbox policy

Clients and mobile > Exchange ActiveSync > Mobile device mailbox policies >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

A mobile device mailbox policy allows you to apply a common set of security and mobile device settings to a group of users. You can create multiple mobile device mailbox policies.

Caution:

When you install Microsoft Exchange Server 2013, a default mobile device mailbox policy is created and all users are automatically assigned this policy.

For additional management tasks related to mobile device mailbox policies, see Mobile device mailbox policies.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mobile Device mailbox policy" entry in the Clients and mobile devices permissions topic.
- You must have one mobile device mailbox policy available in the EAC in **Mobile** > **Mobile Device Mailbox Policies**.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Change a user's mobile device mailbox policy

You can use the EAC or the Shell to change a user's mobile device mailbox policy.

Use the EAC to change a user's mobile device mailbox policy

You change a single user's mobile device mailbox policy using the EAC.

1. In the EAC, click **Recipients** > **Mailboxes** and then select a mailbox.
2. In the Details pane, scroll to **Phone and Voice Features** and select **View details** to display the **Mobile Device Details** screen.
3. The mobile device mailbox policy that's currently assigned is displayed. To change the mobile device mailbox policy, click **Browse**.
4. Choose the appropriate mobile device mailbox policy from the list, click **OK** and then click **Save**.

Use the Shell to add a user to a mobile device mailbox policy

You can change a single user's mobile device mailbox policy using the **Get-CASMailbox** cmdlet in the Shell.

1. In the Shell, run the following command.

```
Get-CASMailbox -Identity tony@contoso.com -
ActiveSyncMailboxPolicy "Sales"
```

How do you know this worked?

To verify that you've successfully changed a user's mobile device mailbox policy, do one of the following:

1. In the EAC, click **Recipients** > **Mailboxes**, and then choose a specific recipient. In the Details pane, scroll down to **Phone and Voice Features** and click **View details**.
2. In the Shell, run the following command.

```
Get-CASMailbox -Identity tony@contoso.com
```

Change the mobile device mailbox policy for multiple users at the same time

If you want to change the mobile device mailbox policy for multiple users at the same time, you can use the bulk edit functionality in the EAC or use the Shell to change the mobile device mailbox policy for a filtered set of users.

Use the bulk edit tool in the EAC to change the mobile device mailbox policy for multiple

users

You can update the mobile device mailbox policy for multiple users at once using the Bulk Edit functionality.

1. In the EAC, click **Recipients** > **Mailboxes**.
2. Select multiple users.
3. In the Details pane, scroll down to **Exchange ActiveSync** and click **Update a policy**.
4. Click **Browse** to choose a mobile device mailbox policy.
5. Click **OK** and then click **Save**.

Use the Shell to change the mobile device mailbox policy for a filtered set of users

You can use the Shell to change the mobile device mailbox policy for a filtered set of users. You can filter users on a variety of attributes.

1. In the Shell, run the following command.

```
Get-Mailbox | where { $_.CustomAttribute1 -match "Manager"
 } | Set-CASMailbox -activesyncmailboxpolicy(Get-
ActiveSyncMailboxPolicy "Contoso").Identity
```

Note:

You can substitute `CustomAttribute1` for any of the properties on the **Get-Mailbox** object. To view the full list, type: `Get-Mailbox username |fl`.

How do you know this worked?

To verify that you've successfully changed a user's mobile device mailbox policy, do one of the following:

1. In the EAC, click **Recipients** > **Mailboxes**, and choose a specific recipient. In the Details pane, scroll down to **Phone and Voice Features** and click **View details**.
2. In the Shell, run the following command.

```
Get-CASMailbox -Identity tony@contoso.com
```

Create or modify a mobile device mailbox policy

Clients and mobile > Exchange ActiveSync > Mobile device mailbox policies >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

A mobile device mailbox policy allows you to apply a common set of security and mobile device settings to a group of users. You can create multiple mobile device mailbox policies. Each recipient in your organization must have a mobile device mailbox policy assigned to them. When you install Microsoft Exchange Server 2013, a default mobile device mailbox policy is created and new users are automatically assigned this policy. To assign specific users to a mobile device mailbox policy, see Add or remove users from a mobile mailbox policy.

For additional information related to mobile device mailbox policies, see Mobile device mailbox policies.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mobile Device mailbox policy" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Create a new mobile device mailbox policy

You can use the EAC or the Shell to create a new mobile device mailbox policy.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mobile Device mailbox policy" entry in the Clients and mobile devices permissions topic.

Use the EAC to create a new mobile device mailbox policy

You can create a new mobile device mailbox policy using the EAC.

Note:

You can only set a subset of mobile device mailbox policy settings in the EAC. To set all the mobile device mailbox policy settings, you need to use the Shell.

1. In the EAC, click **Mobile** > **Mobile Device Mailbox Policies**, and then click **New**.
2. Use the various check boxes and drop-down lists to configure the settings for the mobile device mailbox policy.

Warning:

Select **This is the default policy** to make the new mobile mailbox policy the default mobile

mailbox policy. After you make a mobile mailbox policy the default policy, all new users will be assigned this policy automatically when they are created.

3. Click **Save**.

Use the Shell to create a new mobile device mailbox policy

You create a new mobile device mailbox policy using the **New-MobileDeviceMailboxPolicy** cmdlet.

⚠Warning:

There are two cmdlets that can be used to create a new mobile device mailbox policy. The **New-ActiveSyncMailboxPolicy** cmdlet and the **New-MobileDeviceMailboxPolicy** cmdlets perform identical tasks. In a future version of Microsoft Exchange Server, the **New-ActiveSyncMailboxPolicy** cmdlet will be removed. We recommend that you update your scripts and procedures to use the **New-MobileDeviceMailboxPolicy** cmdlet.

1. In the Shell, run the following command.

```
New-MobileDeviceMailboxPolicy -Name:"Management" -
AllowBluetooth:$true -AllowBrowser:$true -AllowCamera:$true
-AllowPOPIMAPEmail:$false -PasswordEnabled:$true -
AlphanumericPasswordRequired:$true -
PasswordRecoveryEnabled:$true -MaxEmailAgeFilter:10 -
AllowWiFi:$true -AllowStorageCard:$true -
AllowPOPIMAPEmail:$false
```

How do you know this worked?

To verify that you've successfully created a mobile device mailbox policy, use one of the following options:

1. In the EAC, click **Mobile** > **Mobile Device mailbox policies**, and verify that your new policy is displayed in the List view.
2. In the Shell, run the following command.

```
Get-MobileDeviceMailboxPolicy -Identity <PolicyName>
```

Edit an existing mobile device mailbox policy

If you want to edit a mobile device mailbox policy, you can use the EAC or the Shell.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mobile Device mailbox policy" entry in the Clients and mobile devices permissions topic.

Use the EAC to edit a mobile device mailbox policy

You can edit a mobile device mailbox policy using the EAC.

Note:
You can only edit a subset of mobile device mailbox policy settings in the EAC. To edit all the mobile device mailbox policy settings, you need to use the Shell.

1. In the EAC, click **Mobile** > **Mobile Device Mailbox Policies**.
2. Select a policy from the List view and click the **Edit** button.
3. Use the **General** and **Security** tabs to edit the mobile device mailbox policy settings.
4. Click **Save** to update the policy.

Use the Shell to edit mobile device mailbox policy settings

You can use the Shell to edit a mobile device mailbox policy.

Warning:
There are two cmdlets that can be used to edit a mobile device mailbox policy. The **Set-ActiveSyncMailboxPolicy** cmdlet and the **Set-MobileDeviceMailboxPolicy** cmdlets perform identical tasks. In a future version of Microsoft Exchange Server, the **Set-ActiveSyncMailboxPolicy** cmdlet will be removed. We recommend that you update your scripts and procedures to use the **Set-MobileDeviceMailboxPolicy** cmdlet.

1. In the Shell, run the following command.

```
Set-MobileDeviceMailboxPolicy -Identity:Default -
DevicePasswordEnabled:$true -
AlphanumericDevicePasswordRequired:$true -
PasswordRecoveryEnabled:$true -MaxEmailAgeFilter:ThreeDays
-AllowWiFi:$false -AllowStorageCard:$true -
AllowPOPIMAPEmail:$false -IsDefault:$true -
AllowTextMessaging:$true -Confirm:$true
```

How do you know this worked?

To verify that you've successfully edited a mobile device mailbox policy, do one of the following:

1. In the EAC, click **Mobile** > **Mobile Device Mailbox Policy**, and then choose a specific policy. In the Details pane, you'll see a number of the policy settings listed.
2. In the Shell, run the following command.

```
Get-MobileDeviceMailboxPolicy -Identity <PolicyName>
```

Supported mobile device mailbox policies for Windows Phones and

devices

Topic Last Modified: *2013-02-19*

With the release of Windows Phone 8, Windows 8, and Windows RT, there are a number of devices that support Exchange ActiveSync and mobile device mailbox policies. Each device operating system supports a specific set of mobile device mailbox policy settings.

Windows client and Exchange Server compatibility matrix

The following tables list the compatible mobile device mailbox policy parameters for the various versions of Windows Phone, Windows 8, Windows RT, and Exchange server.

Windows Phone 7 supported policy parameters

The following table lists the mobile device mailbox policy settings for Windows Phone 7.

Exchange 2007	Exchange 2010	Exchange 2013
DevicePasswordEnabled	DevicePasswordEnabled	DevicePasswordEnabled
AllowSimpleDevicePassword	AllowSimpleDevicePassword	AllowSimpleDevicePassword
AllowIRDA	AllowIRDA	AllowIRDA
MinPasswordLength	MinDevicePasswordLength	MinDevicePasswordLength
PasswordExpiration	DevicePasswordExpiration	DevicePasswordExpiration
AllowDesktopSync	AllowDesktopSync	AllowDesktopSync
MaxInactivityTimeDeviceLock	MaxInactivityTimeDeviceLock	MaxInactivityTimeDeviceLock
DevicePasswordHistory	DevicePasswordHistory	DevicePasswordHistory
AllowRemoteDesktop	AllowRemoteDesktop	AllowRemoteDesktop
AllowStorageCard	AllowStorageCard	AllowStorageCard
AllowInternetSharing	AllowInternetSharing	AllowInternetSharing
AllowNonProvisionableDevices	AllowNonProvisionableDevices	AllowNonProvisionableDevices

Windows Phone 8 supported policy parameters

The following table lists the mobile device mailbox policy settings for Windows Phone 8.

Exchange 2007	Exchange 2010	Exchange 2013
DevicePasswordEnabled	DevicePasswordEnabled	DevicePasswordEnabled
AllowSimpleDevicePassword	AllowSimpleDevicePassword	AllowSimpleDevicePassword
AlphanumericDevicePasswordRequired	AlphanumericDevicePasswordRequired	AlphanumericDevicePasswordRequired
MinDevicePasswordLength	MinDevicePasswordLength	MinDevicePasswordLength
MinDevicePasswordComplexCharacters	MinDevicePasswordComplexCharacters	MinDevicePasswordComplexCharacters
RequireDeviceEncryption	RequireDeviceEncryption	RequireDeviceEncryption
MaxInactivityTimeDeviceLock	MaxInactivityTimeDeviceLock	MaxInactivityTimeDeviceLock
DevicePasswordHistory	DevicePasswordHistory	DevicePasswordHistory
N/A	IRMEnabled	IRMEnabled
DeviceWipeThreshold	MaxDevicePasswordFailedAttempts	MaxDevicePasswordFailedAttempts
AllowNonProvisionableDevices	AllowNonProvisionableDevices	AllowNonProvisionableDevices
DevicePasswordExpiration	DevicePasswordExpiration	DevicePasswordExpiration

Windows 8 and Windows RT supported policy parameters

The following table lists the mobile device mailbox policy settings for Windows 8 and Windows RT.

Exchange 2007	Exchange 2010	Exchange 2013
DevicePasswordEnabled	DevicePasswordEnabled	DevicePasswordEnabled
AllowSimpleDevicePassword	AllowSimpleDevicePassword	AllowSimpleDevicePassword

MinDevicePasswordLength	MinDevicePasswordLength	MinDevicePasswordLength
MinDevicePasswordComplexCharacters	MinDevicePasswordComplexCharacters	MinDevicePasswordComplexCharacters
RequireDeviceEncryption	RequireDeviceEncryption	RequireDeviceEncryption
MaxInactivityTimeDeviceLock	MaxInactivityTimeDeviceLock	MaxInactivityTimeDeviceLock
DevicePasswordHistory	DevicePasswordHistory	DevicePasswordHistory
DeviceWipeThreshold	MaxDevicePasswordFailedAttempts	MaxDevicePasswordFailedAttempts
AllowNonProvisionableDevices	AllowNonProvisionableDevices	AllowNonProvisionableDevices
DevicePasswordExpiration	DevicePasswordExpiration	DevicePasswordExpiration

Mobile device mailbox policy setting interaction

Clients and mobile > Exchange ActiveSync > Mobile device mailbox policies >

Insert introduction here.

Section Heading

Insert section body here.

Subsection Heading

Insert subsection body here.

Mobile devices

Exchange Server 2013 > Clients and mobile > Exchange ActiveSync >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-24*

Mobile devices that are enabled for Microsoft Exchange ActiveSync let users access most of their Microsoft Exchange mailbox data any time, anywhere. There are many different mobile phones and devices enabled for Exchange ActiveSync. These include Windows Phones, Nokia mobile phones, Android phones and tablets, and the Apple iPhone, iPod, and iPad.

Although both phone and non-phone mobile devices support Exchange ActiveSync, in most Exchange ActiveSync documentation, we use the term *mobile device*. Unless the feature or features we're discussing require a cellular telephone signal, such as SMS message notification, the term mobile device applies to both mobile phones and other mobile devices such as tablets.

Exchange ActiveSync

Exchange ActiveSync is a communications protocol that enables mobile access, over the air, to email messages, scheduling data, contacts, and tasks. Exchange ActiveSync is available on Windows Phones and third-party phones that are enabled for Exchange ActiveSync.

Exchange ActiveSync offers Direct Push technology. Direct Push uses an encrypted HTTPS connection that's established and maintained between the mobile device and the server to push new email messages and other Exchange data to the phone.

To use Direct Push with Microsoft Exchange Server 2013, your users must have a mobile device that's designed to support Direct Push.

Exchange ActiveSync features

Exchange ActiveSync provides access to many different features that enable you to enforce security policies on mobile devices. By using Exchange 2013, you can configure multiple mobile device mailbox policies and control which mobile devices can synchronize with your Exchange server. Exchange ActiveSync enables you to send a remote device wipe command that wipes all data from a mobile device in case that mobile device is lost or stolen. Users can also initiate a remote device wipe from Outlook Web App.

Exchange ActiveSync allows users to generate a recovery password. This recovery password is saved on the mobile device and is used when a user forgets their password. The user generates the recovery password at the same time that they generate the device password or PIN. This recovery password can be used to unlock the mobile device. Immediately after this recovery password is used, the user will be required to create a new PIN.

POP3 and IMAP4

If your mobile device doesn’t support Exchange ActiveSync, or you don’t need the rich feature set

that Exchange ActiveSync provides, you can use POP3 or IMAP4 to access your email on your mobile device. For more information about POP3 and IMAP4 access to your mailbox, see POP3 and IMAP4.

Remote device wipe

Clients and mobile > Exchange ActiveSync > Mobile devices >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-19*

Mobile phones, tablets, and other devices can store sensitive corporate data and provide access to many corporate resources. If a mobile device is lost or stolen, that data can be compromised. Through Microsoft Exchange mobile device mailbox policies, you can add a password requirement to your mobile devices. This requires users to enter a password to access their mobile devices. We recommend that, in addition to requiring a device password, you configure your mobile devices to automatically prompt for a password after a period of inactivity. The combination of a device password and inactivity locking provides enhanced security for your corporate data.

In addition to these features, Exchange Server 2013 provides a remote device wipe feature. You can issue a remote device wipe command from the Exchange Management Shell or the Exchange Administration Center (EAC). Users can issue their own remote device wipe commands from the Microsoft Outlook Web App user interface.

The remote device wipe feature also includes a confirmation function that writes a time stamp in the sync state data of the user's mailbox. This time stamp is displayed in Outlook Web App and in the user's mobile phone properties dialog box in the EAC.

Important:

A remote device wipe can reset a mobile phone to the factory default condition. Although the remote device wipe protocol as implemented in Exchange 2013 only requires the deletion of personal corporate data, all current mobile device manufacturers interpret the command as one that wipes all data on the phone. Many mobile device operating systems also wipe all data on any storage card that's inserted in the mobile device. If you're performing a remote device wipe on a mobile phone in your possession and want to keep the data on the storage card, we recommend removing the storage card before you initiate the remote device wipe.

Caution:

After a remote device wipe has occurred, data recovery is very difficult. However, no data removal process leaves a mobile device as free from residual data as when it's new. Recovery of data from a mobile device may still be possible using sophisticated tools.

Remote device wipe vs. local device wipe

Local device wipe occurs when a mobile device wipes itself without the request coming from the server. If your organization has implemented mobile device mailbox policies that specify a maximum number of unsuccessful password attempts and that maximum is exceeded, the mobile device performs a local device wipe. The result of a local device wipe is the same as that of a remote device wipe. The device is returned to its factory default condition. When a mobile device performs a local device wipe, no confirmation is sent to the Exchange server.

View mobile device information for users

Clients and mobile > Exchange ActiveSync > Mobile devices >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-29*

Users can configure multiple mobile devices for synchronization with Microsoft Exchange Server 2013. You can use the EAC or the Shell to view a list of mobile devices that are associated with a specific user.

For additional management tasks related to mobile devices, see Exchange ActiveSync.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mobile Device mailbox policy" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to view mobile device information for users

The EAC displays a list of mobile devices that are currently synchronizing with a user's mailbox. You can view mobile devices by family, model, phone number, or status.

1. In the EAC, click **Recipients** > **Mailboxes** and choose a mailbox.
2. In the Details pane, scroll to **Phone and Voice Features** and click **View details** to display the **Mobile Device Details** screen.

Use the Shell to view mobile device information for users

You can use the **Get-MobileDevice** cmdlet to view a list of mobile devices for a specific user.

1. Run the following command.

```
Get-MobileDevice -Mailbox useralias
```

Mobile phone and tablet features

Clients and mobile > Exchange ActiveSync > Mobile devices >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-07*

Users can access their email, calendar, contacts, and task information on mobile phones, tablets, and other portable devices through Microsoft Exchange ActiveSync. They can also use it to set up their signature and automatic replies. A wide variety of mobile phones and devices work with Exchange ActiveSync.

Note:

Although we consistently refer to devices that access Exchange Server 2013 as mobile phones, there are many devices that can access Exchange 2013 but don't have cellular phone functionality. The term "mobile phone" in this documentation refers to those devices, as well.

Exchange ActiveSync-compatible devices

Users can take advantage of the rich features of Exchange ActiveSync by selecting mobile phones that are compatible with Exchange ActiveSync. These mobile phones are available from many manufacturers and on many carriers. For more information, see the specific mobile phone documentation.

Mobile phones that are compatible with Microsoft Exchange include the following:

- **Apple** The Apple iPhone, iPod Touch, and iPad all support Exchange ActiveSync.
- **Windows Phone** Windows Phone 8, Windows Phone 7, and previous versions all support Exchange ActiveSync.
- **Android** Many mobile phones and tablets with the Android operating system support Exchange ActiveSync. However, these mobile devices may not support all available mobile device mailbox policies. For more information, see Mobile device mailbox policies.

Windows Phone software features

Mobile phones that have a version of Windows Phone software as their operating system offer the greatest functionality when synchronizing with Exchange 2013. The following table lists the mobile device mailbox policies that are available with Windows Phone 8 and Windows Phone 7.

Windows Phone 7 and 8 features

Operating system	Features
Windows Phone 8	Windows Phone 8 supports the following mobile device mailbox policies: • Allow simple device password • Alphanumeric password required • Device password enabled • Device password expiration • IRM enabled • Maximum device password failed attempts • Maximum inactivity device time lock • Minimum device password complex characters • Minimum device password length • Require device encryption • Remote wipe **⚠Warning:** If your organization uses other mobile device mailbox policy settings, you'll need to set the **Allow non-provisionable devices** policy to true. This can have security implications for your organization, because other mobile phones and devices that don't meet all the requirements of your mobile device policy settings will be allowed to synchronize. For more information, see Mobile device mailbox policies.
Windows Phone 7	Windows Phone 7 mobile phones support only a subset of all Exchange ActiveSync mailbox policy settings: • Password required • Minimum password length • Maximum inactivity time lock

	• Maximum device password failed attempts • Allow simple password • Password expiration • Password history • Disable removable storage • Disable IrDA • Disable desktop sync • Block remote desktop • Block Internet sharing

Perform a remote wipe on a mobile phone

Clients and mobile > Exchange ActiveSync > Mobile devices >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-06*

Your users carry sensitive corporate information in their pockets every day. If one of them loses their mobile phone, your data can end up in the hands of another person. If one of your users loses their mobile phone, you can use the Exchange Administration Center (EAC) or the Exchange Management Shell to wipe their phone clean of all corporate and user information.

Note:

This topic also provides instructions for how to use Microsoft Outlook Web App to perform a remote wipe on a phone. The user must be signed in to Outlook Web App to perform a remote wipe.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mobile devices" entry in the Clients and mobile devices permissions topic.
- This procedure will clear all data on the mobile phone, including installed applications, photos, and personal information.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to wipe a user's phone

You can use the EAC to wipe a user's phone or cancel a remote wipe that has not yet completed.

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. Select the user, and under **Mobile Devices**, choose **View details**.
3. On the **Mobile Device Details** page, select the lost mobile device, and then select **Wipe Data**.
4. Select **Save**.

Use the Shell to wipe a user's phone

You can use the **Clear-MobileDevice** cmdlet in the Shell to wipe a user's phone.

The following command wipes the device named WM_TonySmith and sends a confirmation message to admin@contoso.com.

```
Clear-MobileDevice -Identity WM_TonySmith -
NotificationEmailAddresses "admin@contoso.com"
```

Use Outlook Web App to wipe a user's phone

Your users can wipe their own phone using Outlook Web App.

1. In Outlook Web App, select **Settings** > **Phone** > **Mobile devices**.
2. Select the mobile phone.
3. Click or tap the **Wipe Device** icon.

How do you know this worked?

There are several ways to verify that the remote wipe completed.

- Run the **Clear-MobileDevice** cmdlet with the *–NotificationEmailAddresses* parameter configured. A message will be sent to the supplied email address when the remote wipe has completed.
- In the EAC, check the status of the mobile device. The status will change from **Wipe Pending** to **Wipe Successful**.
- In Outlook Web App, check the status of the mobile device. The status will change from **Wipe Pending** to **Wipe Successful**.

Configure mobile phones to access email

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-19*

You can configure a mobile phone, such as a Windows Phone, to use Microsoft Exchange ActiveSync. You should perform this procedure on each mobile phone in your organization.

Prerequisites

- You've reviewed the manufacturer's documentation for the mobile phone you want to configure.
- Exchange ActiveSync is enabled in your organization.

Configure a mobile phone to use Exchange ActiveSync

Most mobile phones and devices are capable of using Autodiscover to configure the mobile email client to use Exchange ActiveSync. To configure an email account on most mobile phones, you'll need two pieces of information.

- The user's email address
- The user's password

If Autodiscover can't contact the Exchange server, you'll need to set up the mobile phone manually. Manual setup requires the user's email address and password as well as the Exchange ActiveSync server name. In most organizations, the Exchange ActiveSync server name is the same as the Outlook Web App server name without the /owa, for example, mail.contoso.com.

Windows Phone synchronization

If you're configuring a Windows Phone mobile phone to synchronize with an Exchange mailbox using Exchange ActiveSync, only a subset of mobile device mailbox policy settings are supported. Those policy settings are detailed in Supported mobile device mailbox policies for Windows Phones and devices.

If you configure mobile device mailbox policy settings that are not supported for the version of Windows Phone you're using, you must also set the **AllowNonProvisionableDevices** policy setting to true or create a separate mobile device mailbox policy for Windows Phone mobile phones.

POP3 and IMAP4

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-21*

By default, POP3 and IMAP4 are disabled in Microsoft Exchange Server 2013. To support POP3 clients that still rely on these protocols, you need to start two POP3 services: the Microsoft Exchange POP3 service and the Microsoft Exchange POP3 Backend service. To support IMAP4 clients that still rely on these protocols, you need to start two IMAP4 services: the Microsoft Exchange IMAP4 service and the Microsoft Exchange IMAP4 Backend service.

For detailed steps for enabling the POP3 and IMAP4 services, see Enable POP3 in Exchange 2013 and Enable IMAP4 in Exchange 2013.

By default, users who have mailboxes on computers that are running Exchange 2013 can access their mailboxes by using Microsoft Outlook or Outlook Web App, Microsoft Exchange ActiveSync, or Outlook Voice Access. Outlook, Outlook Web App, and Outlook Voice Access enable your email users to use the comprehensive set of features that are available to users who have mailboxes on Exchange 2013 servers.

Contents

Overview of POP3 and IMAP4 functionality

This section describes the POP3 and IMAP4 functionality for Exchange 2013.

The POP3 and IMAP4 protocols have the following benefits and limitations:

- **POP3** POP3 was designed to support offline mail processing. With POP3, email messages are removed from the server and stored on the local POP3 client unless the client has been set to leave mail on the server. This puts the data management and security responsibility in the hands of the user. POP3 doesn't offer advanced collaboration features such as calendaring, contacts, and tasks.

- **IMAP4** IMAP4 offers offline and online access but, like POP3, IMAP4 doesn't offer advanced collaboration features such as calendaring, contacts, and tasks.

POP3 and IMAP4 email applications don’t use POP3 and IMAP4 to send messages to the email server. Email applications that use POP3 and IMAP4 rely on the SMTP protocol to send messages. The connector for receiving email submissions from client applications that use POP3 or IMAP4 is created automatically upon installation of Exchange. For more information about connectors, see Receive connectors.

Note:
Each time a user uses a POP or IMAP based email program to open their Office 365 email, they will experience a delay of several seconds. The delay is due using a proxy server, which introduces an additional hop for authentication. The proxy server first looks up the assigned pod server (client access server), and then authenticates against that.

POP3 and IMAP4 cross-site connectivity

In earlier versions of Exchange, you had to perform a manual configuration step to allow your POP3 and IMAP4 clients to connect to their mail from one site in your organization when their mailbox was located in a different site in your organization. By default, Exchange 2013 automatically proxies from a Client Access server in one site to the correct server.

Using non-standard accounts with POP3 and IMAP4

You can't use an Anonymous account or Guest account to sign in to an Exchange 2013 mailbox through POP3 or IMAP4. This kind of access is blocked because of security vulnerabilities when you use non-standard accounts for POP3 and IMAP4 access. Additionally, you can't connect to the Administrator mailbox through POP3 or IMAP4. This limitation was included intentionally in Exchange 2013 to enhance security for the Administrator mailbox. To access the Administrator mailbox, you must use Microsoft Office Outlook or Outlook Web App.

Understanding differences between POP3 and IMAP4

POP3 is a frequently used email Internet protocol. By default, when POP3 email applications download email messages to a client computer, the downloaded messages are removed from the server. When a copy of your user's email isn't kept on the email server, the user can't access the same email messages from multiple computers. However, some POP3 email applications can be configured to keep copies of the messages on the server so that the same email messages can be accessed from another computer. POP3 client applications can only be used to download messages from the email server to a single folder (usually the Inbox) on the client computer. The POP3 protocol can't synchronize multiple folders on the email server with multiple folders on the client computer.

Email client applications that use IMAP4 are more flexible and generally offer more features than

email client applications that use POP3. By default, when IMAP4 email applications download email messages to a client computer, a copy of downloaded messages remains on the email server. Because a copy of the user's email message is kept on the email server, the user can access the same email message from multiple computers. With IMAP4 email, the user can access and create multiple email folders on the email server. Users can then access any of their messages on the server from computers in multiple locations. For example, most IMAP4 applications can be configured to keep a copy of a user's sent items on the server so that they can view their sent items from any other computer. IMAP4 supports additional features that are supported by most IMAP4 applications. For example, some IMAP4 applications include a feature that lets the user view only the headers of their email messages on the server—who the message is from and the subject—and then download only the messages that they want to read.

Note:

IMAP4 and POP3 clients have limited access to calendar information for Exchange. For more information, see Configure calendar options for POP3 and Configure calendar options for IMAP4.

Send and receive options for POP3 and IMAP4 email applications

POP3 and IMAP4 email applications let users choose when they want to connect to the server to send and receive email. This section discusses some of the most common connectivity options and also provides some factors your users should consider when they select connection options available in their POP3 and IMAP4 email applications.

Common configuration settings

Three of the most common connection settings that can be set on the POP3 or IMAP4 client application are:

- To send and receive messages every time the email application is started. When this option is used, mail is only sent and received upon starting the email application.
- To send and receive messages manually. When this option is used, messages are only sent and received when the user clicks a "send and receive" option in the client user interface.
- To send and receive messages every set number of minutes. When this option is used, the client application connects to the server every set number of minutes to send messages and download any new messages.

For information about how to configure these settings for the email application that you use, see the Help documentation that's provided with the respective email application.

Considerations when selecting send/receive options

If the device or computer that's running the POP3 or IMAP4 email application is always connected to the Internet, users may want to configure their email application to send and receive messages every set number of minutes. Connecting to the server at frequent intervals lets the user keep their email application up-to-date with the most current information on the server. However, if the device or computer that's running the POP3 or IMAP4 email application isn't always connected to the Internet (for example, if the user connects to the Internet by using a dial-up connection), the user may want to configure the email application to send and receive messages manually. In a dial-up connectivity scenario, sending and receiving messages manually can potentially reduce the time that a user is connected to the Internet.

Note:

If the user is using an IMAP4-compliant email application that supports the IMAP4 IDLE command, the user may be able to send email to and receive email from their Exchange mailbox in near real time. For this connection method to work, both the email server application and the client application must support the IMAP4 IDLE command. In most cases, users don't have to configure any settings in their IMAP4 application to use this connection method.

POP3 and IMAP4 applications

Because Exchange 2013 supports POP3 and IMAP4, users can use any applications that support POP3 and IMAP4 client applications to connect to Exchange 2013. These applications include Outlook, Windows Live Mail, Microsoft Outlook Express, Entourage, and many third-party applications such as Mozilla Thunderbird and Eudora. The features supported by each email client applications vary. For information about the specific features offered by specific POP3 and IMAP4 client applications, see the documentation that's included with each application.

User settings to configure POP3 or IMAP4 access to their Exchange 2013 mailboxes

After you enable POP3 and IMAP4 client access, you have to give users the information they need to connect their email programs to their Exchange 2013 mailbox. They'll need the following information:

To connect from inside the corporate network, users will need the following information:

- Internal POP3 or IMAP4 server name
- Internal POP3 or IMAP4 port number
- Internal POP3 or IMAP4 encryption method
- Internal SMTP (outgoing server) name
- Internal SMTP (outgoing server) port number
- Internal SMTP (outgoing server) encryption method

To connect from the Internet, they'll need the following information:

- External POP3 or IMAP4 server name
- External POP3 or IMAP4 port number
- External POP3 or IMAP4 encryption method
- External SMTP (outgoing server) name
- External SMTP (outgoing server) port number
- External SMTP (outgoing server) encryption method

You can make these settings available to your users through email or other manual communication methods. You can also configure Exchange so that your users can use Outlook Web App to look up their own settings.

Configuring Exchange so users can look up their POP3, IMAP4, and SMTP server settings

By default, users can't look up their POP3, IMAP4, and SMTP server settings through Outlook Web App. To allow your users to look up their POP3 and IMAP4 server settings through Outlook Web App, you need to use the **Set-PopSettings** and **Set-ImapSettings** cmdlets. To allow your users to look up their SMTP (outgoing) server settings, you must use the **Set-ReceiveConnector** cmdlet.

Do the following to allow users to look up their own POP3, IMAP4, and SMTP settings

- POP3 settings Run the **Set-POPSettings** cmdlet with the *ExternalConnectionSettings* parameter. Use the format `Set-POPSettings -ExternalConnectionSetting {<FQDN>:995:SSL}`. For example, run `Set-PopSettings -ExternalConnectionSetting {Dublin01.Contoso.com:995:SSL}` if you want clients that connect through the server with FQDN Dublin01.Contoso.com to be able to look up their own POP settings.

You must restart IIS after applying this setting.

- IMAP4 settings Run the **Set-IMAPSettings** cmdlet with the *ExternalConnectionSettings* parameter. Use the format `Set-ImapSettings -ExternalConnectionSetting {<FQDN>:993:SSL}`. For example, run `Set-IMAPSettings -ExternalConnectionSetting {Dublin01.Contoso.com:993:SSL}` if you want clients that connect through the server with FQDN Dublin01.Contoso.com to be able to look up their own IMAP setting.

You must restart IIS after applying this setting.

- SMTP settings Run the **Set-ReceiveConnector** cmdlet with the *AdvertiseClientSettings* parameter. Use the format `Set-ReceiveConnector "Client Frontend <Server Name>" -AdvertiseClientSettings $True -FQDN <FQDN>`. For example, run `Set-ReceiveConnector "Client Frontend <Server Name>" -AdvertiseClientSettings $True -FQDN Dublin01.Contoso.com` if you want clients that connect through the server with FQDN Dublin01.Contoso.com to be able to look up their own SMTP setting.

You must restart IIS after applying this setting.

After you change your default settings by running the **Set-POPSettings**, **Set-IMAPSettings**, and **Set-ReceiveConnector** cmdlets, your users can look up their external POP, IMAP, and SMTP server settings in Outlook Web App by clicking **Settings** > **Options** > **Account** > **My account** > **Settings for POP or IMAP access**.

Leaving a copy of messages on the server

The default setting on some email programs isn't to keep a copy of messages on the server after they're retrieved. Be sure to recommend that your users set up their email program to keep a copy of all messages the client retrieves on the server. By keeping a copy of messages on the server, your users can access their messages from a different email program.

Start and stop the POP3 services

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-13*

By default, the two POP3 services, the Microsoft Exchange POP3 service and the Microsoft Exchange POP3 Backend service, aren't started on computers running Microsoft Exchange Server 2013. You must start these two services to allow your email clients to connect to Exchange using POP3. When these services are running, Exchange 2013 accepts unsecured POP3 client communications on port 110 and over port 995 using Secure Sockets Layer (SSL).

The Microsoft Exchange POP3 service runs on Exchange 2013 computers that are running the Client Access server role. The Microsoft Exchange POP3 Backend service runs on the Exchange 2013 computer that's running the Mailbox server role. In environments where the Client Access and Mailbox roles are running on the same computer, you manage both services on the same computer.

For additional information related to POP3 and IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "POP3 and IMAP4 Permissions" section in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Microsoft Management Console Services snap-in

to start or stop the POP3 services

To start the POP3 services:

1. On the computer running the Client Access server role, click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Services**. Right-click **Microsoft Exchange POP3**, and then click **Start**.
2. On the computer running the Mailbox server role, click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Services**. In the result pane, right-click **Microsoft Exchange POP3 Backend**, and then click **Start**.

To stop the POP3 services:

1. On the computer running the Client Access server role, click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Services**. Right-click **Microsoft Exchange POP3**, and then click **Stop**.
2. On the computer running the Mailbox server role, click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Services**. Right-click **Microsoft Exchange POP3 Backend**, and then click **Stop**.

Use the Shell to start or stop the POP3 services

To start the POP3 services:

1. On the computer running the Client Access server role, from the Shell, run the following command to start the Microsoft Exchange POP3 service.

```
Start-service MSExchangePOP3
```

2. On the computer running the Mailbox server role, from the Shell, run the following command to start the Microsoft Exchange POP3 Backend service.

```
Start-service MSExchangePOP3BE
```

To stop the POP3 services:

1. On the computer running the Client Access server role, from the Shell, run the following command to stop the Microsoft Exchange POP3 service.

```
Stop-service MSExchangePOP3
```

2. On the computer running the Mailbox server role, from the Shell, run the following command to stop the Microsoft Exchange POP3 Backend service.

```
Stop-service MSExchangePOP3BE
```

Use net start to start or stop the POP3 services

To start the POP3 services:

1. On the computer running the Client Access server role, at the command prompt, run the following command to start the Microsoft Exchange POP3 service.

```
Net Start msExchangePOP3
```

2. On the computer running the Mailbox server role, at the command prompt, run the following command to start the Microsoft Exchange POP3 Backend service.

```
Net Start msExchangePOP3BE
```

To stop the POP3 services:

1. On the computer running the Client Access server role, at the command prompt, run the following command to stop the Microsoft Exchange POP3 service.

```
Net Stop MSExchangePOP3
```

2. On the computer running the Mailbox server role, at the command prompt, run the following command to stop the Microsoft Exchange POP3 Backend service.

```
Net Stop MSExchangePOP3BE
```

How do you know this worked?

1. On the Exchange Client Access server, open Windows Task Manager. On the **Services** tab, the status for **MSExchangePOP3** will show as **Running** if the Microsoft Exchange POP3 service is running.
2. On the Exchange Mailbox server, open Windows Task Manager. On the **Services** tab, the status for **MSExchangePOP3BE** will show as **Running** if the Microsoft Exchange POP3 Backend service is running.

Start and stop the IMAP4 services

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-05*

By default, the two IMAP4 services, the Microsoft Exchange IMAP4 service and the Microsoft Exchange IMAP4 Backend service, aren't started on computers running Microsoft Exchange Server 2013. You must start these two services to allow your email clients to connect to Exchange using IMAP4. When these services are running, Exchange 2013 accepts unsecured IMAP4 client communications on port 143 and over port 993 using Secure Sockets Layer (SSL).

The Microsoft Exchange IMAP4 service runs on Exchange 2013 computers that are running the Client Access server role. The Microsoft Exchange IMAP4 Backend service runs on the Exchange 2013 computer that's running the Mailbox server role. In environments where the Client Access and Mailbox server roles are running on the same computer, you manage both services on the same computer.

For additional information related to POP3 and IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "POP3 and IMAP4 Permissions" section in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Microsoft Management Console Services snap-in to start or stop the IMAP4 services

To start the IMAP4 services:

1. On the computer running the Client Access server role, click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Services**. Right-click **Microsoft Exchange IMAP4**, and then click **Start**.
2. On the computer running the Mailbox server role, click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Services**. Right-click **Microsoft Exchange IMAP4 Backend**, and then click **Start**.

To stop the IMAP4 services:

1. On the computer running the Client Access server role, click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Services**. Right-click **Microsoft Exchange IMAP4**, and then click **Stop**.
2. On the computer running the Mailbox server role, click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Services**. Right-click **Microsoft Exchange IMAP4 Backend**, and then click **Stop**.

Use the Shell to start or stop the IMAP4 services

To start the IMAP4 services:

1. On the computer running the Client Access server role, from the Shell, run the following command to start the Microsoft Exchange IMAP4 service.

```
Start-service msExchangeIMAP4
```

2. On the computer running the Mailbox server role, from the Shell, run the following command to start the Microsoft Exchange IMAP4 Backend service.

```
Start-service msExchangeIMAP4BE
```

To stop the IMAP4 services:

1. On the computer running the Client Access server role, from the Shell, run the following command to stop the Microsoft Exchange IMAP4 service.

```
Stop-service msExchangeIMAP4
```

2. On the computer running the Mailbox server role, from the Shell, run the following command to stop the Microsoft Exchange IMAP4 Backend service.

```
Stop-service msExchangeIMAP4BE
```

Use net start to start or stop the IMAP4 services

To start the IMAP4 services:

1. On the computer running the Client Access server role, at the command prompt, run the following command to start the Microsoft Exchange IMAP4 service.

```
net start msExchangeIMAP4
```

2. On the computer running the Mailbox server role, at the command prompt, run the following command to start the Microsoft Exchange IMAP4 Backend service.

```
net start msExchangeIMAP4BE
```

To stop the IMAP4 services:

1. On the computer running the Client Access server role, at the command prompt, run the following command to stop the Microsoft Exchange IMAP4 service.

```
Net Stop MSExchangeIMAP4
```

2. On the computer running the Mailbox server role, at the command prompt, run the following command to stop the Microsoft Exchange IMAP4 Backend service.

```
Net Stop MSExchangeIMAP4BE
```

How do you know this worked?

1. On the Exchange Client Access server, open Windows Task Manager. On the **Services** tab, the status for **MSExchangeIMAP4** will show as **Running** if the Microsoft Exchange IMAP4 service is running.
2. On the Exchange Mailbox server, open Windows Task Manager. On the **Services** tab, the status for **MSExchangeIMAP4BE** will show as **Running** if the Microsoft Exchange IMAP4 Backend service is running.

Enable POP3 in Exchange 2013

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-13*

When you install Microsoft Exchange Server 2013, POP3 client connectivity isn't enabled. To enable POP3 client connectivity, you need to start two POP3 services, the Microsoft Exchange POP3 service and the Microsoft Exchange POP3 Backend service. When you enable POP3, Exchange 2013 accepts unsecured POP3 client communications on port 110 and over port 995 using Secure Sockets Layer (SSL).

The Microsoft Exchange POP3 service runs on Exchange 2013 computers that are running the Client Access server role. The Microsoft Exchange POP3 Backend service runs on the Exchange 2013 computer that's running the Mailbox server role. In environments with the Client Access and Mailbox roles on the same computer, you manage both services on the same computer.

For more information related to setting up POP3 and IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "POP3 and IMAP4 Permissions" section in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Microsoft Management Console Services snap-in to enable POP3

On the computer running the Client Access server role:

1. In the **Services** snap-in, in the console tree, click **Services (Local)**.
2. In the result pane, right-click **Microsoft Exchange POP3**, and then click **Properties**.
3. On the **General** tab, under **Startup type**, select **Automatic**, and then click **Apply**.
4. Under **Service status**, click **Start**, and then click **OK**.

On the computer running the Mailbox server role:

1. In the **Services** snap-in, in the console tree, click **Services (Local)**.
2. In the result pane, right-click **Microsoft Exchange POP3 Backend**, and then click **Properties**.
3. On the **General** tab, under **Startup type**, select **Automatic**, and then click **Apply**.
4. Under **Service status**, click **Start**, and then click **OK**.

Use the Shell to enable POP3

On the computer running the Client Access server role:

1. Set the Microsoft Exchange POP3 service to start automatically.

```
Set-service msExchangePOP3 -startuptype automatic
```

2. Start the Microsoft Exchange POP3 service.

```
Start-service msExchangePOP3
```

On the computer running the Mailbox server role:

1. Set the Microsoft Exchange POP3 Backend service to start automatically.

```
Set-service msExchangePOP3BE -startuptype automatic
```

2. Start the Microsoft Exchange POP3 Backend service.

```
Start-service msExchangePOP3BE
```

How do you know this worked?

1. On the Exchange Client Access server, open Windows Task Manager. On the **Services** tab, the status for **MSExchangePOP3** will show as **Running** if POP3 is enabled.
2. On the Exchange Mailbox server, open Windows Task Manager. On the **Services** tab, the status for **MSExchangePOP3BE** will show as **Running** if POP3 is enabled.

Enable IMAP4 in Exchange 2013

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-05*

When you install Microsoft Exchange Server 2013, IMAP4 client connectivity isn't enabled. To enable IMAP4 client connectivity, you need to start two IMAP services, the Microsoft Exchange IMAP4 service and the Microsoft Exchange IMAP4 Backend service. When you enable IMAP4, Exchange 2013 accepts unsecured IMAP4 client communications on port 143 and over port 993 using Secure Sockets Layer (SSL).

The Microsoft Exchange IMAP4 service runs on Exchange 2013 computers that are running the Client Access server role. The Microsoft Exchange IMAP4 Backend service runs on the Exchange 2013 computer that's running the Mailbox server role. In environments where the Client Access and Mailbox roles are running on the same computer, you manage both services on the same computer.

For more information related to setting up POP3 and IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "POP3 and IMAP4 Permissions" section in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Microsoft Management Console Services snap-in to enable IMAP4

On the computer running the Client Access server role:

1. In the **Services** snap-in, in the console tree, click **Services (Local)**.
2. In the result pane, right-click **Microsoft Exchange IMAP4**, and then click **Properties**.
3. On the **General** tab, under **Startup type**, select **Automatic**, and then click **Apply**.

4. Under **Service status**, click **Start**, and then click **OK**.

On the computer running the Mailbox server role:

1. In the **Services** snap-in, in the console tree, click **Services (Local)**.
2. In the result pane, right-click **Microsoft Exchange IMAP4 Backend**, and then click **Properties**.
3. On the **General** tab, under **Startup type**, select **Automatic**, and then click **Apply**.
4. Under **Service status**, click **Start**, and then click **OK**.

Use the Shell to enable IMAP4

On the computer running the Client Access server role:

1. Set the Microsoft Exchange IMAP4 service to start automatically.

```
Set-service msExchangeIMAP4 -startuptype automatic
```

2. Start the Microsoft Exchange IMAP4 service.

```
Start-service msExchangeIMAP4
```

On the computer running the Mailbox server role:

1. Set the Microsoft Exchange IMAP4 Backend service to start automatically.

```
Set-service msExchangeIMAP4BE -startuptype automatic
```

2. Start the Microsoft Exchange IMAP4 Backend service.

```
Start-service msExchangeIMAP4BE
```

How do you know this worked?

1. On the Exchange Client Access server, open Windows Task Manager. On the **Services** tab, the status for **MSExchangeIMAP4** will show as **Running** if IMAP4 is enabled.
2. On the Exchange Mailbox server, open Windows Task Manager. On the **Services** tab, the status for **MSExchangeIMAP4BE** will show as **Running** if IMAP4 is enabled.

Enable or disable POP3 access for a user

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-06*

You can enable or disable POP3 for a user.

Note:

After you've enabled or disabled POP3 for a user, you must restart the Microsoft Exchange POP3 service and the Microsoft Exchange POP3 Backend service. For more information about how to restart the POP3 service, see Start and stop the POP3 services.

For additional information related to managing user mailboxes, see Manage user mailboxes.

For additional information related to POP3 and IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient provisioning permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or disable POP3 for a user

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the result pane, select the user for which you want to enable or disable POP3, and then click **Edit** .
3. In the **User Mailbox** dialog box, in the console tree, click **Mailbox Features**.

In the result pane, under **Email Connectivity**, do one of the following:

- To disable POP3 for the user, under **POP3: Enabled**, click **Disable**.
- To enable POP3 for the user, under **POP3: Disabled**, click **Enable**.

4. Click **Save**.

Use the Shell to enable or disable POP3 for a user

This example enables POP3 for the user John Smith.

```
Set-CASMailbox -Identity "John Smith" -POPEnabled $true
```

This example disables POP3 for the user John Smith.

```
Set-CASMailbox -Identity "John Smith" -POPEnabled $false
```

How do you know this worked?

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the result pane, select the user for which you want to enable or disable POP3, and then click **Edit**.
3. In the **User Mailbox** dialog box, in the console tree, click **Mailbox Features**.

In the result pane, look under **Email Connectivity**.

- If POP3 is enabled for the user, you will see **POP3: Enabled**.
- If POP3 is disabled for the user, you will see **POP3: Disabled**.

4. Click **Save**.

Enable or disable IMAP4 access for a user

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-18*

You can enable or disable IMAP4 for a user.

Note:

After you've enabled or disabled IMAP4 for a user, you must restart the Microsoft Exchange IMAP4 service and the Microsoft Exchange IMAP4 Backend service. For more information about how to restart the IMAP4 service, see Start and stop the IMAP4 services.

For additional information related to managing user mailboxes, see Manage user mailboxes.

For additional information related to POP3 and IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Recipient provisioning permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server,

Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or disable IMAP4 for a user

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the result pane, select the user for which you want to enable or disable IMAP4, and then click **Edit** .
3. In the **User Mailbox** dialog box, in the console tree, click **Mailbox Features**.

In the result pane, under **Email Connectivity**, do one of the following:

- To disable IMAP4 for the user, under **IMAP4: Enabled**, click **Disable**.
- To enable IMAP4 for the user, under **IMAP4: Disabled**, click **Enable**.

4. Click **Save**.

Use the Shell to enable or disable IMAP4 for a user

This example enables IMAP4 for the user John Smith.

```
Set-CASMailbox -Identity "John Smith" -IMAPEnabled $true
```

This example disables IMAP4 for the user John Smith.

```
Set-CASMailbox -Identity "John Smith" -IMAPEnabled $false
```

How do you know this worked?

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the result pane, select the user for which you want to enable or disable IMAP4, and then click **Edit**.
3. In the **User Mailbox** dialog box, in the console tree, click **Mailbox Features**.

In the result pane, look under **Email Connectivity**.

- If IMAP4 is enabled for the user, you will see **IMAP4: Enabled**.
- If IMAP4 is not enabled for the user, you will see **IMAP4: Disabled**.

4. Click **Save**.

Set connection limits for POP3

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-28*

You can use the EAC or the Shell to manage POP3 connection limits for your organization.

When you specify connection limits for POP3, you can select connection limits for the server, an IP address, or a specific user.

For additional information related to POP3, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "POP3 settings" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EMC to set POP3 connection limits for a server, an IP address, or a user

1. In the EAC, navigate to **Servers > Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .
3. On the server properties page, click **POP3**.
4. Scroll down and click **More options**.
5. Under **Connection limits**, use the following settings:
   - **Maximum connections** Specifies the total number of connections the specified server will accept. This includes authenticated and unauthenticated connections. The default value is 2,147,483,647. The possible values are from 1 through 2,147,483,647.
   - **Maximum connections from a single IP address** Specifies the number of connections that the server will accept from a single IP address. The default value is 2,147,483,647. The possible values are from 1 through 2,147,483,647.
   - **Maximum connections from a single user** Specifies the maximum number of connections that the server will accept from a particular user. The default value is 16. The possible values are from 1 through 2,147,483,647.
   - **Maximum command size (bytes)** specifies the maximum size of a single command. The default size is 512. The possible values are from 40 through 1,024.
6. Click **Apply**, and then click **OK** to save your changes.

After you set connection limits, you must restart the POP3 services. For information about how to restart the POP3 services, see Start and stop the POP3 services.

Use the Shell to set POP3 connection limits for a server, an IP address, or a user

This example sets the connection limit for a server.

```
Set-PopSettings -Identity CAS01 -MaxConnections Value
```

This example sets the connection limit for an IP address.

```
Set-PopSettings -Identity CAS01 -MaxConnectionsFromSingleIP Value
```

This example sets the connection limit for a user.

```
Set-PopSettings -MaxConnectionsPerUser Value
```

This example sets the maximum command size.

```
Set-PopSettings -MaxCommandSize Value
```

After you set connection limits, you must restart the POP3 services. For information about how to restart the POP3 services, see Start and stop the POP3 services.

For more information about syntax and parameters, see Set-PopSettings.

How do you know this worked?

To verify that you've successfully set connection limits, do one of the following:

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .
3. On the server properties page, click **POP3**.
4. Scroll down and click **More options**.
5. Under **Connection limits**, verify the connection settings are correct.

Or

1. Run the following command in the Shell.

```
Get-PopSettings | format-list
```

2. Verify the connection settings are correct.

For more information

After you set POP3 connection limits for a server, an IP address, or a user, you may also want to:

Enable POP3 in Exchange 2013

Set connection time-out limits for POP3

Set connection limits for IMAP4

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-28*

You can use the EAC or the Shell to manage IMAP4 connection limits for your organization.

When you specify connection limits for IMAP4, you can select connection limits for the server, an IP address, or a specific user.

For additional information related to IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "IMAP4 settings" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to set IMAP4 connection limits for a server, an IP address, or a user

1. In the EAC, navigate to **Servers > Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .

3. On the server properties page, click **IMAP4**.
4. Scroll down and click **More options**.
5. Under **Connection limits**, use the following settings:
   - **Maximum connections** Specifies the total number of connections the specified server will accept. This includes authenticated and unauthenticated connections. The default value is 2,147,483,647. The possible values are from 1 through 2,147,483,647.
   - **Maximum connections from a single IP address** Specifies the number of connections that the server will accept from a single IP address. The default value is 2,147,483,647. The possible values are from 1 through 2,147,483,647.
   - **Maximum connections from a single user** Specifies the maximum number of connections that the server will accept from a particular user. The default value is 16. The possible values are from 1 through 2,147,483,647.
   - **Maximum command size (bytes)** Specifies the maximum size of a single command. The default size is 10,240. The possible values are from 1,024 through 16,384.
6. Click **Apply**, and then click **OK** to save your changes.

After you set connection limits, you must restart the IMAP4 services. For information about how to restart the IMAP4 services, see Start and stop the IMAP4 services.

Use the Shell to set IMAP4 connection limits for a server, an IP address, or a user

This example sets the connection limit for a server.

```
Set-ImapSettings -Identity CAS01 -MaxConnections Value
```

This example sets the connection limit for an IP address.

```
Set-ImapSettings -Identity CAS01 -
MaxConnectionsFromSingleIP Value
```

This example sets the connection limit for a user.

```
Set-ImapSettings -MaxConnectionsPerUser Value
```

This example sets the maximum command size.

```
Set-ImapSettings -MaxCommandSize Value
```

After you set connection limits, you must restart the IMAP4 services. For information about how to restart the IMAP4 services, see Start and stop the IMAP4 services.

For more information about syntax and parameters, see Set-ImapSettings.

How do you know this worked?

To verify that you've successfully set connection limits, do one of the following:

1. In the EAC, navigate to **Servers > Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .
3. On the server properties page, click **IMAP4**.
4. Scroll down and click **More options**.
5. Under **Connection limits**, verify the connection settings are correct.

Or

1. Run the following command in the Shell.

```
Get-ImapSettings | format-list
```

2. Verify the connection settings are correct.

For more information

After you set IMAP4 connection limits for a server, IP address, or a user, you may also want to:

Enable IMAP4 in Exchange 2013

Set connection time-out limits for IMAP4

Set connection time-out limits for POP3

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-28*

You can use the EAC or the Shell to configure the connection time-out limits for idle authenticated and unauthenticated POP3 connections.

For additional information related to POP3, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "POP3 settings" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to set connection time-out limits for POP3

1. In the EAC, navigate to **Servers > Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .
3. On the server properties page, click **POP3**.
4. Scroll down and click **More options**.
5. Under **Time-out settings**, use the following settings:
   - **Authenticated time-out (seconds)** Specifies the time to wait before closing an idle authenticated connection. The default value is 1,800. The possible values are from 30 through 86,400.
   - **Unauthenticated time-out (seconds)** Specifies the time to wait before closing an idle connection that isn't authenticated. The default value is 60. The possible values are from 30 through 3,600.
6. Click **Apply**, and then click **OK** to save your changes.

After you've set the connection time-out limits for POP3, you must restart the POP3 services for the settings to take effect. For information about how to restart the POP3 services, see Start and stop the POP3 services.

Use the Shell to set connection time-out limits for POP3

This example sets the connection time-out limit for idle authenticated connections.

```
Set -PopSettings -Identity CAS01 -
AuthenticatedConnectionTimeout TimeValue
```

This example sets the connection time-out limit for idle unauthenticated connections.

```
Set -PopSettings -Identity CAS01 -
PreAuthenticatedConnectionTimeout TimeValue
```

After you've set the connection time-out limits for POP3, you must restart the POP3 services for the settings to take effect. For information about how to restart the POP3 services, see Start and stop the POP3 services.

For more information about syntax and parameters, see Set-PopSettings.

How do you know this worked?

To verify that you've successfully set connection limits, do one of the following:

1. In the EAC, navigate to **Servers > Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .
3. On the server properties page, click **POP3**.
4. Scroll down and click **More options**.
5. Under **Time-out settings**, verify the connection settings are correct.

Or

1. Run the following command in the Shell.

```
Get-PopSettings | format-list
```

2. Verify the connection settings are correct.

For more information

After you set connection time-out limits for POP3, you may also want to:

Enable POP3 in Exchange 2013

Set connection limits for POP3

Set connection time-out limits for IMAP4

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-28*

You can use the EAC or the Shell to configure the connection time-out limits for idle authenticated and unauthenticated IMAP4 connections.

For additional information related to IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "IMAP4 settings" entry in the Clients and mobile devices

permissions topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to set connection time-out limits for IMAP4

1. In the EAC, navigate to **Servers > Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .
3. On the server properties page, click **IMAP4**.
4. Scroll down and click **More options**.
5. Under **Time-out settings**, use the following settings:
   - **Authenticated time-out (seconds)** Specifies the time to wait before closing an idle authenticated connection. The default value is 1,800. The possible values are from 30 through 86,400.
   - **Unauthenticated time-out (seconds)** Specifies the time to wait before closing an idle connection that isn't authenticated. The default value is 60. The possible values are from 30 through 3,600.
6. Click **Apply**, and then click **OK** to save your changes.

After you've set the connection time-out limits for IMAP4, you must restart the IMAP4 services for the settings to take effect. For information about how to restart the IMAP4 services, see Start and stop the IMAP4 services.

Use the Shell to set connection time-out limits for IMAP4

This example sets the connection time-out limit for idle authenticated connections.

```
Set -ImapSettings -Identity CAS01 -
AuthenticatedConnectionTimeout TimeValue
```

This example sets the connection time-out limit for idle unauthenticated connections.

```
Set -ImapSettings -Identity CAS01 -
PreAuthenticatedConnectionTimeout TimeValue
```

After you've set the connection time-out limits for IMAP4, you must restart the IMAP4 services for the settings to take effect. For information about how to restart the IMAP4 services, see Start and stop the IMAP4 services.

For more information about syntax and parameters, see Set-ImapSettings.

How do you know this worked?

To verify that you've successfully set connection limits, do one of the following:

1. In the EAC, navigate to **Servers > Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** ✎.
3. On the server properties page, click **IMAP4**.
4. Scroll down and click **More options**.
5. Under **Time-out settings**, verify the connection settings are correct.

Or

1. Run the following command in the Shell.

```
Get-ImapSettings | format-list
```

2. Verify the connection settings are correct.

For more information

After you set authentication time-out limits for IMAP4, you may also want to:

Enable IMAP4 in Exchange 2013

Set connection limits for IMAP4

Configure calendar options for POP3

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-27*

You can use the Shell to configure calendaring access settings for your users who connect to their mailboxes using POP3 connections. The settings you specify determine how your POP3 users can access their calendar and exchange calendar information (for example, send or respond to a meeting request) with other users.

For additional information related to POP3, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "POP3 settings" entry in the Clients and mobile devices permissions topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to set the calendar options for POP3

This example enables POP3 users to use the iCalendar standard, a standard for exchanging calendar information.

```
Set-PopSettings -Identity CAS01 -
CalendarItemRetrievalOption iCalendar
```

This example enables POP3 users to access calendar information from an internal server.

```
Set-PopSettings -Identity CAS01 -
CalendarItemRetrievalOption IntranetUrl
```

This example enables POP3 users to access calendar information from the Internet on an external server.

```
Set-PopSettings -CalendarItemRetrievalOption InternetUrl
```

This example enables POP3 users to access calendar information by using a direct Outlook Web App URL. If you're using `Custom`, you must specify an Outlook Web App URL using the *OWAServerUrl* parameter.

```
Set-PopSettings -CalendarItemRetrievalOption Custom -
OwaServerUrl "https://OwaServer01"
```

After you've specified the calendar options for POP3, you must restart the POP3 services. For information about how to restart the POP3 services, see Start and stop the POP3 services.

For more information about syntax and parameters, see Set-PopSettings.

How do you know this worked?

To verify that you've successfully set calendar options, do the following:

Run the following command in the Shell.

```
Get-PopSettings | format-list
```

Verify that the calendar settings are correct.

For more information

After you set the calendar options for POP3, you may also want to:

Configure POP3 and IMAP4 message retrieval format options

Set connection time-out limits for POP3

Configure calendar options for IMAP4

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-27*

You can use the Shell to configure calendaring access settings for your users who connect to their mailboxes using IMAP4 connections. The settings you specify determine how your IMAP4 users can access their calendar and exchange calendar information (for example, send or respond to a meeting request) with other users.

For additional information related to IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "IMAP4 settings" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to set the calendar options for IMAP4

This example enables IMAP4 users to use the iCalendar standard, a standard for exchanging calendar information.

```
Set-ImapSettings -Identity CAS01 -
CalendarItemRetrievalOption iCalendar
```

This example enables IMAP4 users to access calendar information from an internal server.

```
Set-ImapSettings -Identity CAS01 -
CalendarItemRetrievalOption IntranetUrl
```

This example enables IMAP4 users to access calendar information from the Internet on an external server.

```
Set-ImapSettings -CalendarItemRetrievalOption InternetUrl
```

This example enables IMAP4 users to access calendar information by using a direct Outlook Web App URL. If you're using `Custom`, you must specify an Outlook Web App URL using the *OWAServerUrl* parameter.

```
Set-Imap4Settings -CalendarItemRetrievalOption Custom -
OwaServerUrl "https://OwaServer01"
```

After you've specified the calendar options for IMAP4, you must restart the IMAP4 services. For information about how to restart the IMAP4 services, see Start and stop the IMAP4 services.

For more information about syntax and parameters, see Set-ImapSettings.

How do you know this worked?

To verify that you've successfully set calendar options, do the following:

Run the following command in the Shell.

```
Get-ImapSettings | format-list
```

Verify that the calendar settings are correct.

For more information

After you set the calendar options for IMAP4, you may also want to:

Configure POP3 and IMAP4 message retrieval format options

Set connection time-out limits for IMAP4

Configure POP3 and IMAP4 message retrieval format options

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-28*

You can configure the message retrieval format for users who connect to their email using POP3 and IMAP4. Message retrieval options can be configured at the server level using the EAC or the Shell, and can be configured at the user level using the Shell.

For additional information related to POP3 and IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "POP3 settings" and "IMAP4 settings" entries in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Set the POP3 message retrieval format at the server level

Use the EAC to set the POP3 message retrieval format at the server level

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .
3. On the server properties page, click **POP3**.
4. Under **Message MIME format**, choose from the following settings:
   - Text
   - HTML
   - HTML and alternative text
   - Enriched text
   - Enriched text and alternative text

- Best body format
- TNEF

5. Click **Save**.

After you've set the message retrieval format settings for POP3, you must restart the POP3 services for the settings to take effect. For information about how to restart the POP3 services, see Start and stop the POP3 services.

Use the Shell to set the POP3 message retrieval format at the server level

This example sets the message retrieval format option to text only for all POP3 users on server CAS01.

```
Set-PopSettings -Identity CAS01 -MessageRetrievalMimeFormat TextOnly
```

You can choose from the following settings. You can specify the value for the *MessageRetrievalMimeFormat* parameter by using a numerical value or a text string.

Message format	**Value**
Text	0 or `TextOnly`
HTML	1 or `HtmlOnly`
HTML and alternative text	2 or `HtmlAndTextAlternative`
Enriched text	3 or `TextEnriched`
Enriched text and alternative text	4 or `TextEnrichedAndTextAlternative`
Best body format	5 or `BestBodyFormat`
TNEF	6 or `Tnef`

After you've set the message retrieval format settings for POP3, you must restart the POP3 services for the settings to take effect. For information about how to restart the POP3 services, see Start and stop the POP3 services.

For more information about syntax and parameters, see Set-PopSettings.

How do you know this worked?

Do the following to verify that you've successfully set POP3 message retrieval settings on a server.

1. Run the following command in the Shell.

```
Get-PopSettings | format-list
```

2. Verify the *MessageRetrievalMimeFormat* setting is correct.

Set the IMAP4 message retrieval format at the server level

Use the EAC to set the IMAP4 message retrieval format at the server level

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .
3. On the server properties page, click **IMAP4**.
4. Under **Message MIME format**, choose from the following settings:
   - Text
   - HTML
   - HTML and alternative text
   - Enriched text
   - Enriched text and alternative text
   - Best body format
   - TNEF
5. Click **Save**.

After you've set the message retrieval format settings for IMAP4, you must restart the IMAP4 services for the settings to take effect. For information about how to restart the IMAP4 services, see Start and stop the IMAP4 services.

Use the Shell to set the IMAP4 message retrieval format at the server level

This example sets the message retrieval format option to text only for all IMAP4 users on server CAS01.

```
Set-ImapSettings -Identity CAS01 -
MessageRetrievalMimeFormat TextOnly
```

You can choose from the following settings. You can specify the value for the *MessageRetrievalMimeFormat* parameter by using a numerical value or a text string.

Message format	**Value**
Text	0 or `TextOnly`
HTML	1 or `HtmlOnly`
HTML and alternative text	2 or `HtmlAndTextAlternative`
Enriched text	3 or `TextEnriched`

Enriched text and alternative text	4 or `TextEnrichedAndTextAlternative`
Best body format	5 or `BestBodyFormat`
TNEF	6 or `Tnef`

After you've set the message retrieval format settings for IMAP4, you must restart the IMAP4 services for the settings to take effect. For information about how to restart the IMAP4 services, see Start and stop the IMAP4 services.

For more information about syntax and parameters, see Set-ImapSettings.

How do you know this worked?

Do the following to verify that you've successfully set IMAP4 message retrieval settings on a server.
1. Run the following command in the Shell.

```
Get-ImapSettings | format-list
```

2. Verify the *MessageRetrievalMimeFormat* setting is correct.

Set the POP3 message retrieval format for a user

Use the Shell to set the POP3 message retrieval format for a user

This example sets the message retrieval format to text only for POP3 access for `USER01`.

```
Set-CASMailbox -Identity USER01 -
PopMessagesRetrievalMimeFormat TextOnly
```

You can choose from the following settings. You can specify the value for the *PopMessagesRetrievalMimeFormat* parameter by using a numerical value or a text string.

Message format	**Value**
Text	0 or `TextOnly`
HTML	1 or `HtmlOnly`
HTML and alternative text	2 or `HtmlAndTextAlternative`
Enriched text	3 or `TextEnriched`
Enriched text and alternative text	4 or `TextEnrichedAndTextAlternative`

Best body format	5 or `BestBodyFormat`
TNEF	6 or `Tnef`

After you've set the message retrieval format settings for POP3, you must restart the POP3 services for the settings to take effect. For information about how to restart the POP3 services, see Start and stop the POP3 services.

For more information about syntax and parameters, see Set-CASMailbox.

How do you know this worked?

Do the following to verify that you've successfully set POP3 message retrieval format options for a user.

1. Run the following command in the Shell.

```
Get-CAS Mailbox <identity> | format-list
```

2. Verify the value for *PopMessagesRetrievalMimeFormat* is correct.

Set the IMAP4 message retrieval format for a user

Use the Shell to set the IMAP4 message retrieval format for a user

This example sets the message retrieval format to text only for IMAP4 access for USER01.

```
Set-CASMailbox -Identity USER01 -
ImapMessagesRetrievalMimeFormat TextOnly
```

You can specify the value for the *ImapMessagesRetrievalMimeFormat* parameter by using a numerical value or a text string.

Message format	**Value**
Text	0 or `TextOnly`
HTML	1 or `HtmlOnly`
HTML and alternative text	2 or `HtmlAndTextAlternative`
Enriched text	3 or `TextEnriched`
Enriched text and alternative text	4 or `TextEnrichedAndTextAlternative`
Best body format	5 or `BestBodyFormat`

TNEF	6 or `Tnef`

After you've set the message retrieval format settings for IMAP4, you must restart the IMAP4 services for the settings to take effect. For information about how to restart the IMAP4 services, see Start and stop the IMAP4 services.

For more information about syntax and parameters, see Set-CASMailbox.

How do you know this worked?

Do the following to verify that you've successfully set IMAP4 message retrieval format options for a user.

1. Run the following command in the Shell.

```
Get-CAS Mailbox <identity> | format-list
```

2. Verify the value for *ImapMessagesRetrievalMimeFormat* is correct.

For more information

After you set the message retrieval format for IMAP4 and POP3 users, you may also want to:

Enable or disable POP3 access for a user

Enable or disable IMAP4 access for a user

Configure calendar options for IMAP4

Configure calendar options for POP3

Configure IP addresses and ports for POP3 and IMAP4 access

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-28*

You can use the EAC and the Shell to configure the Microsoft Exchange POP3 and IMAP4 services to use IP addresses and ports that are different from the default settings.

Note:
Enter IP addresses and IP address ranges in the Internet Protocol Version 4 (IPv4) format, Internet Protocol Version 6 (IPv6) format, or both formats. A default installation of Windows Server 2008 enables support for IPv4 and IPv6.

For additional information related to POP3 and IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "POP3 settings" and “IMAP4 settings” entries in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Configure IP addresses and ports for POP3

Use the EAC to configure IP addresses and ports for POP3

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .
3. On the server properties page, click **POP3**.
4. Under **TLS or unencrypted connections**, click **Add +**.
5. On the **Add IP address** page, under **IP address**, choose one of the following:
   - **All available IPv4 addresses** Use all available IPv4 IP addresses for a server.
   - **All available IPv6 addresses** Use all available IPv6 IP addresses for a server.
   - **Specify an IP address** Use a specific IP address.
6. Under **Port**, enter a port number, or accept the default port.
7. Click **Save** to save your changes.

After you've set the IP address and port settings for POP3, you must restart the POP3 service for the settings to take effect. For information about how to restart the POP3 service, see Start and stop the POP3 services.

Use the Shell to configure IP addresses and ports for POP3

This example sets the IP address and port for communicating with Exchange by using POP3 with Secure Sockets Layer (SSL).

```
Set-PopSettings -SSLBindings: IPaddress:Port
```

This example sets the IP address and port for communicating with Exchange by using POP3 with no encryption or Transport Layer Security (TLS) encryption.

```
Set-PopSettings -UnencryptedOrTLSBindings IPaddress:Port
```

After you've set the IP address and port settings for POP3, you must restart the POP3 service for the settings to take effect. For information about how to restart the POP3 service, see Start and stop the POP3 services.

For more information about syntax and parameters, see Set-PopSettings.

How do you know this worked?

Do the following to verify that you have changed POP3 IP address and port settings on a server.
1. Run the following command in the Shell.

```
Get-PopSettings | format-list
```

2. Verify the *UnencryptedOrTLSBindings* and *SSLBindings* settings are correct.

Configure IP addresses and ports for IMAP4

Use the EAC to configure IP addresses and ports for IMAP4

1. In the EAC, navigate to **Servers > Servers**.
2. In the list of servers, select the Client Access server, and then click **Edit** .
3. On the server properties page, click **IMAP4**.
4. If you want to set TLS or unencrypted connection settings, under **TLS or unencrypted connections**, click **Add +**. If you want to change Secure Sockets Layer (SSL) connection settings, under **Secure Sockets Layer (SSL) connections**, click **Add +**.
5. On the **Add IP address** page, under **IP address**, choose one of the following:
   - **All available IPv4 addresses** Use all available IPv4 IP addresses for a server.
   - **All available IPv6 addresses** Use all available IPv6 IP addresses for a server.
   - **Specify an IP address** Use a specific IP address.
6. Under **Port**, enter a port number, or accept the default port.
7. Click **Save** to save your changes.

After you've set the IP address and port settings for IMAP4, you must restart the IMAP4 services for the settings to take effect. For information about how to restart the IMAP4 services, see Start and stop the IMAP4 services.

Use the Shell to configure IP addresses and ports for IMAP4

This example sets the IP address and port for communicating with Exchange by using IMAP4.

```
Set-ImapSettings -SSLBindings: IPaddress:Port
```

This example sets the IP address and port for communicating with Exchange by using IMAP4 with no encryption or TLS encryption.

```
Set-ImapSettings -UnencryptedOrTLSBindings IPaddress:Port
```

After you've set the IP address and port settings for IMAP4, you must restart the IMAP4 service for the settings to take effect. For information about how to restart the IMAP4 service, see Start and stop the IMAP4 services.

For more information about syntax and parameters, see Set-ImapSettings.

How do you know this worked?

Do the following to verify that you have changed IMAP4 IP address and port settings on a server.
1. Run the following command in the Shell.

```
Get-ImapSettings | format-list
```

2. Verify the *UnencryptedOrTLSBindings* and *SSLBindings* settings are correct.

For more information

After you configure IP addresses and ports for POP3 and IMAP4, you may also want to:

Enable IMAP4 in Exchange 2013

Enable POP3 in Exchange 2013

Allow POP3, IMAP4, and SMTP server settings to be viewed by end users in Outlook Web App

Exchange Server 2013 > Clients and mobile > POP3 and IMAP4 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-28*

If you have users who use POP3 or IMAP4 to connect to their Microsoft Exchange Server 2013 mailboxes, they need to know the correct server settings to connect. After a default Exchange 2013 installation, your users can't look up their own incoming POP3 or IMAP4 server settings or outgoing SMTP server settings. However, you can configure Exchange so that your users can look up their own settings using Microsoft Outlook Web App.

After you perform these procedures, your users can look up their server settings in Outlook Web App as follows:
1. In Outlook Web App, click **Settings** > **Options**.

2. In **Options**, click **Account** > **My account** > **Settings for POP or IMAP access**.

For additional information related to POP3 and IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to allow POP3 and IMAP4 users to view their incoming POP3 and IMAP4 settings in Outlook Web App

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "POP3 settings" and "IMAP4 settings" entries in the Clients and mobile devices permissions topic.

This example allows external POP3 server settings to be viewed by end users.

```
Set-PopSettings -ExternalConnectionSettings
{Dublin01.Contoso.com:995:SSL}
```

For detailed syntax and parameter information, see Set-PopSettings.

This example allows external IMAP4 server settings to be viewed by end users.

```
Set-ImapSettings -ExternalConnectionSettings
{Dublin01.Contoso.com:993:SSL}
```

For detailed syntax and parameter information, see Set-ImapSettings.

To apply these changes, you must restart IIS. You don't need to restart the POP3 services. To restart IIS, from a command prompt, enter the following:

```
iisreset
```

How do you know this worked?

To verify that you've configured Exchange to allow users to view their POP3 server settings:
1. Run the following command in the Shell.

```
Get-PopSettings | format-list
```

2. Verify that the *ExternalConnectionSettings* property is set.

To verify that you've configured Exchange to allow users to view their IMAP4 server settings:
1. Run the following command in the Shell.

```
Get-ImapSettings | format-list
```

2. Verify that the *ExternalConnectionSettings* property is set.

Use the Shell to allow POP3 and IMAP4 users to view their outgoing SMTP settings in Outlook Web App

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Receive connectors" entry in the Mail flow permissions topic.

This example allows internal and external SMTP server settings to be viewed by end users using Outlook Web App.

```
Get-ReceiveConnector "*Client Frontend*" | Set-
ReceiveConnector -Fqdn Server.Contoso.com -
AdvertiseClientSettings $true
```

For detailed syntax and parameter information, see Set-ReceiveConnector.

How do you know this worked?

To verify that you've configured Exchange to allow users to view their SMTP server settings:
1. Run the following command in the Shell.

```
Get-ReceiveConnector | format-list
```

2. If the *AdvertiseClientSettings* property is set to `true`, users can view their SMTP server settings in Outlook Web App. If *AdvertiseClientSettings* is set to `false`, users can't view their SMTP sever settings in Outlook Web App.

For more information

After you make it possible for end users to view their POP3, IMAP4, and SMTP settings, you may also want to:

Enable or disable POP3 access for a user

Protocol logging for POP3 and IMAP4

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-27*

You can use protocol logging to review the POP3 and IMAP4 connections in your Exchange environment. This can be useful if you're troubleshooting issues related to POP3 or IMAP4 performance.

Enabling POP3 and IMAP4 protocol logging

You can enable, disable, or change protocol logging using the Exchange Management Shell. If you enable protocol logging using the Shell, the default protocol logging settings will be used. In most cases, the default settings will be sufficient.

Alternatively, you can enable, disable, and modify protocol logging options by editing the Microsoft.Exchange.Pop3.exe.config and Microsoft.Exchange.Imap4.exe.config configuration files located on your Microsoft Exchange Server 2013 Client Access server. For more information about how to manage POP3 and IMAP4 protocol settings, see Configure protocol logging for POP3 and IMAP4.

Reviewing the protocol log

The protocol log files are text files that contain data in the comma-separated value (CSV) file format. The protocol log stores each protocol event on a single line. The information stored on each line is organized by fields. These fields are separated by commas. The following table describes the fields that are used to classify each protocol event.

Fields used to classify each protocol event

Field name	Description
date-time	The date and time of the protocol event. The value is formatted as *yyyy-mm-ddhh:mm:ss.fffZ*, where *yyyy* = year, *mm* = month, *dd* = day, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and *Z* signifies Zulu. Zulu is another

	way to indicate Coordinated Universal Time (UTC).
connector-id	This field isn't used for POP3 and IMAP4 protocol logging.
session-id	A GUID that uniquely identifies the SMTP session that is associated with a protocol event.
sequence-number	A counter that starts at 0 and is incremented for each event in the same session.
local-endpoint	The local endpoint of a POP3 or IMAP4 session. This consists of an IP address and TCP port number, formatted as follows: <*IP address*>:<*port*>.
remote-endpoint	The remote endpoint of a POP3 or IMAP4 session. This consists of an IP address and TCP port number, formatted as follows: <*IP address*>:<*port*>.
event	A single character that represents the protocol event. The possible values for the event are as follows: • + Connect • - Disconnect • > Send • < Receive • * Information
data	Text information that's associated with the POP3 or IMAP4 event.
context	This field isn't used for POP3 and IMAP4 protocol logging.

Configure protocol logging for POP3 and IMAP4

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-27*

You can use the Shell to enable, disable, or modify protocol logging settings for POP3 and IMAP4. By default, protocol logging isn't enabled.

Protocol logging lets you review the POP3 and IMAP4 connections in your Exchange environment. This can be useful if you're troubleshooting issues related to POP3 or IMAP4 performance. For more information, see Protocol logging for POP3 and IMAP4. For more information related to POP3 and IMAP4, see POP3 and IMAP4.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "POP3 settings" and "IMAP4 settings" entries in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to enable protocol logging for POP3 or IMAP4

This example enables protocol logging for IMAP4 or POP3 on the Client Access server CAS01.

```
Set-ImapSettings -Server "CAS01" -ProtocolLogEnabled $true
Set-PopSettings -Server "CAS01" -ProtocolLogEnabled $true
```

Note:

After you've changed protocol logging settings for POP3 or IMAP4, you must restart whichever services you're using: POP3 or IMAP4. For information about how to restart the POP3 and IMAP4 services, see Start and stop the POP3 services and Start and stop the IMAP4 services.

For detailed syntax and parameter information, see Set-ImapSettings and Set-PopSettings.

Use the Shell to disable protocol logging for POP3 or IMAP4

This example disables protocol logging for IMAP4 or POP3 on the Client Access server CAS01.

```
Set-ImapSettings -Server "CAS01" -protocolLogEnabled $false
Set-PopSettings -Server "CAS01" -protocolLogEnabled $false
```

Note:

After you've changed protocol logging settings for POP3 or IMAP4, you must restart whichever services you're using: POP3 or IMAP4. For information about how to restart the POP3 and IMAP4 services, see Start and stop the POP3 services and Start and stop the IMAP4 services.

For detailed syntax and parameter information, see Set-ImapSettings and Set-PopSettings.

Use the Shell to modify protocol logging for POP3 or IMAP4

To modify POP3 or IMAP4 logging settings, run the **Set-ImapSettings** or **Set-PopSettings** cmdlets with one or more of the following parameters.

- *LogFileLocation* This parameter specifies the location for the POP3 or IMAP4 protocol log files. By default, POP3 protocol log files are located in the C:\Program Files\Microsoft\Exchange Server \V15\Logging\Pop3 directory. This example turns on POP3 protocol logging on the Client Access server CAS01. It also changes the POP3 protocol logging directory to C:\Pop3Logging.

```
Set-PopSettings -Server "CAS01" -ProtocolLogEnabled $true -
LogFileLocation "C:\Pop3Logging"
```

- *LogFileRollOverSettings* This parameter defines how frequently POP3 or IMAP4 protocol logging creates a new log file. By default, a new log file is created every day. The possible values are:

Hourly

Daily

Weekly

Monthly

This setting applies only when the value for the parameter *LogPerFileSizeQuota* is set to zero. This example changes the POP3 protocol logging on the Client Access server CAS01 to create a new log

file every hour.

```
Set-PopSettings -Server "CAS01" -LogPerFileSizeQuota 0 -
LogFileRollOverSettings Hourly
```

- *LogPerFileSizeQuota* This parameter defines the maximum size of a POP3 or IMAP4 protocol log file in bytes. By default, this value is set to zero. When this value is set to zero, a new protocol log file is created at the frequency specified by the *LogFileRollOverSettings* parameter.

This example changes the POP3 protocol logging on the Client Access server CAS01 to create a new log file when a log file reaches 2 megabytes (MB).

```
Set-PopSettings -Server "CAS01" -LogPerFileSizeQuota
2000000
```

This example changes the POP3 protocol logging on the Client Access server CAS01 to use the same log file regardless of its creation date and size.

```
Set-PopSettings -Server "CAS01" -LogPerFileSizeQuota
unlimited
```

Note:

After you've changed protocol logging settings for POP3 or IMAP4, you must restart whichever services you're using: POP3 or IMAP4. For information about how to restart the POP3 and IMAP4 services, see Start and stop the POP3 services and Start and stop the IMAP4 services.

For detailed syntax and parameter information, see Set-ImapSettings and Set-PopSettings.

How do you know this worked?

Run the following command in the Shell to verify POP3 protocol logging settings. If POP3 protocol logging is enabled, the value for the *ProtocolLogEnabled* parameter is `True`. If POP3 protocol logging is disabled, the value is `False`. You can also make sure the values for the *LogFileLocation*, *LogPerFileSizeQuota*, and *LogFileRollOverSettings* parameters are correct.

```
Get-PopSettings | format-list
```

Run the following command in the Shell to verify IMAP4 protocol logging settings. If IMAP4 protocol logging is enabled, the value for the *ProtocolLogEnabled* parameter is `True`. If IMAP4 protocol logging is disabled, the value is `False`. You can also make sure the values for the *LogFileLocation*, *LogPerFileSizeQuota*, and *LogFileRollOverSettings* parameters are correct.

```
Get-ImapSettings | format-list
```

For more information

After you configure protocol logging settings for POP3 and IMAP4, you may also want to:

Start and stop the IMAP4 services

Start and stop the POP3 services

Office Web Apps Server integration

Exchange Server 2013 > Clients and mobile >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-26*

Outlook Web App in Microsoft Exchange Server 2013 provides rich attachment preview functionality. All attachments in an email message are displayed in a filmstrip that includes a thumbnail of each attachment. Users are able to preview attachments online in full fidelity. For Office attachments, this means users can use a rich user interface to preview and modify the attachment online. This functionality is made possible by the integration of Microsoft Office Web Apps Server.

By default, the following file types are displayed using Office Web Apps Server:

- Word documents (doc, docx, dotx, dot, dotm extensions)
- Excel documents (xls, xlsx, xlsm, xlm, xlsb extensions)
- PowerPoint documents (ppt, pptx, pps, ppsx, potx, pot, pptm, potm, ppsm extensions)

Note:
Office Web Apps Server won't be used to render attachments in IRM protected messages.

In Exchange 2010, the attachment previews were displayed using the web-ready document viewing technology, which is built in to Exchange. With Office Web Apps Server integration in Exchange Server 2013, when the user wants to preview an Office attachment, Exchange makes a call to the Office Web Apps Server which renders the document instead. This provides a richer preview experience for the user.

Office Web Apps Server integration for attachment previews is available to all Exchange Online customers. Exchange on-premises customers need to deploy an Office Web Apps Server to enable the functionality.

Configure Office Web Apps Server integration

This section provides detailed steps for configuring Office Web Apps Server integration with Exchange Server 2013 for on-premises customers. This procedure doesn't apply to Exchange Online customers because the functionality is already enabled in Exchange Online.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- You have Office Web Apps Server deployed in your organization. For more information, see Deploy the infrastructure: Office Web Apps Server
- Your Office Web Apps Server is accessible from the Internet. If it isn't accessible from the Internet, Office Web Apps Server integration will work only when users access Outlook Web App within your corporate network.
- You can't use the Exchange Administration Center (EAC) to perform these procedures. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to configure the Office Web Apps Server URL

To use Office Web Apps Server to render attachments in Outlook Web App, you must specify the URL of your Office Web Apps Server. Use the **Set-OrganizationConfig** cmdlet to configure the URL.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange server configuration settings" entry in the Exchange and Shell infrastructure permissions topic.

This example sets the Office Web Apps Server URL to **https://Server1/hosting/discovery**.

```
Set-OrganizationConfig -WACDiscoveryEndPoint https://
Server1/hosting/discovery
```

For detailed syntax and parameter information, see Set-OrganizationConfig.

How do you know this worked?

To verify that you have configured the Office Web App Server URL correctly, do the following:

1. Run the following Shell command:

```
Get-OrganizationConfig | Format-List WACDiscoveryEndPoint
```

2. Verify that the correct URL is listed for the *WACDiscoveryEndPoint* attribute.
3. Once you verify that the URL is successfully updated, recycle the **MSExchangeOWAAppPool** on your CAS servers.

4. At this point when users log on to OWA, the CAS server will perform Office Web Apps discovery on the URL you configured. In the CAS server application event log, look for event ID 142 with source MSExchange OWA. This event indicates a successful Office Web Apps discovery, and OWA will use Office Web Apps to render attachments.
5. If you see event ID 141 with source MSExchange OWA, it means the CAS server was unable to locate the Office Web Apps on the given URL, In this case, make sure that the URL you configured is correct.

Use the Shell to enable or disable Office Web Apps Server rendering

You can enable rendering of attachments using Office Web Apps Server for both public and private computers. Use the **Set-OwaVirtualDirectory** cmdlet for both options.

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Outlook Web App virtual directories" entry in the Clients and mobile devices permissions topic.

This example enables Office Web Apps Server rendering on the default Outlook Web App virtual directory on server Server01 for users who logged on to Outlook Web App using the **Private** option:

```
Set-OwaVirtualDirectory "Server01\owa (Default Web Site)" -WacViewingOnPrivateComputersEnabled $true
```

This example disables Office Web Apps Server rendering on the default Outlook Web App virtual directory on server Server01 for users who logged on to Outlook Web App using the **Public** option.

```
Set-OwaVirtualDirectory "Server01\owa (Default Web Site)" -WacViewingOnPublicComputersEnabled $false
```

For detailed syntax and parameter information, see Set-OwaVirtualDirectory.

How do you know this worked?

To verify that you have configured Office Web Apps server rendering correctly, do the following:
1. Run the following Shell command:

```
Get-OwaVirtualDirectory "Server01\owa (Default Web Site)" | Format-List Name,WacViewing*
```

2. Verify that the *WacViewingOnPrivateComputersEnabled* attribute is set to `True` and that *WacViewingOnPublicComputersEnabled* is set to `False`.

Use the Shell to force Office Web Apps Server rendering

You can force users to render attachments using the Office Web Apps Server first before they can open them directly.

You need to be assigned permissions before you can perform this procedure or procedures. To see

what permissions you need, see the "Outlook Web App virtual directories" entry in the Clients and mobile devices permissions topic.

This example configures the Outlook Web App virtual directory on Server01 so that users always first view supported attachments using Office Web Apps Server before they can open them, regardless of the option they chose when logging on to Outlook Web App.

```
Set-OwaVirtualDirectory "Server01\owa (Default Web Site)" -
ForceWacViewingFirstOnPublicComputers $true -
ForceWacViewingFirstOnPrivateComputers $true
```

For detailed syntax and parameter information, see Set-OwaVirtualDirectory.

How do you know this worked?

To verify that you have configured Office Web Apps server rendering correctly, do the following:
1. Run the following Shell command:

```
Get-OwaVirtualDirectory "Server01\owa (Default Web Site)" |
Format-List Name,ForceWacViewing*
```

2. Verify that the *ForceWacViewingFirstOnPrivateComputers* and *ForceWacViewingFirstOnPublicComputers* attributes are both set to `True`.

Client protocol management

Exchange Server 2013 > Clients and mobile >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-05*

Management of the client protocols of Exchange ActiveSync, Outlook Web App, POP3, IMAP4, the Autodiscover service, Exchange Web Services, and the Availability service occurs in three different areas: the Exchange Administration Center (EAC), the Exchange Management Shell, and Internet Information Services (IIS) Manager. The settings that are managed in each location vary per client protocol.

Managing Outlook Web App settings

Most of the settings that affect which Outlook Web App features are available to users can be set on the Outlook Web App virtual directory or can be configured in an Outlook Web App mailbox policy. By using Outlook Web App mailbox policies, you can define the features available to individual users. Mailbox policy settings override virtual directory settings. For more information on

managing Outlook Web App, see Outlook Web App.

Managing Exchange ActiveSync settings

In Exchange 2010, all client access protocols were implemented and managed on a single server role, the Client Access server role. Management of the protocols was performed on a single instance of IIS, there was a single virtual directory object in Active Directory for each client protocol, and a single set of cmdlets were used to configure the virtual directory.

In Exchange 2013, the client protocol management for Exchange ActiveSync is split between the Client Access server and the Mailbox server. Because of this architecture change, you can run different virtual directory management tasks on both the Client Access server and the Mailbox server. If these two servers aren't installed on the same physical computer, the parameters that you use with the virtual directory cmdlets will change based on the server role on which you are running them.

For more information about the architecture changes in Exchange 2013, see What's new in Exchange 2013.

There are two types of settings that can be applied to the Exchange ActiveSync virtual directory:

- Settings applicable to the mailbox session
- Settings applicable to the server and the virtual directory

The settings that are applicable to the mailbox session are user session settings. When a user connects to a Client Access server, the connection is proxied to the Mailbox server that contains the user's mailbox. A unique identifier of the virtual directory is included with the proxied request. The Mailbox server then retrieves the virtual directory settings from Active Directory and applies them to the session. The virtual directory settings are cached on the Mailbox server to improve performance.

If the connection is proxied to a different Active Directory site, the virtual directory settings will be loaded from the Client Access server in the same site as the Mailbox server, not from the Client Access server where the connection originated.

The following tables indicate which virtual directory settings can be managed on which servers. If you try to manage a particular setting on a server for which it isn't applicable, you will receive an error message indicating that the property you are trying to set is read-only for the server that you are operating on.

Exchange ActiveSync virtual directory settings on Client Access servers

Setting	Server
BadItemReportingEnabled	Client Access
BasicAuthEnabled	Client Access

ClientCertAuth	Client Access
CompressionEnabled	Client Access
ExternalAuthenticationMethods	Client Access
ExternalURL	Client Access
InternalAuthenticationMethods	Client Access
InternalURL	Client Access
MobileClientCertificateAuthorityURL	Client Access
MobileClientCertificateProvisioningEnabled	Client Access
MobileClientCertTemplateName	Client Access
RemoteDocumentsActionForUnknownServers	Client Access
RemoteDocumentsAllowedServers	Client Access
RemoteDocumentsBlockedServers	Client Access
RemoteDocumentsInternalDomainSuffixList	Client Access
SendWatsonReport	Client Access

Exchange ActiveSync virtual directory settings on Client Access and Mailbox servers

Setting	Server
ApplicationRoot	Client Access and Mailbox
AppPoolID	Client Access and Mailbox
MetabasePath	Client Access and Mailbox
Name	Client Access and Mailbox
Path	Client Access and Mailbox
ProxySubVdir	Client Access and Mailbox

VirtualDirectoryName	Client Access and Mailbox
WebsiteName	Client Access and Mailbox

Managing POP3 and IMAP4 settings

In Exchange 2013, the implementation of the POP3 and IMAP4 protocols has also been segmented between the Client Access and Mailbox server roles. Due to the new implementation, POP3 and IMAP4 connectivity are each managed by a service on the Client Access server, as well as by a service on the Mailbox server. The names of the services that run on the Client Access server are the same as the names that existed in Exchange 2010: Microsoft Exchange IMAP4 service and Microsoft Exchange POP3 service. The names of the two new services that run on the Mailbox server are the Microsoft Exchange IMAP4 Backend service and the Microsoft Exchange POP3 Backend service.

Consider the following as you manage POP3 and IMAP4 connectivity in your organization:

- If you are running the Client Access server role and the Mailbox server role on the same computer, any changes you make to POP3 or IMAP4 settings are automatically applied to the correct POP3 and IMAP4 services.
- If you are running the Client Access server role and the Mailbox server role on separate computers, you need to manage the settings on the computer that manages the setting you want to change.

Use the following tables indicate which POP/IMAP settings are each server role.

POP3 and IMAP4 settings on Client Access server

Setting	Server
AuthenticatedConnectionTimeout	Client Access
Banner	Client Access
ExternalConnectionSettings	Client Access
InternalConnectionSettings	Client Access
MaxCommandSize	Client Access
MaxConnectionFromSingleIP	Client Access
MaxConnections	Client Access
MaxConnectionsPerUser	Client Access
PreAuthenticatedConnectionTimeout	Client Access

UnencryptedOrTLSBindings	Client Access

POP3 and IMAP4 settings on Mailbox server

Setting	Server
CalendarItemRetrivalOption	Mailbox
EnableExactRFC822Size	Mailbox
MessageRetrievalSortOrder	Mailbox
OWAServerURL	Mailbox
ProxyTargetPort	Mailbox
ShowHiddenFoldersEnabled	Mailbox
SuppressReadReceipt	Mailbox

POP3 and IMAP4 settings on Client Access and Mailbox servers

Setting	Server
X509CertificateName	Client Access and Mailbox
EnforceCertificateErrors	Client Access and Mailbox
LogFileLocation	Client Access and Mailbox
LogFileRolloverSettings	Client Access and Mailbox
LoginType	Client Access and Mailbox
LogPerFileSizeQuota	Client Access and Mailbox
ProotocolLogEnabled	Client Access and Mailbox
Server	Client Access and Mailbox
X509CertificateName	Client Access and Mailbox

Virtual directory management

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-05*

Many of the client protocols used with Exchange Server 2013 are accessed through virtual directories. A virtual directory is used by Internet Information Services (IIS) to allow access to a web application such as Exchange ActiveSync, Outlook Web App, or the Autodiscover service. You can manage a variety of virtual directory settings on Exchange 2013 including authentication, security, and reporting settings.

Understanding virtual directories

The tasks that you can perform on the various virtual directories vary per client protocol. For example, on the Exchange ActiveSync virtual directory you can enable bad item logging and configure the accepted authentication types for the Exchange ActiveSync virtual directory among other tasks. For the Outlook Web App virtual directory, you can configure authentication, segmentation, and file access settings.

Managing virtual directories

Virtual directory management can be performed in three places. You can manage a variety of settings in the Exchange Administration Center (EAC), as well as the Exchange Management Shell. You can also manage certain virtual directory settings in Internet Information Services Manager. For more information about the various settings you can manage on the virtual directories for Exchange ActiveSync, Outlook Web App, and the Autodiscover service, see the following topics:

- View or configure Outlook Web App virtual directories
- Exchange ActiveSync virtual directory management tasks

Exchange ActiveSync virtual directory management tasks

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-05*

You can manage several of the Exchange ActiveSync application settings in Exchange Server 2013 through the Exchange ActiveSync virtual directory. A virtual directory is used by Internet Information Services (IIS) to allow access to a web application such as Exchange ActiveSync. Some of the virtual

directory settings you can manage for Exchange ActiveSync include authentication, security, and reporting.

Exchange ActiveSync virtual directory settings

You can modify the following properties and settings on the Exchange ActiveSync virtual directory:

- **InternalURL** The InternalURL is the URL that internal clients can use to access the virtual directory. It is usually in the format https://servername/Microsoft-Server-ActiveSync. For example, if the server's NetBIOS name is Sequoia, the InternalURL would be https://sequoia/Microsoft-Server-ActiveSync.
- **ExternalURL** The ExternalURL is the URL that external clients can use to access the virtual directory. This URL should be accessible from outside your internal network. For example, your ExternalURL could be https://www.contoso.com/.
- **Authentication settings** The two methods of authentication you can configure for the Exchange ActiveSync virtual directory are Basic authentication and Client certificate authentication.

Default settings for Exchange virtual directories

Clients and mobile > Client protocol management > Virtual directory management >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-19*

Exchange Server 2013 automatically configures multiple Internet Information Services (IIS) virtual directories during installation. This topic contains information about the default IIS authentication settings and default Secure Sockets Layer (SSL) settings for the Client Access and Mailbox servers.

Client Access server

The following table lists the default settings on a stand-alone Exchange 2013 Client Access server.

Default Client Access server IIS authentication and SSL settings

Virtual directory	Authentication method	SSL settings	Management method
Default website	• Anonymous	• Required	IIS management console
aspnet_client	• Anonymous	• SSL required	IIS management

	authentication	• Requires 128-bit encryption	console
Autodiscover	• Anonymous authentication • Basic authentication • Windows authentication	• SSL required • Requires 128-bit encryption	Exchange Management Shell (Shell)
ecp	• Anonymous authentication • Basic authentication	• SSL required • Requires 128-bit encryption	Exchange admin center (EAC) or Shell
EWS	• Anonymous authentication • Windows authentication	• SSL required • Requires 128-bit encryption	Shell
Microsoft-Server-ActiveSync	• Basic authentication	• SSL required • Requires 128-bit encryption	EAC or Shell
OAB	• Windows authentication	• Not required	EAC or Shell
owa	• Basic authentication	• SSL required • Requires 128-bit encryption	EAC or Shell
PowerShell	• Anonymous authentication	• Not required	Shell
Rpc	• Basic authentication • Windows authentication	• SSL required • Requires 128-bit encryption	Shell
RpcWithCert	By default, all authentication methods are disabled.	• Required	

Mailbox server

The following table lists the default settings on a stand-alone Exchange 2013 Mailbox server.

Default Mailbox server IIS authentication and SSL settings

Virtual directory	Authentication method	SSL settings	Management method
Default website	• Anonymous authentication	• SSL required • Requires 128-bit encryption	This virtual directory can't be configured by the user.
PowerShell	• Anonymous authentication	• Not required	Shell

Outlook Web App

Exchange Server 2013 > Clients and mobile >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-24*

By default, when you install Microsoft Exchange 2013, you enable Outlook Web App. Microsoft Outlook Web App lets users access their Exchange mailbox from almost any Web browser.

The Client Access server role provides proxy and redirection services for Outlook Web App.

Note:
Outlook Web App was called Outlook Web Access in versions of Microsoft Exchange earlier than Exchange 2010.

For information about new features, see What's new in Exchange 2013. For information about the Client Access server role in Exchange 2013, see Client Access server.

Overview of Outlook Web App

Fully supported web browsers give users access to features such as conversation view, Inbox rules, the reading pane, and the Scheduling Assistant. Browsers that aren't fully supported can still be used, but users will see the light version of Outlook Web App, which has fewer features. For information about new features in Outlook Web App, see What's new for Outlook Web App in Exchange 2013.

Managing Outlook Web App

In Exchange 2013, the most common Outlook Web App management tasks can be accomplished in

the Exchange Administration Center (EAC). All these tasks, and many others, can be accomplished by using the Exchange Management Shell. You'll still use tools such as Internet Information Services (IIS) Manager for some tasks, for example, to configure Secure Sockets Layer (SSL) or set up simple URLs for users.

Outlook Web App mailbox policies

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-05*

Use Microsoft Outlook Web App mailbox policies to create organization-level policies to manage access to features in Outlook Web App.

Contents

Outlook Web App mailbox policies

In Exchange 2013, you can create multiple Outlook Web App mailbox policies and apply them to individual mailboxes. When an Outlook Web App mailbox policy is applied to a mailbox, it will override the settings of the virtual directory.

Outlook Web App features can also be managed by configuring the Outlook Web App virtual directories. Virtual directory settings will be used for any mailbox that a mailbox policy hasn't been applied to.

Creating or deleting Outlook Web App mailbox policies

A default Outlook Web App mailbox policy is created automatically when Exchange is installed. By default, all options are enabled on the default Outlook Web App mailbox policy. You can create as many Outlook Web App mailbox policies as necessary to meet the needs of your organization.

Note:
The default Outlook Web App mailbox policy isn't automatically applied to any mailboxes.

For information about creating or removing mailbox policies, see Create an Outlook Web App

mailbox policy and Remove an Outlook Web App mailbox policy from Exchange.

Configuring Outlook Web App mailbox policies

The default Outlook Web App mailbox policy has all options enabled by default. For information about configuring Outlook Web App mailbox policies, see View or configure Outlook Web App mailbox policy properties.

Applying Outlook Web App mailbox policies

Only one Outlook Web App mailbox policy can be applied to a mailbox.

If there's no Outlook Web App mailbox policy applied to a mailbox, the settings defined on the virtual directory will be applied.

An Outlook Web App mailbox policy can be applied to a mailbox by using the Exchange Administration Center (EAC) to modify an existing mailbox, or by using the Shell and the Set-CASMailbox cmdlet to apply a mailbox policy. For more information, see Apply or remove an Outlook Web App mailbox policy on a mailbox.

Outlook Web App mailbox policy procedures

Clients and mobile > Outlook Web App > Outlook Web App mailbox policies >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-09-22*

Create an Outlook Web App mailbox policy

Apply or remove an Outlook Web App mailbox policy on a mailbox

Remove an Outlook Web App mailbox policy from Exchange

View or configure Outlook Web App mailbox policy properties

Create an Outlook Web App mailbox policy

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-05-30*

You can create an Outlook Web App mailbox policy to apply a common set of policy settings. Outlook Web App mailbox policies are useful for applying and standardizing settings, for example, attachment settings, for specific groups of users.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Web App mailbox policies" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to create an Outlook Web App mailbox policy

1. In the EAC, click **Permissions** > **Outlook Web App policies**.
2. Click the **New** button.
3. Enter a name for your policy.
4. Use the check boxes to enable or disable features. By default, the most common features are displayed. To see all features that can be enabled or disabled, click **More options**.

Note:

Features settings for Outlook Web App mailbox policies override Outlook Web App virtual directory settings. You can change segmentation settings for individual users by using the **Set-CASMailbox** cmdlet in the Shell.

5. Click **Save** to save the policy.

Use the Shell to create an Outlook Web App mailbox

policy

This example creates an Outlook Web App mailbox policy named `Policy1`.

- In the Shell, run the following command.

```
New-OwaMailboxPolicy -Name Policy1
```

For more information about syntax and parameters, see New-OwaMailboxPolicy. For information about using the Shell to configure an Outlook Web App mailbox policy, see Set-OwaMailboxPolicy.

How do you know this worked?

To verify that you've successfully created an Outlook Web App mailbox policy:

- In the EAC, click **Permissions** > **Outlook Web App Policies**, and look for your new mailbox policy.

Apply or remove an Outlook Web App mailbox policy on a mailbox

Outlook Web App > Outlook Web App mailbox policies > Outlook Web App mailbox policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-12*

You can apply an Outlook Web App mailbox policy to one or more mailboxes or remove one using either the EAC or the Shell.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Outlook Web App mailbox policies" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Apply an Outlook Web App mailbox policy

Use the EAC to apply an Outlook Web App mailbox policy

1. In the EAC, click **Recipients** > **Mailboxes**.
2. In the work pane, click to select the mailbox that you want to apply an Outlook Web App mailbox policy to. You can also select multiple mailboxes.
3. **If you've selected one mailbox:**
   a. Scroll down in the details pane to **Email Connectivity** and click **View Details**.
   b. Click **Browse** to view and select from the available mailbox policies.
   c. Click **Save** to assign the selected policy to the selected mailbox.

If you've selected more than one mailbox:

   d. Scroll down in the details pane to **Outlook Web App** and click **Assign a policy**.
   e. Click **Browse** to view and select from the available mailbox policies.
   f. Click **Save** to assign the selected policy to the selected mailboxes.

Use the Shell to apply an Outlook Web App mailbox policy to an existing mailbox

This example applies the Outlook Web App mailbox policy named "Calendar" to the mailbox of the user tony@contoso.com.

```
Set-CASMailbox -Identity tony@contoso.com -
OwaMailboxPolicy:Calendar
```

For more information about syntax and parameters, see Set-CASMailbox.

Remove an Outlook Web App mailbox policy

Use the EAC to remove an Outlook Web App mailbox policy

1. In the EAC, click **Recipients** > **Mailboxes**.
2. In the work pane, click to select the mailbox that you want to remove an Outlook Web App mailbox policy from.
3. Scroll down in the details pane to **Email Connectivity** and click **View details**.

If a mailbox policy has been assigned, click **Clear** to remove it from the mailbox.

4. Click **Save** to save your changes.

Use the Shell to remove an Outlook Web App mailbox policy from an existing mailbox.

This example removes the Outlook Web App mailbox policy from mailbox of the user tony@contoso.com.

```
Set-CASMailbox -Identity tony@contoso.com -
OwaMailboxPolicy:$null
```

For more information about syntax and parameters, see Set-CASMailbox.

Remove an Outlook Web App mailbox policy from Exchange

Outlook Web App > Outlook Web App mailbox policies > Outlook Web App mailbox policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-15*

You can remove a Microsoft Outlook Web App mailbox policy from an Exchange organization by using either the EAC or the Shell.

For additional management tasks related to Outlook Web App mailbox policies, see Outlook Web App mailbox policies.

What do you need to know before you begin?

- Estimated time to complete each procedure: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Outlook Web App mailbox policies" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to remove an Outlook Web App mailbox policy

1. In the EAC, click **Permissions** > **Outlook Web App policies**.
2. In the result pane, click to select the mailbox policy you want to remove.

3. Click the **Delete** button.
4. In the confirmation window, click **Yes** to remove the mailbox policy, or click **No** to cancel.

Use the Shell to remove an Outlook Web App mailbox policy

This example removes an Outlook Web App mailbox policy named `Policy1`.

```
Remove-OwaMailboxPolicy -Name Policy1
```

For more information about syntax and parameters, see Remove-OwaMailboxPolicy.

How do you know this worked?

To verify that you've successfully removed an Outlook Web App mailbox policy:

- In the EAC, click **Permissions** > **Outlook Web App policies**. The policy you removed should no longer appear in the list.

View or configure Outlook Web App mailbox policy properties

Outlook Web App > Outlook Web App mailbox policies > Outlook Web App mailbox policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-11*

After you create an Outlook Web App mailbox policy, you can configure a variety of options to control the features available to users in Outlook Web App. For example, you can enable or disable Inbox rules or create a list of allowed file types for attachments.

What do you need to know before you begin?

- Estimated time to complete each procedure: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Outlook Web App mailbox policies" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to view or configure Outlook Web App mailbox policies

1. In the EAC, click **Permissions** > **Outlook Web App policies**.
2. In the result pane, click to select the mailbox policy you want to view or configure.
3. Click the **Edit** button.
4. On the **General** tab, you can view and edit the name of the policy.
5. On the **Features** tab, use the check boxes to enable or disable features. By default, the most common features are displayed. To see all features that can be enabled or disabled, click **More options**.

Note:

Features settings for Outlook Web App mailbox policies override Outlook Web App virtual directory settings. You can change segmentation settings for individual users by using the **Set-CASMailbox** cmdlet in the Shell.

Note:

The option to enable or disable the standard version of Outlook Web App by using the **Premium client** check box has been deprecated and will be removed from the settings. The standard version of Outlook Web App is always enabled.

6. On the **File Access** tab, use the check boxes to configure the file access and viewing options for users. File access lets a user open or view the contents of files attached to an email message.

File access can be controlled based on whether a user has signed in on a public or private computer. The option for users to select private computer access or public computer access is available only when you're using forms-based authentication. All other forms of authentication default to private computer access.

- **Direct file access** Select this check box if you want to enable direct file access. Direct file access lets users open files attached to email messages.
- **WebReady Document Viewing** Select this check box if you want to enable supported documents to be converted to HTML and displayed in a web browser.
- **Force WebReady Document Viewing when a converter is available** Select this check box if you want to force documents to be converted to HTML and displayed in a web browser before users can open them in the viewing application. Documents can be opened in the viewing application only if direct file access has been enabled.

7. On the **Offline access** tab, use the option buttons to configure offline access availability.
8. Click **Save** to update the policy.

Use the Shell to configure Outlook Web App mailbox policies

This example enables calendar access in the default mailbox policy.

```
Set-OwaMailboxPolicy -Identity Default -CalendarEnabled $true
```

For more information about syntax and parameters, see Set-OwaMailboxPolicy.

Use the Shell to view Outlook Web App mailbox policies

This example retrieves the properties of the Outlook Web App mailbox policy `Executives` in the organization `Fabrikam`.

```
Get-OwaMailboxPolicy -Identity Fabrikam\Executives
```

For more information about syntax and parameters, see Get-OwaMailboxPolicy.

How do you know this worked?

To verify that you've successfully edited an Outlook Web App mailbox policy:

1. In the EAC, click **Permissions** > **Outlook Web App Policies**, and then choose a specific Outlook Web App mailbox policy.
2. Click the **Edit** button to view the properties of the mailbox policy.
3. Click **Save** or **Cancel** to close the properties page.

Integrate Outlook Web App with Lync Server

Exchange Server 2013 > Clients and mobile > Outlook Web App >

Topic Last Modified: *2013-02-17*

Outlook Web App supports the integration of Outlook Web App and Microsoft Lync Server 2013. When integration with Lync Server 2013 is configured and instant messaging (IM) is enabled in Outlook Web App, users can initiate or respond to IM sessions by using Outlook Web App.

To integrate Outlook Web App with Lync Server 2013, see Integrating Microsoft Lync Server 2013

and Microsoft Outlook Web App 2013.

View or configure Outlook Web App virtual directories

Exchange Server 2013 > Clients and mobile > Outlook Web App >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-08-12*

You can use the EAC or the Shell to view or configure the properties of an Outlook Web App virtual directory.

Warning:

In Exchange Online, administrators don't have the ability to view or configure Outlook Web App virtual directories.

If you use the Shell to view the properties of an Outlook Web App virtual directory, the information returned is a subset of the information that's available. For example, if you use the **Get-OWAVirtualDirectory** cmdlet to view properties, Exchange returns the following information:

- Virtual directory name
- Server name
- Exchange server version

You can also retrieve information for a specific virtual directory on a specific server by using the available parameters. For more information about the parameters for the **Get-OWAVirtualDirectory** cmdlet, see Get-OwaVirtualDirectory.

If you use the EAC to view the properties of an Outlook Web App virtual directory, you'll be able to view most of the properties for the virtual directory that you're viewing.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Outlook Web App virtual directories" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to view or configure Outlook Web App virtual directory properties

1. In the EAC, click **Servers** > **Virtual Directories**.

You can use the drop-down lists to select the server and virtual directory type. By default, all servers and virtual directories are displayed.

2. In the result pane, click to select the virtual directory you want to view or edit, then click **Edit**.
3. On the **General** tab, you can view the properties of the Outlook Web App default website and specify an external URL and an internal URL. View or select the following options:
   - **Server** (Read-only.) **Server** displays the name of the server that hosts the Outlook Web App virtual directory.
   - **Version** (Read-only.) **Version** displays the version of the Exchange server that the virtual directory is on.
   - **Web site** (Read-only.) **Web site** displays the name of the website.
   - **Outlook Web App version** (Read-only.) **Outlook Web App version** displays the Exchange server version.
   - **Modified** (Read-only.) **Modified** displays the last date and time that the virtual directory was modified.
   - **Internal URL** In this text box, specify the URL used to access this website from an internal network. An internal URL is configured automatically during Exchange 2013 Setup. The default internal URL setting for an Internet-facing or non-Internet-facing server is https://<Computer Name>/owa.
   - **External URL** In this text box, specify the URL used to access the website from the Internet. By default, **External URL** is blank. For Internet-facing Client Access servers, **External URL** should be set to the value published in DNS for that Active Directory site. For Exchange 2013 servers that don't have an Internet presence, the **External URL** setting should remain blank.
4. On the **Authentication** tab, specify the authentication methods, sign-in format, and sign-in domain.
   - **Use one or more standard authentication methods** Select this option to use one or more of the following standard authentication methods:

Integrated Windows authentication This method requires that users have a valid Windows Server 2008 or Windows Server 2012 user account name and password to access information. Users aren't prompted for their account names and passwords. Instead, the server negotiates with the Windows security packages installed on the client computer. Integrated Windows authentication enables the server to authenticate users without prompting them for information and without transmitting information that isn't encrypted over the network. For this method to work, the client computer must be a member of the same domain as the servers running Exchange, or of a domain that's trusted by the domain that the Exchange server is in.

Digest authentication for Windows domain servers This method transmits passwords over the network as a hash value for additional security. Digest authentication can be used only in Windows Server 2008 and Windows Server 2012 domains for users who have an account that's stored in Active Directory. For more information about Digest authentication, see the Windows Server documentation.

Basic authentication (password is sent in clear text) This method is a simple authentication mechanism defined by the HTTP specification that encodes a user's sign-in name and password before the user's credentials are sent to the server. To make sure that the password is as secure as possible, you should use Secure Sockets Layer (SSL) encryption between client computers and the server that has the Client Access server role installed.

- **Use forms-based authentication** Forms-based authentication provides enhanced security for Outlook Web App virtual directories. Forms-based authentication creates a sign-in page for Outlook Web App. You can configure the type of sign-in prompt used by forms-based authentication. For example, you can configure forms-based authentication to require users to provide their domain and user name information, in the domain\user name format on the Outlook Web App sign-in page.

Important:

Forms-based authentication won't provide a secure channel unless SSL is enabled.

Select one of the following:

Domain\user name This requires the user to enter their domain and user name in the format domain\user name. For example, for a user named Kweku in the domain Contoso, the sign-in would be contoso\kweku.

User principal name (UPN) If the user principal name (UPN) sign-in format is specified, the **User name** box on the Outlook Web App sign-in page guides users to enter their email address, for example, kweku@contoso.com. If a user's UPN isn't identical to the email address, the user can't access Outlook Web App by using the **PrincipalName** sign-in prompt. It's a best practice to use the **PrincipalName** sign-in prompt only if users' UPNs match their email addresses.

User name only The user enters their user name only, without the domain name, for example, Kweku. If you use the **User name only** sign-in prompt for forms-based authentication, you must also specify the **Logon Domain** property. The **Logon Domain** property determines the default domain to use when a user tries to sign in to Outlook Web App. For example, if the default domain is Contoso, and a domain user named Kweku signs in to Outlook Web App, only Kweku must be entered as the user name. The server will use the default domain Contoso. If the user isn't a member of the Contoso domain, the domain and user name must be entered.

5. On the **Features** tab, specify the features that you want to enable or disable for Outlook Web App users on a virtual directory.

Note:

Features settings for individual users override virtual directory settings. You can change segmentation settings for individual users by using the **Set-CASMailbox** cmdlet or by using Outlook Web App mailbox policies. For more information, see Outlook Web App mailbox policies.

Use the check boxes to enable or disable features. By default, the most common features are displayed. To see all features that can be enabled or disabled, click **More options**.

Note:
The option to enable or disable the standard version of Outlook Web App by using the **Premium client** check box has been deprecated and will be removed from the settings. The standard version of Outlook Web App is always enabled.

6. On the **File access** tab, use the check boxes to configure the file access and viewing options for users. File access lets a user open or view the contents of files attached to an email message.

File access can be controlled based on whether a user has signed in on a public or private computer. The option for users to select private computer access or public computer access are available only when you're using forms-based authentication. All other forms of authentication default to private computer access.

- **Direct file access** Select this check box if you want to enable direct file access. Direct file access lets users open files attached to email messages.
- **WebReady Document Viewing** Select this check box if you want to enable supported documents to be converted to HTML and displayed in a web browser.
- **Force WebReady Document Viewing when a converter is available** Select this check box if you want to force documents to be converted to HTML and displayed in a web browser before users can open them in the viewing application. Documents can be opened in the viewing application only if direct file access has been enabled.

7. Click **Save** to update the policy.

Use the Shell to configure Outlook Web App virtual directory properties

This example enables forms-based authentication on the default Outlook Web App virtual directory on the server Contoso.

```
set-OwaVirtualDirectory -Identity "Contoso\owa (default web site)" -FormsAuthentication $true
```

For more information about syntax and parameters, see Set-OwaVirtualDirectory.

Use the Shell to view Outlook Web App virtual directory properties

This example lets you view the properties for all Outlook Web App virtual directories in all Internet Information Services (IIS) websites on all computers that have the Client Access server role installed in an Exchange organization.

```
Get-OWAVirtualDirectory
```

This example lets you view the properties for an Outlook Web App virtual directory on the default IIS website on the local Exchange server.

```
Get-OWAVirtualDirectory -identity "<Exchange Server Name>
\owa (default web site)"
```

This example lets you view the properties for all Outlook Web App virtual directories on an IIS website on a specific Exchange server.

```
Get-OWAVirtualDirectory -server <Exchange Server Name>
```

This example lets you view the values of the properties for every Outlook Web App virtual directory in all IIS websites on all Client Access servers in an Exchange organization.

```
Get-OWAVirtualDirectory | format-list
```

For more information about syntax and parameters, see Get-OwaVirtualDirectory.

How do you know this worked?

To verify that you've successfully edited an Outlook Web App virtual directory:

1. In the EAC, click **Servers** > **Virtual Directories**, and then choose a specific Outlook Web App virtual directory.
2. Click the **Edit** button to view the properties of the virtual directory.
3. Click **Save** or **Cancel** to close the properties page.

Simplify the Outlook Web App URL

Exchange Server 2013 > Clients and mobile > Outlook Web App >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-25*

You can use Internet Information Services (IIS) Manager to simplify the Microsoft Outlook Web App URL that users use to access their Exchange Server 2013 mailbox.

To simplify access to Outlook Web App for your users, you may want to configure the Outlook Web App Web page, which is usually the default website in IIS, to automatically redirect users to https. The procedure in the "Use IIS Manager to simplify the Outlook Web App URL and force redirection to SSL" section redirects a request for http://*server* to https://*server*/owa. To help secure the information that's sent between the client and the server, the default Web site is set to require

Secure Sockets Layer (SSL) at installation.

When you configure redirection from a top-level directory in Windows Server 2008, the settings are propagated to lower-level directories. For example, when you configure redirection on the default website to the /owa virtual directory, the settings that you configure also appear on the HTTP Redirect page of all the virtual directories, such as /Autodiscover, /Exchange, and /Public. Therefore, you must remove redirection from all the virtual directories except the one that you want redirected.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "IIS Manager" entry in the Outlook Web App Permissions section of the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use IIS Manager to simplify the Outlook Web App URL and force redirection to SSL

1. Start IIS Manager.
2. Expand the local computer, expand **Sites**, and then click **Default Web Site**.
3. At the bottom of the Default Web Site Home pane, click **Features View** if this option isn't already selected.
4. In the **IIS** section, double-click **HTTP Redirect**.
5. Select the **Redirect requests to this destination** check box.
6. Type the absolute path of the /owa virtual directory. For example, type **https://mail.contoso.com/owa**.
7. Under **Redirect Behavior**, select the **Only redirect requests to content in this directory (not subdirectories)** check box.
8. In the **Status code** list, click **Found (302)**.
9. In the Actions pane, click **Apply**.
10. Click **Default Web Site**.
11. In the Default Web Site Home pane, double-click **SSL Settings**.
12. In **SSL Settings**, clear **Require SSL**.

Note:

If you don't clear **Require SSL**, users won't be redirected when they enter an unsecured URL. Instead, they'll get an access denied error.

13. For the new settings to take effect, open a Command Prompt window, and then type **iisreset /noforce** to restart IIS.

How do you know this worked?

To verify that you have successfully simplified the Outlook Web App URL and redirected it to an SSL connection, do the following:

1. Open a web browser and enter the new URL for Outlook Web App, using the format http://<*URL*>.
2. You should be redirected to the Outlook Web App sign-in page through an SSL connection.

Use IIS Manager to remove redirection from a virtual directory

To remove redirection from a virtual directory, perform the following steps:

1. Start IIS Manager.
2. Navigate to the virtual directory.
3. Double-click the **HTTP Redirect** icon in the **Features** view of the virtual directory.
4. Clear the **Redirect requests to this destination** check box.
5. In the Actions pane, click **Apply**.
6. For the new settings to take effect, open a Command Prompt window, and then type **iisreset /noforce** to restart IIS.

You may not be able to use the procedure above to remove redirection from a virtual directory that doesn't have a physical path, such as /Exchange, /Exchweb, or /Public. Use the following procedure to remove redirection from a virtual directory that doesn't appear in IIS Manager.

1. Open a Command Prompt window.
2. Navigate to <*Window directory*>\System32\Inetsrv.
3. Run the following commands:
   a. `appcmd set config "Default Web Site/autodiscover" /section:httpredirect /enabled:false -commit:apphost`
   b. `appcmd set config "Default Web Site/ecp" /section:httpredirect /enabled:false -commit:apphost`
   c. `appcmd set config "Default Web Site/ews" /section:httpredirect /enabled:false -commit:apphost`
   d. `appcmd set config "Default Web Site/owa" /section:httpredirect /enabled:false -commit:apphost`
   e. `appcmd set config "Default Web Site/oab" /section:httpredirect /enabled:false -commit:apphost`
   f. `appcmd set config "Default Web Site/powershell" /section:httpredirect /enabled:false -`

```
commit:apphost
```

g. `appcmd set config "Default Web Site/rpc" /section:httpredirect /enabled:false -commit:apphost`

h. `appcmd set config "Default Web Site/rpcwithcert" /section:httpredirect /enabled:false -commit:apphost`

i. `appcmd set config "Default Web Site/Microsoft-Server-ActiveSync" /section:httpredirect /enabled:false -commit:apphost`

4. Finish by running the command `iisreset/noforce`.

When you configure redirection from a top-level directory, a web.config file may be created under <*drive*>\Program Files\Microsoft\Exchange Server\<*version*>\ClientAccess\oab. If this has happened and you later remove redirection, Outlook may freeze when users click **Send and Receive**. To avoid this happening after you remove redirection, delete the web.config file from <*drive*>\Program Files\Microsoft\Exchange Server\<*version*>\ClientAccess\oab.

Create a theme for Outlook Web App

Exchange Server 2013 > Clients and mobile > Outlook Web App >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-05-01*

A *theme* defines the background color, fonts, highlight colors, icons, and header that are used by Microsoft Outlook Web App. Each theme is a collection of media files and cascading style sheets (.css) files that are stored on the Microsoft Exchange server in the installation directory in \Client Access\OWA*version*\Owa2\resources\themes. Each theme is stored in its own subdirectory of \themes.

The default theme is found in \Client Access\OWA*version*\Owa2\resources\themes\base. Each theme folder contains all the files that are needed to define a theme. These files include CSS files, graphics, and an .xml file that defines the name of the theme. Additional themes are created by copying all the files from one theme into a new folder and modifying those files as needed.

By default, multiple themes are installed when you install Exchange Server 2013, as follows:

- CSS (.css) files define colors, gradients, and fonts.
- Image (.png) files provide the icons and other graphic elements. If you edit any of the icons, don't change their size. If you change the size of any graphic elements, test your changes to verify that the elements still fit together correctly.

These files are stored on the Client Access server in the installation directory in \Client Access\OWA\<*version*>\themes. Each theme is stored in a subdirectory of themes. You can create additional themes by copying an existing theme and modifying the copy.

After you create a theme, you may also want to Customize the Outlook Web App sign-In, language

selection, and error pages.

Note:
The light version of Outlook Web App doesn't support themes.

Caution:
If you have multiple servers that support Outlook Web App, you must copy your custom theme to each server. You should also create a backup copy of your custom theme. If you reinstall or upgrade Exchange, all files in the themes folders will be overwritten. You'll have to copy your theme back to the appropriate folder after the reinstallation or upgrade is complete.
Make backup copies of all the files that you'll be changing before you start to create your custom theme.

What do you need to know before you begin?

- Estimated time to complete this task: one hour.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Outlook Web App virtual directories" entry in the Clients and mobile devices permissions topic.
- You need local server administrator access to perform these procedures.
- You'll need a text editor to change the default colors, and a graphics editor to change the images. If you must match a specific color and you can't find a match for it at Color Table, you can use an image editing tool to sample a color and determine its HTML RGB value.
- As a best practice, we recommend that you use the following guidelines any time that you change or create an Outlook Web App theme:
  - If you decide to edit an existing theme, make backup copies of the original files before you start editing them.
  - Do not delete the folder \Client Access\OWA*version*\Owa2\resources\themes\base or any of the files in it.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

How do you do this?

Step 1: Create a new Outlook Web App theme

To start, you'll create a folder for a new theme, and then copy the files from an existing theme into the new folder.

1. Log on to the Exchange server that is hosting the Outlook Web App virtual directory by using an account that has been delegated membership in the local Administrators group.

2. Open Windows Explorer, and then find the Exchange server installation directory.
3. In \Client Access\OWA*version*\Owa2\resources\themes, create a new folder and name it, for example, Fourth Coffee.
4. Copy all the files from another theme to the new folder.

Step 2: Name your new theme

To set the display name for your new theme, do the following:

1. Open the copy of themeinfo.xml that's in the custom theme folder you just created.
2. Find the theme `displayname` value, and change the value to the name you want to use. For example `displayname = "Fourth Coffee Theme"`.
3. Save and close themeinfo.xml.

Step 3: Change the sort order of your new theme (optional)

If you want, you can change the sort order of the new theme by editing the themeinfo.xml file. The sort order determines the theme position in the **Change theme** panel in the Settings menu.

To change the sort order of the new theme by using the themeinfo.xml file, do the following:

1. Open the copy of themeinfo.xml that's in the custom theme folder.
2. Find the theme `sortorder` value, and change the value to reflect where you want your new theme to appear in the list. The themes will be ordered by the numeric value in increasing order. By default, the base theme is the first one and its `sortorder` value is "0". For example `sortorder="<number>"`.
3. Save and close themeinfo.xml.

Step 4: Modify your new theme

Now that you've copied over the files and have named your theme, you can customize it. The following elements can be customized in an Outlook Web App theme:

- Image files, which define the header area and icons.
- CSS files, which define fonts and colors.

Image files

Theme images are stored in two folders in \themes\<*theme name*>\images\. The \images\0 folder contains images that will be used in left-to-right languages (like English), and languages that are read from right to left will use the images in the \images\rtl folder.

Note:
Some of the images in the \images\rtl folder are the same the images as in the \images\0 folder, but they're mirrored.

To customize the theme, you can use an image editing tool to open and modify the following

images:

- Headerbgmain.png
  - This is the main header image. We recommend that the image doesn't exceed the header height of 30 pixels. The default theme doesn't use a background image, so this image is transparent. For an example of a theme that has a custom background image, see the image in the **Blueprint** theme folder.
- Headerbgright.png
  - This is used as a tiling image behind the header. The default theme doesn't use a tiling background image, so this image is transparent. For an example of a theme that has a custom tiling background image, see the image in the **Blueprint** theme folder.
- sprite1.mouse.png
  - This contains most of the images used in a theme. You can change the color of the images to match your theme, and also change the default Outlook Web App text logo.
  - To avoid any issues, don't change the size of any individual icons in the sprite, and make sure that it's saved as a transparent .png file.
- themepreview.png
  - This image will be used to represent the theme in the **Change theme** panel in the Settings menu in Outlook Web App.

Colors and fonts

Cascading style sheet (.css) files define the colors and fonts used in a theme and are stored in multiple folders under \themes\<*theme name*>. The \<*theme name*>\0 folder contains .css files that will be used in left-to-right languages (like English), and languages that are read from right to left will use the .css files in the \<*theme name*>\rtl folder. There are also language-specific folders, (for example, \ja, \ko, \zhs, and \zht) that contain .css files to be used with those languages.

Start by modifying the \<*theme name*>\0 folder. There are four colors used throughout each theme that can be customized.

- BrandColor: #0072C6
- NavBarHoverColor: #4C9CD7
- UnreadColor: #2A8DD4
- FocusColor: #DFEDFA

You can use a text editor like Notepad to search for and replace all the instances of these values with the colors of your theme in the following two files: owa2styles.mouseCSS and owa2styles2.mouseCSS. This has to be done in every folder in your new theme that contains those .css files.

Step 5: Set the default Outlook Web App theme

Setting a new default theme will only affect users who haven't changed their theme through the Settings menu in Outlook Web App.

To force all users to use the default theme, you must disable theme selection in addition to setting

a default theme.

Use the Shell to set the default theme for Outlook Web App

This example sets the default theme for Outlook Web App, where the server name is `fourthcoffee`, the virtual directory name is `owa`, the website name is `default web site`, and the theme is in the folder named `Custom`.

```
set-owavirtualdirectory -identity "fourthcoffee\owa
(default web site)" -defaulttheme Custom
```

For detailed syntax and parameter information, see Set-OwaVirtualDirectory.

Use the Shell to disable theme selection for Outlook Web App

This example disables theme selection in Outlook Web App, where the server name is `fourthcoffee`, the virtual directory name is `owa`, and the website name is `default web site`.

```
set-owavirtualdirectory -identity "fourthcoffee\owa
(default web site)" -themeselectionenabled $false
```

You can also complete both commands at the same time as shown in the following example:

```
set-owavirtualdirectory -identity "fourthcoffee\owa
(default web site)" -defaulttheme Custom -
themeselectionenabled $false
```

For detailed syntax and parameter information, see Set-OwaVirtualDirectory.

Step 6: Run iisreset/noforce to save your changes

If you add or change a theme, change the name of a theme, or change the sort order of a theme, you must stop and start Internet Information Services (IIS) for the change to take effect. To do this, open a Command Prompt window on the server where you've created your new theme, and run the command **iisresest /nforce**.

How do you know this task worked?

1. Sign in to Outlook Web App using the virtual directory on the server where you've created your new theme. If you're testing the changes to the default website on the Exchange server that's hosting the Outlook Web App virtual directory, you can test your theme by opening Internet Explorer and entering the URL https://localhost/owa.
2. Switch to your custom theme by selecting the Settings menu > **Change theme** and selecting your custom theme.

If you don’t see your latest changes and have run iisreset/noforce

1. On the Internet Explorer toolbar, select the Settings menu > **Internet options**.
2. On the **General** tab, under **Browsing history**, select **Delete**, and then verify that **Temporary Internet files and website files** is checked. Then select **Delete** to remove those files.
3. Select **OK** to close **Internet options**.
4. Select **Refresh** to see your changes.

You may have to repeat these steps to see your changes every time that you make a change to the theme files. If you’re making several changes, you can leave Outlook Web App open and repeat running **iisreset/noforce** on the server and clearing temporary files from Internet Explorer as needed.

Customize the Outlook Web App sign-In, language selection, and error pages

Exchange Server 2013 > Clients and mobile > Outlook Web App >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-05-01*

This topic explains how to customize the color and images of the sign-in, language selection, and error pages for Outlook Web App.

The Outlook Web App sign-in, language selection, and error pages are created based on graphics and .css files in the themes resources folder. Outlook Web App uses only one set of sign-in, language selection, and error pages for all themes. Any modifications to those pages will be seen by all users. You can find the theme resources folder in the Exchange installation directory at V15 \FrontEnd\HttpProxy\owa\auth\version\themes\resources. You’ll need a text editor to change the default colors, and a graphics editor to change the images. If you must match a specific color and you can't find a match for it at Color Table, you can use an image editing tool to sample a color and determine its HTML RGB value.

If you have multiple servers supporting Outlook Web App and want them all to use the same sign-in, language, and error pages, you must copy the modified files to each server. You should also create a back-up copy of your customized files. If you reinstall or upgrade Exchange, all files in the themes folders will be overwritten. You'll have to copy your customized files back to the appropriate folder after the reinstallation or upgrade is complete.

Important:

Back up copies of all the files that you'll be changing before you start to create your custom sign-in and sign-out pages.

For information about creating a custom theme, see Create a theme for Outlook Web App.

What do you need to know before you begin?

- Estimated time to complete this task: 45 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Graphics editor" entry under “Outlook Web App Permissions” in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Customize the color of the sign-in page

1. Log on to the Exchange server and use Windows Explorer to go to the Exchange server installation directory and find \V15\FrontEnd\HttpProxy\owa\auth\<version>\themes\resources.
2. Use a text editor, such as Notepad, to open logon.css.
3. Search for the default color value #0072c6 and replace it with the HTML RGB value for the color you want to use. You can find HTML RGB values here: Color Table.
4. Save and close the file.

Customize the color of the error page

1. Log on to the Exchange server and use Windows Explorer to go to the Exchange server installation directory and find \V15\FrontEnd\HttpProxy\owa\auth\<version>\themes\resources.
2. Use a text editor, such as Notepad, to open errorFE.css.
3. Search for the default color value #0072c6 and replace it with the HTML RGB value for the color you want to use. You can find HTML RGB values here: Color Table.
4. Save and close the file.

Customize the color of the language selection page

1. Log on to the Exchange server and use Windows Explorer to go to the Exchange server installation directory and find \V15\Client Access\OWA\version\Owa2\resources\styles.
2. Use a text editor, such as Notepad, to open languageselection.css.

3. Search for the default color value #0072c6 and replace it with the HTML RGB value for the color you want to use. You can find HTML RGB values here: Color Table.
4. Save and close the file.

Customize the images on the sign-in and error pages

Use an image editing tool to open and edit the images used to build the sign-in and error pages.

1. Log on to the Exchange server and use Windows Explorer to go to the Exchange server installation directory and find \V15\FrontEnd\HttpProxy\owa\auth\<version>\themes\resources.
2. Use a graphics editor to open and modify the following files:
   - owa_text_blue.png, to change the "Outlook Web App" text logo.
   - olk_logo_white.png, to change the app logo in the left bar.
   - olk_logo_white_cropped.png, to change the image in the left side panel of the error page.
   - sign_in_arrow.png, to change the icon left of the "sign in" button.
   - olk_exchange_text_blue.png, to change the "Outlook Mobile" logo on tnarrow layout.
   - olk_logo_white_small.png is used in tnarrow.
   - olk_exchange_text_stacked_white_small.png is used in tnarrow.
3. Search for the default color value #0072c6 and replace it with the HTML RGB value for the color you want to use. You can find HTML RGB values here: Color Table.
4. Save and close the file.

How do you know this worked?

Open the Outlook Web App sign-in page in Internet Explorer. If you're testing the changes to the default website on the server that's hosting the Outlook Web App virtual directory, you can test them by opening Internet Explorer and entering the URL https://localhost/owa.

If you don't see your changes, do the following:

1. On the toolbar, select **Settings** > **Internet options** > **General**.
2. Under **Browsing history**, select **Delete**.
3. Select **Temporary Internet files and website files**, and then select **Delete**.
4. When Internet Explorer is finished deleting, select **OK** to close **Internet options**.
5. Refresh the browser window.

Tip:
To see the effects of your changes, you can keep the .css file that you're editing open and refresh the browser window after saving each change.

Configuring push notifications proxying for OWA for Devices

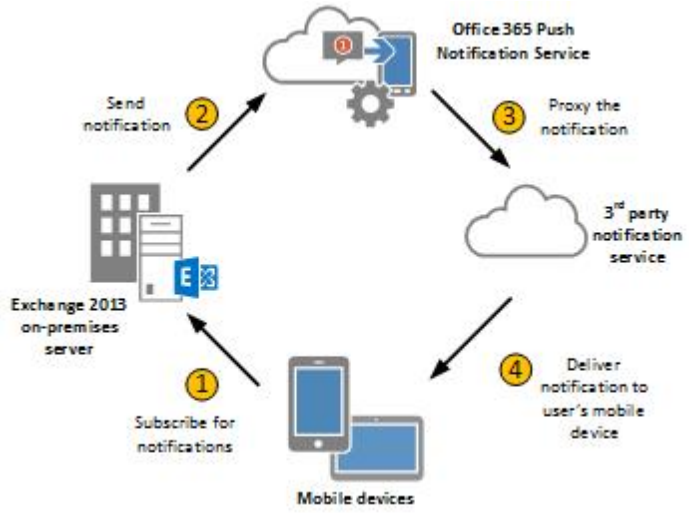
Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-02-15*

Enabling push notifications for OWA for Devices (OWA for iPhone and OWA for iPad) for an on-premises deployment of Microsoft Exchange 2013 lets a user receive updates on the Outlook Web App icon on his or her OWA for iPhone and OWA for iPad indicating the number of unseen messages in the user's inbox. If push notifications aren't configured and enabled, a user with OWA for Devices has no way of knowing that unseen messages are in the inbox without launching the app. When a new message is available, the OWA for Devices badge is updated on the user's device and looks like the following badge.

How do I enable push notifications?

In order to enable push notifications, the on-premises Exchange 2013 servers must connect to the Office 365 Push Notification Service to send push notifications to iPhones and iPads. Exchange 2013 on-premises servers route their update notifications through the Office 365 notification services to remove the need for enrolling developer accounts with third-party push notification services. The following diagram shows the process of how iPhone and iPad users can get badge updates for unseen messages.



To enable push notifications, the admin must:

1. Enroll your organization in Office 365 for business.
2. Update all on-premises servers to Exchange Server 2013 Cumulative Update 3 (CU3) or later.
3. Set up On-premises Exchange 2013 to Office 365 Authentication
4. Enable push notifications from the on-premises Exchange Server 2013 to Office 365 and verify that push notifications are working.

Enroll your organization in Office 365 for business

Office 365 is a cloud-based service that is designed to help meet your organization's needs for robust security, reliability, and user productivity. Office 365 refers to subscription plans that include access to Office applications plus other productivity services that are enabled over the Internet (cloud services), such as Lync web conferencing and Exchange Online hosted email for business.

Many Office 365 plans also include the desktop version of the latest Office applications, which users can install across multiple computers and devices. All Office 365 plans are paid for on a subscription basis, monthly or annually. To find out more or to enroll in Office 365 for your organization, see What is Office 365 for business?. For more about each of the services offered through Office 365, see Office 365 Service Descriptions.

Update to CU3 or later

Cumulative Update 3 (CU3) for Exchange Server 2013 resolves issues that were found in Exchange Server 2013 since the software was released since RTM. It contains all of the issues and fixes in CU1 and CU2 and includes other fixes and updates since CU2 was released. This update is highly recommended for all Exchange Server 2013 on-premises customers but is required for push notifications. To read about cumulative updates, including CU3, see Updates for Exchange 2013.

Set up On-premises Exchange 2013 to Office 365

Authentication

Using a single, standardized method for server-to-server authentication is the approach used by Exchange Server 2013. Exchange Server 2013 (as well as Lync Server 2013 and SharePoint 2013) and Office 2013 support the OAuth (Open Authorization) protocol for server-to-server authentication and authorization. With OAuth, a standard authorization protocol used by a number of major websites, user credentials and passwords aren't passed from one computer to another. Instead, authentication and authorization are based on the OAuth security tokens; these tokens grant access to a specific set of resources for a specific amount of time.

OAuth authentication typically involves three components: a single authorization server and the two realms that need to communicate with one another. Security tokens are issued by the authorization server (also known as a security token server) to the two realms that need to communicate; these tokens verify that communications originating from one realm should be trusted by the other realm. For example, the authorization server might issue tokens that verify that users from a specific Lync Server 2013 realm are able to access a specified Exchange 2013 realm, and vice versa.

Tip:
A realm is a security container.

However, for on-premises server-to-server authentication there is no need to use a third-party token server. Server products such as Lync Server 2013 and Exchange 2013 each have a built-in token server that can be used for authentication purposes with other Microsoft servers (such as SharePoint Server) that support server-to-server authentication. For example, Lync Server 2013 can issue and sign a security token by itself, then use that token to communicate with Exchange 2013. In a case like this, there is no need for a third-party token server.

In order to configure server-to-server authentication for an on-premises implementation of Exchange Server 2013 to Office 365, you must complete two steps:

Step 1 – Assign a certificate to the built-in token issuer of the on-premises Exchange Server. First, an on-premises Exchange admin must use the following Exchange Management Shell script to create a certificate if one wasn't created before and assign it to the built-in token issuer of the on-premises Exchange Server. This is a one-time process; after a certificate has been created, that certificate should be reused for other authentication scenarios and not replaced. Make sure to update the value of *$tenantDomain* to be the name of your domain. To do this, copy and paste the following code.

⚠Warning:

Copying and pasting the code into a text editor like Notepad and saving it with a .ps1 extension makes it easier to run Shell scripts.

```
# Make sure to update the following $tenantDomain with your
Office 365 tenant domain.
$tenantDomain = "Fabrikam.com"
# Check whether the cert returned from Get-AuthConfig is
valid and keysize must be >= 2048
$c = Get-ExchangeCertificate | ?{$_.CertificateDomains -eq
$env:USERDNSDOMAIN -and $_.Services -ge "SMTP" -and
$_.PublicKeySize -ge 2048 -and $_.FriendlyName -match
"OAuth"}
If ($c.Count -eq 0)
{
    Write-Host "Creating certificate for oAuth..."
    $ski = [System.Guid]::NewGuid().ToString("N")
    $friendlyName = "Exchange S2S OAuth"
    New-ExchangeCertificate -FriendlyName $friendlyName -
DomainName $env:USERDNSDOMAIN -Services Federation -KeySize
2048 -PrivateKeyExportable $true -SubjectKeyIdentifier $ski
    $c = Get-ExchangeCertificate | ?{$_.friendlyname -eq
$friendlyName}
```

```
}
ElseIf ($c.Count -gt 1)
{
    $c = $c[0]
}
$a = $c | ?{$_.Thumbprint -eq (get-
authconfig).CurrentCertificateThumbprint}
If ($a.Count -eq 0)
{
    Set-AuthConfig -CertificateThumbprint $c.Thumbprint
}
Write-Host "Configured Certificate Thumbprint is:"(get-
authconfig).CurrentCertificateThumbprint
# Export the certificate
Write-Host "Exporting certificate..."
if((test-path $env:SYSTEMDRIVE\OAuthConfig) -eq $false)
{
    md $env:SYSTEMDRIVE\OAuthConfig
}
cd $env:SYSTEMDRIVE\OAuthConfig
$oAuthCert = (dir Cert:\LocalMachine\My) | where
{$_.FriendlyName -match "OAuth"}
$certType =
[System.Security.Cryptography.X509Certificates.X509ContentT
ype]::Cert
$certBytes = $oAuthCert.Export($certType)
$CertFile = "$env:SYSTEMDRIVE\OAuthConfig\OAuthCert.cer"
[System.IO.File]::WriteAllBytes($CertFile, $certBytes)
# Set AuthServer
$authServer = Get-AuthServer MicrosoftSts;
if ($authServer.Length -eq 0)
{
    Write-Host "Creating AuthServer Config..."
    New-AuthServer MicrosoftSts -AuthMetadataUrl https://
accounts.accesscontrol.windows.net/metadata/json/1/?
realm=$tenantDomain
}
elseif ($authServer.AuthMetadataUrl -ne "https://
accounts.accesscontrol.windows.net/metadata/json/1/?
```

```
realm=$tenantDomain")
{
    Write-Warning "AuthServer config already exists but the
AuthMetdataUrl doesn't match the appropriate value.
Updating..."
    Set-AuthServer MicrosoftSts -AuthMetadataUrl https://
accounts.accesscontrol.windows.net/metadata/json/1/?
realm=$tenantDomain
}
else
{
    Write-Host "AuthServer Config already exists."
}
Write-Host "Complete."
```

The expected result should be similar to the following output.

```
Configured Certificate Thumbprint is:
7595DBDEA83DACB5757441D44899BCDB9911253C
Exporting certificate...
Complete.
```

⚠Warning:

Before you continue, the Windows Azure Active Directory Module for Windows PowerShell cmdlets is required. If the Windows Azure Active Directory Module for Windows PowerShell cmdlets (previously known as the Microsoft Online Services Module for Windows PowerShell) hasn't been installed, you can install it from Manage Windows Azure AD using Windows PowerShell.

Step 2 – Configure Office 365 to communicate with Exchange 2013 on-premises. Configure the Office 365 server that Exchange Server 2013 will communicate with to be a partner application. For example, if Exchange Server 2013 on-premises needs to communicate with Office 365, you need to configure Exchange on-premises to be a partner application. A partner application is any application that Exchange 2013 can directly exchange security tokens with, without having to go through a third-party security token server. An on-premises Exchange 2013 administrator must use the following Exchange Management Shell script to configure the Office 365 tenant that Exchange 2013 will communicate with to be a partner application. During execution, there will be a prompt to enter the administrator user name and password of the Office 365 tenant domain—for example, administrator@fabrikam.com. Make sure to update the value of *$CertFile* to the location of the certificate if not created from the previous script. To do this, copy and paste the following code.

```
# Make sure to update the following $CertFile with the path
```

```
to the cert if not using the previous script.
$CertFile = "$env:SYSTEMDRIVE\OAuthConfig\OAuthCert.cer"
If (Test-Path $CertFile)
{
    $ServiceName = "00000002-0000-0ff1-ce00-000000000000";
    $objFSO = New-Object -ComObject
Scripting.FileSystemObject;
    $CertFile = $objFSO.GetAbsolutePathName($CertFile);
    $cer = New-Object
System.Security.Cryptography.X509Certificates.X509Certifica
te
    $cer.Import($CertFile);
    $binCert = $cer.GetRawCertData();
    $credValue =
[System.Convert]::ToBase64String($binCert);
    Write-Host "Please enter the administrator user name
and password of the Office 365 tenant domain..."
    Connect-MsolService;
    Import-Module msonlineextended;
    Write-Host "Adding a key to Service Principal..."
    $p = Get-MsolServicePrincipal -ServicePrincipalName
$ServiceName
    New-MsolServicePrincipalCredential -AppPrincipalId
$p.AppPrincipalId -Type asymmetric -Usage Verify -Value
$credValue -StartDate $cer.GetEffectiveDateString() -
EndDate $cer.GetExpirationDateString()
}
Else
{
    Write-Error "Cannot find certificate."
}
```

The expected result should be as follows.

```
Please enter the administrator user name and password of
the Office 365 tenant domain...
Adding a key to Service Principal...
Complete.
```

Enable push notifications proxying

After OAuth authentication has been successfully set up following the preceding steps, an on-premises admin must enable push notification proxying by using the following script. Make sure to update the value of *$tenantDomain* to be the name of your domain. To do this, copy and paste the following code.

```
$tenantDomain = "Fabrikam.com"
Enable-PushNotificationProxy -Organization:$tenantDomain
```

The expected result should be similar to the following output.

```
RunspaceId          : 4f2eb5cc-b696-482f-92bb-5b254cd19d60
DisplayName         : On Premises Proxy app
Enabled             : True
Organization        : fabrikam.com
Uri                 : https://outlook.office365.com/
PushNotifications
Identity            : OnPrem-Proxy
IsValid             : True
ExchangeVersion     : 0.20 (15.0.0.0)
Name                : OnPrem-Proxy
DistinguishedName   : CN=OnPrem-Proxy,CN=Push Notifications
Settings,CN=First Organization,CN=Microsoft
                      Exchange,CN=Services,CN=Configuration,D
C=Domain,DC=extest,DC=microsoft,DC=com
Guid                : 8b567958-58a4-403c-a8f0-524d7f1e9279
ObjectCategory      : fabrikam.com/Configuration/Schema/ms-
Exch-Push-Notifications-App
ObjectClass         : {top, msExchPushNotificationsApp}
WhenChanged         : 8/27/2013 7:23:47 PM
WhenCreated         : 8/14/2013 1:30:27 PM
WhenChangedUTC      : 8/28/2013 2:23:47 AM
WhenCreatedUTC      : 8/14/2013 8:30:27 PM
OrganizationId      :
OriginatingServer   : server.fabrikam.com
ObjectState         : Unchanged
```

Verify that push notifications are working

After the preceding steps have been completed, push notifications can be tested by one of the following:

- **Sending a test email message to the user's mailbox:**
  1. Set up an account in OWA for Devices on a mobile device to subscribe for notifications.
  2. Return to the device home screen, which puts OWA for Devices in the background.
  3. Send an email message from another device, such as a PC, that goes to the inbox of the account set up on the mobile device.
  4. This should result in an unseen count being indicated on the app icon within a few minutes.
- **Enabling monitoring.** An alternate method to test push notifications, or to investigate why notifications are failing, is to enable monitoring on a mailbox server in your organization. An on-premises Exchange 2013 server admin must invoke push notification proxy monitoring by using the following script. To do this, copy and paste the following code.

```
# Send a push notification to verify connectivity.
$s = Get-ExchangeServer | ?{$_.ServerRole -match "Mailbox"}
If ($s.Count -gt 1)
{
    $s = $s[0]
}
If ($s.Count -ne 0)
{
    # Restart the monitoring service to clear the cache
from when push was previously disabled.
    Restart-Service MSExchangeHM
    # Give the monitoring service enough time to load.
    Start-Sleep -Seconds:120
    Invoke-MonitoringProbe PushNotifications.Proxy
\PushNotificationsEnterpriseConnectivityProbe -
Server:$s.Fqdn | fl ResultType, Error, Exception
}
Else
{
    Write-Error "Cannot find a Mailbox server in the
current site."
}
```

The expected result should be similar to the following output.

```
ResultType : Succeeded
Error      :
Exception  :
```

Using AD FS claims-based authentication with Outlook Web App and EAC

Topic Last Modified: *2014-09-02*

For on-premises Exchange 2013 Service Pack 1 (SP1) deployments, installing and configuring Active Directory Federation Services (AD FS) means you can now use AD FS claims-based authentication to connect to Outlook Web App and EAC. You can integrate AD FS and claims-based authentication with Exchange 2013 SP1, but you can't when you have a deployment that includes Exchange 2007, Exchange 2010, or even Exchange 2013 RTM servers. Using claims-based authentication replaces traditional authentication methods, including the following:

- Windows authentication
- Forms authentication
- Digest authentication
- Basic authentication
- Active Directory client certificate authentication

Authentication is the process of confirming the identity of a user. Authentication validates that the user is who he or she claims to be. Claims-based identity is another approach to authentication. Claims-based authentication removes the management of authentication from the application—in this case, Outlook Web App and EAC—to make it easier to manage accounts by centralizing authentication. Outlook Web App and EAC aren't responsible for authenticating users, storing user accounts and passwords, looking up user identity details, or integrating with other identity systems. Centralizing authentication helps make it easier to upgrade to authentication methods in the future.

Note:
OWA for Devices doesn't support AD FS claims-based authentication.

Exchange 2013 SP1 can be installed on servers running Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2. There are multiple versions of AD FS that can be used, as summarized by the following table.

Windows Serverversion	Installation	AD FS version
Windows Server 2008 R2	**Download and install** AD FS 2.0, which is an add-on Windows component.	AD FS 2.0

Windows Server 2012	Install the **built-in** AD FS server role.	AD FS 2.1
Windows Server 2012 R2	Install the **built-in** AD FS server role.	AD FS 3.0

The tasks that you will perform here are based on Windows Server 2012 R2 that includes the AD FS role service.

Overview of the required steps

Step 1 - Review the certificate requirements for AD FS

Step 2 - Install and configure Active Directory Federation Services (AD FS)

Step 3 - Create a relying party trust for Outlook Web App and EAC

Step 4 - Add AD FS claim rules for Outlook Web App and EAC

Step 5 - Install the Web Application Proxy role service

Step 6 - Configure the Web Application Proxy role service

Step 7 - Publish Outlook Web App and EAC using Web Application Proxy

Step 8 - Configure Exchange 2013 to use AD FS authentication

Step 9 - Enable AD FS authentication on the OWA and ECP virtual directories

Step 10 - Restart or recycle Internet Information Services (IIS)

Additional information you might want to know

What do I need to know before I begin?

- At a minimum, you need to install separate Windows Server 2012 R2 servers: one as a domain controller that uses Active Directory Domain Services (AD DS), an Exchange 2013 server, a Web Application Proxy server, and an Active Directory Federation Services (AD FS) server. Verify that all updates are installed.
- Install AD DS on the appropriate number of Windows Server 2012 R2 servers in your organization. You can also use **Notifications** in **Server Manager** > **Dashboard** to **Promote this server to a domain controller**.
- Install the appropriate number of Client Access and Mailbox servers for your organization. Verify that all updates are installed, including SP1, on all Exchange 2013 servers in your organization. To download SP1, see Updates for Exchange 2013.
- Deploying Web Application Proxy on a server requires local administrator permissions. You must deploy AD FS on a server running Windows Server 2012 R2 in your organization before you can deploy Web Application Proxy.
- Install and configure the AD FS role and create relying party trusts and claim rules on Windows Server 2012 R2. To do this, you need to log on with a user account that is a member of the

Domain Admins, Enterprise Admins, or local Administrators group.

- Determine the required permissions for Exchange 2013 by seeing Feature permissions.
- You need to be assigned permissions for managing Outlook Web App. To see what permissions you need, see the "Outlook Web App permissions" entry in the Clients and mobile devices permissions topic.
- You need to be assigned permissions for managing EAC. To see what permissions you need, see the "Exchange Administration Center connectivity" entry in the Exchange and Shell infrastructure permissions topic.
- You might be able to use only the Shell to perform some procedures. To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Step 1 – Review the certificate requirements for AD FS

Certificates play a critical role in securing communications between Exchange 2013 SP1 servers; web clients such as Outlook Web App; and EAC, Windows Server 2012 R2 servers, including Active Directory Federation Services (AD FS) servers and Web Application Proxy servers. The requirements for certificates vary depending on whether you are setting up an AD FS server, AD FS Proxy, or Web Application Proxy server. The certificates that are used for AD FS services including the SSL and token signing certificates must be imported into the Trust Root Certification Authorities store on all of your Exchange, AD FS and Web Application Proxy servers. The thumbprint for the certificate that is imported is also used on the Exchange 2013 SP1 servers when you use the Set-OrganizationConfig cmdlet.

In any AD FS design, various certificates must be used to secure communication between users on the Internet and AD FS servers. Each federation server must have a service communication certificate or Secure Socket Layer (SSL) certificate and a token-signing certificate before AD FS servers, Active Directory domain controllers, and Exchange 2013 servers can communicate and authenticate. Depending on your security and budget requirements, carefully consider which of your certificates will be obtained by a public CA or an Enterprise CA. If you want to install and configure an Enterprise Root or Subordinate CA, you can use Active Directory Certificate Services (AD CS). If you want to know more about AD CS, see Active Directory Certificate Services Overview.

Although AD FS doesn't require certificates be issued by a CA, the SSL certificate (the SSL certificate that is also used by default as the service communications certificate) must be trusted by the AD FS clients. We recommend that you don't use self-signed certificates. Federation servers use an SSL certificate to secure web services traffic for SSL communication with web clients and with federation server proxies. Because the SSL certificate must be trusted by client computers, we recommend that you use a certificate that is signed by a trusted CA. All certificates that you select

must have a corresponding private key. After you receive a certificate from a CA (Enterprise or public), make sure that all certificates are imported into the Trust Root Certification Authorities store on all servers. You can import certificates into the store with the **Certificates** MMC snap-in or distribute the certificates by using Active Directory Certificate Services. It's important that if the certificate that you imported expires, you manually import another valid certificate.

Important:

If you use the self-signed token signing certificate from AD FS, you must import this certificate into the Trust Root Certification Authorities store on all of your Exchange 2013 servers. If the self-signed token signing certificate isn't used and Web Application Proxy is deployed, then you must update the public key in the Web Application Proxy configuration and all AD FS relying party trusts.

When you are setting up Exchange 2013 SP1, AD FS, and Web Application Proxy, follow these certificate recommendations:

- **Mailbox servers** The certificates that are used on the Mailbox servers are self-signed certificates are they are created when Exchange 2013 is installed. Because all clients connect to an Exchange 2013 Mailbox server through an Exchange 2013 Client Access server, the only certificates that you need to manage are those on the Client Access servers.
- **Client Access servers** An SSL certificate used for service communications is required. If your existing SSL certificate already includes the FQDN you are using to set up the relying party trust endpoint, no additional certificates are required.
- **AD FS** Two types of certificates are required by AD FS:
  - SSL certificate used for service communications
    - Subject name: **adfs.contoso.com** (AD FS deployment name)
    - Subject Alternative Name (SAN): None
  - Token signing certificate
    - Subject name: **tokensigning.contoso.com**
    - Subject Alternative Name (SAN): None

Note:

When you are replacing the token signing certificate on AD FS any existing relying party trusts must be updated to use the new token-signing certificate.

- **Web Application Proxy**
  - SSL certificate used for service communications
    - Subject name: **owa.contoso.com**
    - Subject Alternative Name (SAN): None

Note:

If your Web Application Proxy External URL is the same as your internal URL, you can reuse Exchange's SSL certificate here.

  - AD FS Proxy SSL certificate
    - Subject name: **adfs.contoso.com** (AD FS deployment name)
    - Subject Alternative Name (SAN): None
  - Token signing certificate - This will be copied over from AD FS automatically as part of the

steps below. If this certificate is used, it must be trusted by the Exchange 2013 servers in your organization.

See the certificate requirements section in Review the requirements for deploying AD FS for more information about certificates.

Note:
An SSL encryption certificate is still needed for Outlook Web App and EAC even if you have an SSL certificate for AD FS. The SSL certificate is used on the OWA and ECP virtual directories.

Step 2 – Install and configure Active Directory Federation Services (AD FS)

AD FS in Windows Server 2012 R2 provides simplified, secured identity federation and web single sign-on (SSO) capabilities. AD FS includes a federation service that enables browser-based web SSO, multifactor, and claims-based authentication. AD FS simplifies access to systems and applications by using a claims-based authentication and access authorization mechanism to maintain application security.

To install AD FS on Windows Server 2012 R2:

1. Open **Server Manager** on the **Start** screen or **Server Manager** on the taskbar on the desktop. Click **Add Roles and Features** on the **Manage** menu.
2. On the **Before You Begin** page, click **Next**.
3. On the **Select Installation Type** page, click **Role-based or Feature-based installation**, and then click **Next**.
4. On the **Select Destination Server** page, click **Select a server from the server pool**, verify that the local computer is selected, and then click **Next**.
5. On the **Select Server Roles** page, click **Active Directory Federation Services**, and then click **Next**.

On the **Select Features** page, click **Next**. The required prerequisites or features are already selected for you. You do not need to select any other features.

6. On the **Active Directory Federation Service (AD FS)** page, click **Next**.
7. On the **Confirm Installation Selections** page, check **Restart the destination server automatically if required**, and then click **Install**.

Note:
Do not close the wizard during the installation process.

After you install the required AD FS servers and generate the required certificates, you must configure AD FS and then test that AD FS is working correctly. You can also use the checklist here to help you in setting up and configuring AD FS: Checklist: Setting Up a Federation Server.

To configure Active Directory Federation Services:

1. On the **Installation Progress** page, in the window under **Active Directory Federation Services**, click **Configure the federation service on this server**. The Active Directory Federation Service

Configuration Wizard opens.

2. On the **Welcome** page, click **Create the first federation server in a federation server farm**, and then click **Next**.
3. On the **Connect to AD DS** page, specify an account with domain administrator rights for the correct Active Directory domain that this computer is joined to, and then click **Next**. If you need to select a different user, click **Change**.
4. On the **Specify Service Properties** page, do the following, and then click **Next**:
   - Import the SSL certificate that you obtained earlier from AD CS or a public CA. This certificate is the required service authentication certificate. Browse to the location of your SSL certificate. For details on creating and importing SSL certificates, see Server Certificates.
   - Enter a name for your federation service—for example, type **adfs.contoso.com**.
   - To provide a display name for your federation service, type the name of your organization—for example, **Contoso, Ltd.**.
5. On the **Specify Service Account** page, select **Use an existing domain user account or group Managed Service Account**, and then specify the GMSA account (FsGmsa) that you created when you created the domain controller. Include the account password, and then click **Next**.

Note:

Globally Managed Service Account (GMSA) is an account that must be created when you configure a domain controller. The GMSA account is required during the AD FS installation and configuration. If you haven't created this account yet, run the following Windows PowerShell command. It creates the account for the contoso.com domain and the AD FS server:

```
Add-KdsRootKey -EffectiveTime (Get-Date).AddHours(-10)

New-ADServiceAccount FsGmsa -DNSHostName adfs.contoso.com -
ServicePrincipalNames http/adfs.contoso.com
```

6. On the **Specify Configuration Database** page, select **Create a database on this server using Windows Internal Database**, and then click **Next**.
7. On the **Review Options** page, verify your configuration selections. You can optionally use the **View Script** button to automate additional AD FS installations. Click **Next**.
8. On the **Prerequisite Checks** page, verify that all the prerequisite checks were successfully completed, and then click **Configure**.
9. On the **Installation progress** page, verify that everything installed correctly, and then click **Close**.
10. On the **Results** page, review the results, check whether the configuration completed successfully, and then click **Next steps required for completing your federation service deployment**.

The following Windows PowerShell cmdlet does the same thing as the preceding steps.

```
Import-Module ADFS
Install-AdfsFarm `-
CertificateThumbprint:"0E0C205D252002D535F6D32026B6AB074FB8
40E7"` -FederationServiceDisplayName:"Contoso Corporation"
```

```
`-FederationServiceName:"adfs.contoso.com" `-
GroupServiceAccountIdentifier:"contoso\FSgmsa$"
```

For details and syntax, see Install-AdfsFarm.

To verify the installation: On the AD FS server, open your web browser, and then browse to the URL of the federation metadata—for example, **https://adfs.contoso.com/federationmetadata/2007-06/federationmetadata.xml**.

Step 3 - Create a relying party trust for Outlook Web App and EAC

For all applications and services that you want to publish through Web Application Proxy, you must configure a relying party on the AD FS server. For deployments with multiple Active Directory sites that use separate namespaces, a relying party trust for Outlook Web App and EAC must be added for each namespace.

Note:
EAC uses the ECP virtual directory. You can view or configure settings for EAC by using the Get-EcpVirtualDirectory and the Set-EcpVirtualDirectory cmdlets. To access EAC, you must use a web browser and go to **http://server1.contoso.com/ecp**.

For Outlook Web App, to create relying party trusts by using the AD FS Management snap-in in Windows Server 2012 R2:

1. In **Server Manager**, click **Tools**, and then select **AD FS Management**.
2. In **AD FS snap-in**, under **AD FS\Trust Relationships**, right-click **Relying Party Trusts**, and then click **Add Relying Party Trust** to open the **Add Relying Party Trust** wizard.
3. On the **Welcome** page, click **Start**.
4. On the **Select Data Source** page, click **Enter data about the relying party manually**, and then click **Next**.
5. On the **Specify Display Name** page, in the **Display Name** box, type **Outlook Web App**, and then under **Notes**, type a description for this relying party trust (such as **This is a trust for https://mail.contoso.com/owa**) and then click **Next**.
6. On the **Choose Profile** page, click **AD FS profile**, and then click **Next**.
7. On the **Configure Certificate** page, click **Next**.
8. On the **Configure URL** page, click **Enable support for the WS-Federation Passive protocol**, and then under **Relying party WS-Federation Passive protocol URL**, **type https://mail.contoso.com/owa**, and then click **Next**.
9. On the **Configure Identifiers** page, specify one or more identifiers for this relying party, click **Add** to add them to the list, and then click **Next**.
10. On the **Configure Multi-factor Authentication Now?** page, select **Configure multi-factor authentication settings for this relying party trust**.
11. On the **Configure Multi-factor Authentication** page, verify that **I do not want to configure**

multi-factor authentication settings for this relying party trust at this time is selected, and then click **Next**.

12. On the **Choose Issuance Authorization Rules** page, select **Permit all users to access this relying party**, and then click **Next**.
13. On the **Ready to Add Trust** page, review the settings, and then click **Next** to save your relying party trust information.
14. On the **Finish** page, verify that **Open the Edit Claim Rules dialog for this relying party trust when the wizard closes** isn't selected, and then click **Close**.

To create a relying party trust for EAC, you must do these steps again and create a second relying party trust, but instead of putting in **Outlook Web App** for the display name, enter **EAC**. For the description, enter **This is a trust for the Exchange Admin Center**, and the **Relying party WS-Federation Passive protocol URL** is **https://mail.contoso.com/ecp**.

If you are creating the relying party trusts and claim rules for Outlook Web App and EAC, you need to do the following by using Windows PowerShell:

1. Create the two .txt files IssuanceAuthorizationRules.txt and IssuanceTransformRules.txt.
2. Import their content into two variables.
3. Run the following two cmdlets to create the relying party trusts. In this example, this will also configure the claim rules.

IssuanceAuthorizationRules.txt contains:

```
@RuleTemplate = "AllowAllAuthzRule"
=> issue(Type = "http://schemas.microsoft.com/
authorization/claims/permit",
Value = "true");
```

IssuanceTransformRules.txt contains:

```
@RuleName = "ActiveDirectoryUserSID"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/
identity/claims/windowsaccountname", Issuer == "AD
AUTHORITY"]
=> issue(store = "Active Directory", types = ("http://
schemas.microsoft.com/ws/2008/06/identity/claims/
primarysid"), query = ";objectSID;{0}", param = c.Value);
@RuleName = "ActiveDirectoryGroupSID"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/
identity/claims/windowsaccountname", Issuer == "AD
AUTHORITY"]
=> issue(store = "Active Directory", types = ("http://
schemas.microsoft.com/ws/2008/06/identity/claims/
```

```
groupsid"), query = ";tokenGroups(SID);{0}", param =
c.Value);
@RuleName = "ActiveDirectoryUPN"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/
identity/claims/windowsaccountname", Issuer == "AD
AUTHORITY"]
=> issue(store = "Active Directory", types = ("http://
schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"), query
= ";userPrincipalName;{0}", param = c.Value);
```

Run the following cmdlets:

```
[string]$IssuanceAuthorizationRules=Get-Content -Path C:
\IssuanceAuthorizationRules.txt
[string]$IssuanceTransformRules=Get-Content -Path c:
\IssuanceTransformRules.txt
Add-ADFSRelyingPartyTrust -Name "Outlook Web App" -Enabled
$true -Notes "This is a trust for https://mail.contoso.com/
owa" -WSFedEndpoint https://mail.contoso.com/owa -
Identifier https://mail.contoso.com/owa -
IssuanceTransformRules $IssuanceTransformRules -
IssuanceAuthorizationRules $IssuanceAuthorizationRules
Add-ADFSRelyingPartyTrust -Name "Exchange Admin Center
(EAC)" -Enabled $true -Notes "This is a trust for https://
mail.contoso.com/ecp" -WSFedEndpoint https://
mail.contoso.com/ecp -Identifier https://mail.contoso.com/
ecp -IssuanceTransformRules $IssuanceTransformRules -
IssuanceAuthorizationRules $IssuanceAuthorizationRules
```

Step 4 – Add AD FS claim rules for Outlook Web App and EAC

In a claims-based identity model, the function of Active Directory Federation Services (AD FS) as a federation service is to issue a token that contains a set of claims. Claims rules govern the decisions in regard to claims that AD FS issues. Claim rules and all server configuration data are stored in the AD FS configuration database.

It's required that you create three claim rules:

- Active Directory user SID
- Active Directory group SID

- Active Directory UPN

To add the required claims:

1. In **Server Manager**, click **Tools**, and then click **AD FS Management**.
2. In the console tree, under **AD FS\Trust Relationships**, click either **Claims Provider Trusts** or **Relying Party Trusts**, and then click the relying party trust for Outlook Web App.
3. In the **Relying Party Trusts** window, right-click the Outlook Web App trust, and then click **Edit Claim Rules**.
4. In the **Edit Claim Rules** window, on the **Issuance Transform Rules** tab, click **Add Rule** to start the Add Transform Claim Rule Wizard.
5. On the **Select Rule Template** page, under **Claim rule template**, select **Send Claims Using a Custom Rule** in the list, and then click **Next**.
6. On the **Configure Rule** page, in the **Choose Rule Type** step, under **Claim rule name**, enter the name for the claim rule. Use a descriptive name for the claim rule—for example, **ActiveDirectoryUserSID**. Under **Custom rule**, enter the following claim rule language syntax for this rule:

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/
identity/claims/windowsaccountname", Issuer == "AD
AUTHORITY"]
=> issue(store = "Active Directory", types = ("http://
schemas.microsoft.com/ws/2008/06/identity/claims/
primarysid"), query = ";objectSID;{0}", param = c.Value);
```

7. On the **Configure Rule** page, click **Finish**.
8. In the **Edit Claim Rules** window, on the **Issuance Transform Rules** tab, click **Add Rule** to start the Add Transform Claim Rule Wizard.
9. On the **Select Rule Template** page, under **Claim rule template**, select **Send Claims Using a Custom Rule** in the list, and then click **Next**.
10. On the **Configure Rule** page, on the **Choose Rule Type** step, under **Claim rule name**, enter the name for the claim rule. Use a descriptive name for the claim rule—for example, **ActiveDirectoryGroupSID**. Under **Custom rule**, enter the following claim rule language syntax for this rule:

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/
identity/claims/windowsaccountname", Issuer == "AD
AUTHORITY"]
=> issue(store = "Active Directory", types = ("http://
schemas.microsoft.com/ws/2008/06/identity/claims/
groupsid"), query = ";tokenGroups(SID);{0}", param =
c.Value);
```

11. On the **Configure Rule** page, click **Finish**.
12. In the **Edit Claim Rules** window, on the **Issuance Transform Rules** tab, click **Add Rule** to start

the Add Transform Claim Rule Wizard.

13. On the **Select Rule Template** page, under **Claim rule template**, select **Send Claims Using a Custom Rule** in the list, and then click **Next**.
14. On the **Configure Rule** page, on the **Choose Rule Type** step, under **Claim rule name**, enter the name for the claim rule. Use a descriptive name for the claim rule—for example, **ActiveDirectoryUPN**. Under **Custom rule**, enter the following claim rule language syntax for this rule:

```
c:[Type == "http://schemas.microsoft.com/ws/2008/06/
identity/claims/windowsaccountname", Issuer == "AD
AUTHORITY"]
=> issue(store = "Active Directory", types = ("http://
schemas.xmlsoap.org/ws/2005/05/identity/claims/upn"), query
= ";userPrincipalName;{0}", param = c.Value);
```

15. Click **Finish**.
16. In the **Edit Claim Rules** window, click **Apply**, and then **OK**.

To test the AD FS claims for Outlook Web App:

1. In your web browser, sign in to Outlook Web App—for example, **https://mail.contoso.com/owa**
2. In the browser window, if you get a certificate error, just continue on to the Outlook Web App website.
3. In the browser window, type your user name (domain\user) and password, and then click **Sign in**.
4. Outlook Web App should load in the window.

To test the AD FS claims for EAC:

1. In your web browser, go to **https://mail.contoso.com/ecp**.
2. In the browser window, if you get a certificate error, just continue on to the ECP website.
3. In the browser window, type your user name (domain\user) and password, and then click **Sign in**.
4. EAC should load in the window.

Step 5 – Install the Web Application Proxy role service

Web Application Proxy is a new Remote Access role service in Windows Server 2012 R2. Web Application Proxy provides reverse proxy functionality for web applications inside your corporate network to allow users on many devices to access them from outside the corporate network. Web Application Proxy preauthenticates access to web applications by using Active Directory Federation Services (AD FS) and also functions as an AD FS proxy. Although Web Application Proxy isn't required, it is recommended when AD FS is accessible to external clients. However, offline access in Outlook Web App isn't supported when using AD FS authentication through Web Application Proxy. You can find more information about integrating with Web Application Proxy by seeing Installing and Configuring Web Application Proxy for Publishing Internal Applications

⚠Warning:
You can't install Web Application Proxy on the same server with AD FS installed.

To deploy Web Application Proxy, you must install the Remote Access server role with the Web Application Proxy role service on a server that will act as the Web Application Proxy server. To install the Web Application Proxy role service:

1. On the Web Application Proxy server, in **Server Manager**, click **Manage**, and then click **Add Roles and Features**.
2. In the Add Roles and Features Wizard, click **Next** three times to get to the **Server Roles** page.
3. On the **Server Roles** page, select **Remote Access** in the list, and then click **Next**.
4. On the **Features** page, click **Next**.
5. On the **Remote Access** page, read the information, and then click **Next**.
6. On the **Role Services** page, select **Web Application Proxy**. Then in the **Add Roles and Features Wizard** window, click **Add Features**, and then click **Next**.
7. In the **Confirmation** window, click **Install**. You can also check **Restart the destination server automatically if required**.
8. In the **Installation progress** dialog box, verify that the installation was successful, and then click **Close**.

The following Windows PowerShell cmdlet does the same thing as the preceding steps.

```
Install-WindowsFeature Web-Application-Proxy -
IncludeManagementTools
```

Step 6 – Configure the Web Application Proxy role service

You must configure Web Application Proxy to connect to the AD FS server. Repeat this procedure for all of the servers that you want to deploy as Web Application Proxy servers.

To configure the Web Application role service:

1. On the Web Application Proxy server, in **Server Manager**, click **Tools**, and then click **Remote Access Management**.
2. In the **Configuration** pane, click **Web Application Proxy**.
3. In the **Remote Access Management** console, in the middle pane, click **Run the Web Application Proxy Configuration Wizard**.
4. In the Web Application Proxy Configuration Wizard, in the **Welcome** dialog box, click **Next**.
5. On the **Federation Server** page, do the following, and then click **Next**:
   - In the **Federation service name** box, enter the fully qualified domain name (FQDN) of the AD FS server—for example, **adfs.contoso.com**.
   - In the **User name** and **Password** boxes, type the credentials of a local administrator account on the AD FS servers.
6. In the **AD FS Proxy Certificate** dialog box, in the list of certificates currently installed on the Web Application Proxy server, select the certificate to be used by Web Application Proxy for AD FS proxy functionality, and then click **Next**. The certificate you choose here should be the one whose

subject is the Federation Service name—for example, **adfs.contoso.com**.

7. In the **Confirmation** dialog box, review the settings. If required, you can copy the Windows PowerShell cmdlet to automate additional installations. Click **Configure**.
8. In the **Results** dialog box, verify that the configuration was successful, and then click **Close**.

The following Windows PowerShell cmdlet does the same thing as the preceding steps.

```
Install-WebApplicationProxy -CertificateThumprint
'1a2b3c4d5e6f1a2b3c4d5e6f1a2b3c4d5e6f1a2b' -
FederationServiceName adfs.contoso.com
```

Step 7 – Publish Outlook Web App and EAC by using Web Application Proxy

In step 4, you created claims for Outlook Web App and EAC, and you now need to publish both of these applications. But before you do this, verify that a relying party trust for them was created, and verify that you have a certificate on the Web Application Proxy server that is suitable for Outlook Web App and EAC. For all the AD FS endpoints that you require to be published by Web Application Proxy, in the AD FS Management console, you must set the endpoint to be **Proxy Enabled**.

Follow the steps to publish Outlook Web App by using Web Application Proxy. For EAC, you repeat these steps. When you publish EAC, you need to change the name, external URL, external certificate, and back-end URL.

To publish Outlook Web App and EAC by using Web Application Proxy:

1. On the Web Application Proxy server, in the **Remote Access Management** console, in the **Navigation** pane, click **Web Application Proxy**, and then in the **Tasks** pane, click **Publish**.
2. In the Publish New Application Wizard, on the **Welcome** page, click **Next**.
3. On the **Preauthentication** page, click **Active Directory Federation Services (AD FS)**, and then click **Next**.
4. On the **Relying Party** page, in the list of relying parties, select the relying party for the application that you want to publish, and then click **Next**.
5. On the **Publishing Settings** page, do the following, and then click **Next**:
   a. In the **Name** box, enter a friendly name for the application. This name is used only in the list of published applications in the **Remote Access Management console**. You can use **OWA** and **EAC** for the names.
   b. In the **External URL** box, enter the external URL for this application—for example, **https://external.contoso.com/owa** for Outlook Web App and **https://external.contoso.com/ecp** for EAC.
   c. In the **External certificate** list, select a certificate whose subject name matches the host name of the external URL.
   d. In the **Backend server URL** box, enter the URL of the back-end server. Note that this value is

automatically entered when you enter the external URL, and you should change it only if the back-end server URL is different—for example, **https://mail.contoso.com/owa** for Outlook Web App and **https://mail.contoso.com/ecp** for EAC.

Note:
Web Application Proxy can translate host names in URLs but cannot translate paths. Therefore, you can enter different host names, but you must enter the same path. For example, you can enter an external URL of *https://external.contoso.com/app1/* and a back-end server URL of *https://mail.contoso.com/app1/*. However, you cannot enter an external URL of *https://external.contoso.com/app1/* and a back-end server URL of *https://mail.contoso.com/internal-app1/*.

6. On the **Confirmation** page, review the settings, and then click **Publish**. You can copy the Windows PowerShell command to set up additional published applications.
7. On the **Results** page, make sure that the application published successfully, and then click **Close**.

The following Windows PowerShell cmdlet performs the same tasks as the preceding procedure for Outlook Web App.

```
Add-WebApplicationProxyApplication -BackendServerUrl
'https://mail.contoso.com/owa/' -
ExternalCertificateThumbprint
'E9D5F6CDEA243E6E62090B96EC6DE873AF821983' -ExternalUrl
'https://external.contoso.com/owa/' -Name 'OWA' -
ExternalPreAuthentication ADFS -ADFSRelyingPartyName
'Outlook Web App'
```

The following Windows PowerShell cmdlet performs the same tasks as the preceding procedure for EAC.

```
Add-WebApplicationProxyApplication -BackendServerUrl
'https://mail.contoso.com/ecp/' -
ExternalCertificateThumbprint
'E9D5F6CDEA243E6E62090B96EC6DE873AF821983' -ExternalUrl
'https://external.contoso.com/ecp/' -Name 'EAC' -
ExternalPreAuthentication ADFS -ADFSRelyingPartyName
'Exchange Admin Center'
```

Step 8 – Configure Exchange 2013 to use AD FS authentication

When you are configuring AD FS to be used for claims-based authentication with Outlook Web App and EAC in Exchange 2013, you must enable AD FS for your Exchange organization. You must

use the Set-OrganizationConfig cmdlet to configure AD FS settings for your organization:

- Set the AD FS issuer to **https://adfs.contoso.com/adfs/ls**.
- Set the AD FS URIs to **https://mail.contoso.com/owa** and **https://mail.contoso.com/ecp**.
- Find the AD FS token signing certificate thumbprint by using Windows PowerShell on the AD FS server and entering `Get-ADFSCertificate -CertificateType "Token-signing"`. Then, assign the token-signing certificate thumbprint that you found. If the AD FS token-signing certificate has expired, the thumbprint from the new AD FS token-signing certificate must be updated by using the Set-OrganizationConfig cmdlet.

Using the Exchange Management Shell, enter the following code.

```
$uris = @(" https://mail.contoso.com/owa","https://
mail.contoso.com/ecp")
Set-OrganizationConfig -AdfsIssuer "https://
adfs.contoso.com/adfs/ls/" -AdfsAudienceUris $uris -
AdfsSignCertificateThumbprints"88970C64278A15D642934DC2961D
9CCA5E28DA6B"
```

Note:

The *-AdfsEncryptCertificateThumbprint* parameter isn't supported for these scenarios.

For details and syntax, see Set-OrganizationConfig and Get-ADFSCertificate.

Step 9 – Enable AD FS authentication on the OWA and ECP virtual directories

For the OWA and ECP virtual directories, enable AD FS authentication as the only authentication method and disable all other forms of authentication.

Caution:

You must configure the ECP virtual directory before you configure the OWA virtual directory.

Configure the ECP virtual directory by using the Exchange Management Shell. In the Shell window, enter the following code.

```
Get-EcpVirtualDirectory | Set-EcpVirtualDirectory -
AdfsAuthentication $true -BasicAuthentication $false -
DigestAuthentication $false -FormsAuthentication $false -
WindowsAuthentication $false
```

Configure the OWA virtual directory by using the Exchange Management Shell. In the Shell window, enter the following code.

```
Get-OwaVirtualDirectory | Set-OwaVirtualDirectory -
AdfsAuthentication $true -BasicAuthentication $false -
DigestAuthentication $false -FormsAuthentication $false -
WindowsAuthentication $false OAuthAuthentication $false
```

Note:

The preceding Exchange Management Shell commands configure the OWA and ECP virtual directories on every Client Access server in your organization. If you don't want to apply these settings to all Client Access servers, use the *–Identity* parameter and specify the Client Access server. It's likely you will want to apply these settings only to the Client Access servers in your organization that are Internet facing.

For details and syntax, see Get-OwaVirtualDirectory and Set-OwaVirtualDirectory or Get-EcpVirtualDirectory and Set-EcpVirtualDirectory.

Step 10 – Restart or recycle Internet Information Services (IIS)

After you have completed all of the required steps, including making changes to Exchange virtual directories, you need to restart Internet Information Services. To do this, you can use one of the following methods:

- Using Windows PowerShell:

```
Restart-Service W3SVC,WAS -noforce
```

- Using a command line: Click **Start**, click **Run**, type `IISReset /noforce`, and then click **OK**.
- Using Internet Information Servers (IIS) Manager: In **Server Manager** > **IIS**, click **Tools**, and then click **Internet Information Services (IIS) Manager**. In the **Internet Information Servers (IIS) Manager** window, in the action pane under **Manage Server**, click **Restart**.

Additional information you might want to know

Multifactor authentication

For on-premises Exchange 2013 SP1 deployments, deploying and configuring Active Directory Federation Services (AD FS) 2.0 by using claims means that Outlook Web App and EAC in Exchange 2013 SP1 can support multifactor authentication methods, such as certificate-based authentication, authentication or security tokens, and fingerprint authentication. Two-factor authentication is often confused with other forms of authentication. Multifactor authentication requires the use of two of the three authentication factors. These factors are:

- Something only the user knows (for example, the password, PIN, or pattern)
- Something only the user has (for example, an ATM card, security token, smart card, or mobile phone)

- Something only the user is (for example, a biometric characteristic, such as a fingerprint)

For details on multifactor authentication in Windows Server 2012 R2, see Overview: Manage Risk with Additional Multi-Factor Authentication for Sensitive Applications and Walkthrough Guide: Manage Risk with Additional Multi-Factor Authentication for Sensitive Applications.

In the Windows Server 2012 R2 AD FS role service, the federation service functions as a security token service, provides the security tokens that are used with claims, and gives you the ability to support multifactor authentication. The federation service issues tokens based on the credentials that are presented. After the account store verifies a user's credentials, the claims for the user are generated according to the rules of the trust policy and then added to a security token that is issued to the client. For more information about claims, see Understanding Claims.

MailTips

Exchange Server 2013 > Clients and mobile >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-02-03*

MailTips are informative messages displayed to users while they're composing a message. Microsoft Exchange Server 2013 analyzes the message, including the list of recipients to which it's addressed, and if it detects a potential problem, it notifies the user with MailTips prior to sending the message. With the help of the information provided by MailTips, senders can adjust the message they're composing to avoid undesirable situations or non-delivery reports (NDRs).

How MailTips work

MailTips are implemented as a web service in Exchange 2013. When a sender is composing a message, the client software makes an Exchange web service call to the Client Access server to get the list of MailTips. The server responds with the list of MailTips that apply to that message, and the client software displays the MailTips to the sender.

The following unproductive messaging scenarios are common in any messaging environment:

- NDRs resulting from messages that violate restrictions configured in an organization such as message size restrictions or maximum number of recipients per message.
- NDRs resulting from messages sent to recipients that don't exist, recipients that are restricted, or users whose mailboxes are full.
- Sending messages to users with Automatic Replies configured.

All of these scenarios involve the user sending a message, expecting it to be delivered, and instead receiving a response stating that the message isn't delivered. Even in the best-case scenario, like the automatic reply, these events result in lost productivity. In the case of an NDR, this scenario could

result in a costly call to the Help desk.

There are also several scenarios where sending a message won't result in an error, but can have undesirable, even embarrassing consequences:

- Messages sent to extremely large distribution groups.
- Messages sent to inappropriate distribution groups.
- Messages inadvertently sent to recipients outside your organization.
- Selecting **Reply to All** to a message that was received as a Bcc recipient.

All of these problematic scenarios can be mitigated by informing users of the possible outcome of sending the message as they're composing the message. For example, if senders know that the size of the message they're trying to send exceeds the corporate policy, they won't attempt to send the message. Similarly, if senders are notified that the message they're sending will be delivered to people outside the organization, they're more likely to ensure that the content and the tone of the message are appropriate.

The following messaging clients support MailTips:

- Outlook Web App
- Microsoft Outlook 2010 or later

MailTips in Exchange

The following table lists the available MailTips in Exchange 2013.

MailTip	Availability	Scenario
Invalid Internal Recipient	Outlook	The Invalid Internal Recipient MailTip is displayed if the sender adds a recipient that appears to be internal to the organization but doesn't exist. This could happen if the sender addresses a message to a user who is no longer with the company but whose address resolves due to name resolution cache or an entry in the sender's Contacts folder. It can also happen if the sender types an SMTP address with a domain for which Exchange is authoritative

		and the address doesn't resolve to an existing recipient. The MailTip indicates the invalid recipient and gives the sender the option to remove the recipient from the message.
Mailbox Full	Outlook Outlook Web App	The Mailbox Full MailTip is displayed if the sender adds a recipient whose mailbox is full and your organization has implemented a Prohibit Receive restriction for mailboxes over a specified size. The MailTip indicates the recipient whose mailbox is full and gives the sender the option to remove the recipient from the message. The MailTip is accurate at the time of display. If the message isn't immediately sent, the MailTip is updated every two hours. This also applies to messages that were saved in the Drafts folder and reopened after two hours.
Automatic Replies	Outlook Outlook Web App	The Automatic Replies MailTip is displayed if the sender adds a recipient who has turned on Automatic Replies. The MailTip indicates the recipient has Automatic Replies turned on and also displays the first 250

		characters of the automatic reply configured by the recipient. The MailTip is accurate at the time of display. If the message isn't immediately sent, the MailTip is updated every two hours. This also applies to messages that were saved in the Drafts folder and reopened after two hours. If part of your user mailboxes are hosted on Exchange Online and you're in a coexistence with Exchange Online scenario, the setting on the remote domain object that represents the remote part of your organization has a direct effect on how this MailTip is processed. In Exchange 2013, users can configure different Automatic Replies for internal and external senders. If the remote domain is configured as an internal domain (by setting the *IsInternal* parameter on the remote domain object to `$true`), the internal automatic reply is returned to all users in the organization regardless of where their mailbox resides. However, if the remote domain isn't configured as an internal domain, the internal

		automatic reply is returned to all users whose mailboxes are in the local domain and the external automatic reply is returned to users whose mailboxes are in the remote domain.
Custom	Outlook Outlook Web App	A custom MailTip is displayed if the sender adds a recipient for whom a customized MailTip is configured. A custom MailTip can be useful for providing specific information about a recipient. For example, you can create a custom MailTip for a distribution group explaining its purpose to reduce its misuse. For more information, see Configure custom MailTips for recipients. By default, custom MailTips aren't displayed if the sender isn't allowed to send to that recipient. In that case, the Restricted Recipient MailTip is displayed. However, you can change this configuration and have the custom MailTip also display.
Restricted Recipient	Outlook Outlook Web App	The Restricted Recipient MailTip is displayed if the sender adds a recipient for which delivery restrictions are configured prohibiting this sender from

		sending messages. The MailTip indicates the recipient to which the sender isn't allowed to send messages and gives the sender the option to remove the recipient from the message. It also clearly informs the sender that the message won't be delivered if sent. If the restricted recipient is an external recipient, or if it's a distribution group that contains external recipients, this information is also provided to the sender. However, the following MailTips, if applicable, are suppressed: • Automatic Replies • Mailbox Full • Custom MailTip • Moderated Recipient • Oversize Message
External Recipients	Outlook Outlook Web App	The External Recipients MailTip is displayed if the sender adds a recipient that's external, or adds a distribution group that contains external recipients. This MailTip informs senders if a message they're composing will leave the organization, helping them make the correct decisions about wording, tone, and content.

		By default, this MailTip is turned off. You can turn it on using the **Set-OrganizationConfig** cmdlet. For details, see MailTips over organization relationships. If part of your user mailboxes are hosted on Exchange Online and you're in coexistence with an Exchange Online scenario, the setting on the remote domain object that represents the remote part of your organization has a direct effect on how this MailTip is processed. If the remote domain is configured as an internal domain (by setting the *IsInternal* parameter on the remote domain object to `$true`), any recipients in this remote domain will be treated as internal and therefore the External Recipients MailTip won't be displayed. However, if the remote domain isn't configured as an internal domain, the recipients in that domain will be considered external and this MailTip will be displayed when a message is being composed to those recipients. **Note:** This MailTip isn't evaluated when composing a message to a

		distribution group in the remote domain.
Large Audience	Outlook Outlook Web App	The Large Audience MailTip is displayed if the sender adds a distribution group that has more than the large audience size configured in your organization. By default, Exchange displays this MailTip for messages to distribution groups that have more than 25 members. For details, see Configure the large audience size for your organization. The size of distribution groups isn't calculated each time. Instead, the distribution group information is read from the group metrics data.
Moderated Recipient	Outlook Outlook Web App	The Moderated Recipient MailTip is displayed if the sender adds a recipient that's moderated. The MailTip indicates which recipient is moderated and informs the sender that this may result in delay of the delivery. If the sender is also the moderator, this MailTip isn't displayed. It's also not displayed if the sender has been explicitly allowed to send messages to the recipient (by adding the sender's

		name to the Accept Messages Only From list for the recipient). For instructions on how to configure moderated recipients in Exchange 2013, see Forward a message to a manager for approval. For instructions on how to configure moderated recipients in Exchange Online, see **Configure a moderated recipient in Exchange Online**.
Reply-All on Bcc	Outlook Web App	The Reply-All on Bcc MailTip is displayed if the sender receives a Bcc copy of a message and selects **Reply to All**. When a user selects **Reply to All** to such a message, the fact that the user received a Bcc of that message is revealed to the rest of the audience to which the message was sent. In almost all cases, this is an undesirable situation, and this MailTip informs the user of this condition.
Oversize Message	Outlook	The Oversize Message MailTip is displayed if the message the sender is composing is larger than configured message size limits in your organization. The MailTip is displayed if the

		message size violates one of the following size restrictions: • Maximum send size setting on the sender's mailbox • Maximum receive size setting on the recipient's mailbox • Maximum message size restriction for the organization **Note:** Due to the complexity of the implementation, the message size limits on the connectors in your organization aren't taken into account.

MailTip restrictions

MailTips are subject to the following restrictions:

- MailTips aren't supported when working in offline mode in Outlook.
- When a message is addressed to a distribution group, the MailTips for individual recipients that are members of that distribution group aren't evaluated. However, if any of the members is an external recipient, the External Recipients MailTip is displayed, which shows the sender the number of external recipients in the distribution group.
- If the message is addressed to more than 200 recipients, individual mailbox MailTips aren't evaluated due to performance reasons.
- Custom MailTips are limited to 250 characters.
- If the sender starts composing a message and leaves it open for an extended period of time, the Automatic Replies and Mailbox Full MailTips are evaluated every two hours.

Enable or disable MailTips

Exchange Server 2013 > Clients and mobile > MailTips >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-27*

You can use the Exchange Management Shell to configure various settings that define how you use MailTips in your organization.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "MailTips" entry in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to enable or disable MailTips

You use the **Set-OrganizationConfig** cmdlet to enable or disable MailTips in your organization. MailTips are enabled by default when you install a new Exchange organization. This example shows how to enable MailTips in your organization.

```
Set-OrganizationConfig -MailTipsAllTipsEnabled $true
```

For detailed syntax and parameter information, see Set-OrganizationConfig.

Configure the large audience size for your organization

Exchange Server 2013 > Clients and mobile > MailTips >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-09-27*

You can use the Exchange Management Shell to configure various settings that define how you use MailTips in your organization.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "MailTips" entry in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to configure the large audience size for your organization

You use the **Set-OrganizationConfig** cmdlet to configure the large audience size for your organization. When senders address messages to more recipients than the size you configure, they are shown the Large Audience MailTip. The large audience size is set to 25 by default. This example configures the large audience size to 50 in your organization.

```
Set-OrganizationConfig -MailTipsLargeAudienceThreshold 50
```

For detailed syntax and parameter information, see Set-OrganizationConfig.

Configure custom MailTips for recipients

Exchange Server 2013 > Clients and mobile > MailTips >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-06-01*

MailTips are informative messages displayed to users in the InfoBar in Outlook Web App and Microsoft Outlook 2010 or later when a user does any of the following while composing an e-mail message:

- Add a recipient
- Add an attachment
- Reply or Reply all
- Open a message from the Drafts folder that's already addressed to recipients

In addition to the built-in MailTips that are available, you can create custom MailTips for all types of recipients. For more information about the built-in MailTips, see MailTips.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "MailTips" entry in the Mail flow permissions topic.
- You can configure the primary MailTip in the Exchange admin center (EAC) or in the Shell. However, you can only configure additional MailTip translations in the Shell.
- When you add a MailTip to a recipient, two things happen:
  - HTML tags are automatically added to the text. For example, if you enter the text: `This mailbox is not monitored`, the MailTip automatically becomes: `<html><body>This mailbox is not monitored</body></html>`. Additional HTML tags in the MailTip aren't supported.
  - The text is automatically added to the *MailTipTranslations* property of the recipient as the default value. If you modify the MailTip text, the default value is automatically updated in the *MailTipTranslations* property.
- The length of a MailTip can't exceed 175 displayed characters.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Configure MailTips for recipients

Use the EAC to configure MailTips for recipients

1. In the EAC, navigate to **Recipients**.
2. Select any of the following recipient tabs based on the recipient type:
   - **Mailboxes**
   - **Groups**
   - **Resources**
   - **Contacts**
   - **Shared**
3. On the recipient tab, select the recipient you want to modify, and click **Edit** .
4. In the recipient properties page that appears, click **MailTips**.
5. Enter the text for the MailTip. When you are finished, click **Save**.

Use the Shell to configure MailTips for recipients

To configure a MailTip for a recipient, use the following syntax.

```
Set-<RecipientType> <RecipientIdentity> -MailTip "<MailTip text>"
```

<RecipientType> can be any type of recipient. For example, `Mailbox`, `MailUser`, `MailContact`,

`DistributionGroup`, or `DynamicDistributionGroup`.

For example, suppose you have a mailbox named "Help Desk" for users to submit support requests, and the promised response time is two hours. To configure a custom MailTip that explains this, run the following command:

```
Set-Mailbox "Help Desk" -MailTip "A Help Desk
representative will contact you within 2 hours."
```

Use the Shell to configure additional MailTips in different languages

To configure additional MailTip translations without affecting the existing MailTip text or other existing MailTip translations, use the following syntax:

```
Set-<RecipientType> -MailTipTranslations
@{Add="<culture1>:<localized text
1>","<culture2>:<localized text 2>"...;
Remove="<culture1>:<localized text
1>","<culture2>:<localized text 2>"...}
```

<culture> is a valid ISO 639 two-letter culture code associated with the language.

For example, suppose the mailbox named Notifications currently has the MailTip: "This mailbox is not monitored." To add the Spanish translation, run the following command:

```
Set-Mailbox -MailTipTranslations @{Add="ES:Esta caja no se
supervisa."}
```

How do you know this worked?

To verify that you have successfully configured a MailTip for a recipient, do the following:

1. In Outlook Web App or Outlook 2010 or later, compose an email message addressed to the recipient, but don't send it.
2. Verify the MailTip appears in the InfoBar.
3. If you configured additional MailTip translations, compose the message in Outlook Web App where the language setting matches the language of the MailTip translation to verify the results.

MailTips over organization relationships

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-15*

Microsoft Exchange Server 2013 allows you to configure organization relationships with Microsoft Exchange Online or other Exchange organizations. Establishing an organization relationship allows you to enhance the user experience when dealing with the other organization. For example, you can share free or busy data, configure secure message flow, and enable message tracking across both organizations.

Controlling the MailTips access level

You may want to restrict certain types of MailTips. You can either allow all MailTips to be returned or allow only a limited set that would prevent NDRs. You can configure this setting with the *MailTipsAccessLevel* parameter on the **Set-OrganizationRelationship** cmdlet. The following table shows which MailTips are returned over the organization relationship.

MailTip	Is the MailTip available when the access level is set to All?	Is the MailTip available when the access level is set to Limited?
Large Audience	Yes	No
Automatic Replies	Yes If the remote domain of the recipient is specified as internal, the internal automatic reply is displayed. Otherwise, the external automatic reply is displayed.	Yes The external automatic reply is displayed.
Moderated Recipient	Yes	No
Oversize Message	Yes	Yes
Restricted Recipient	Yes	Yes
Mailbox Full	Yes	No
Custom MailTips	Yes	No
External Recipients	Yes	Yes

	If the remote domain of the recipient is specified as internal, this MailTip is suppressed. Otherwise, the external MailTip is returned.	If the remote domain of the recipient is specified as internal, this MailTip is suppressed. Otherwise, the external MailTip is returned.

For detailed steps about how to configure MailTips access levels, see Manage MailTips for organization relationships.

Controlling the MailTips access scope

When you enable MailTips over an organization relationship and set the access level to `All`, the recipient-specific MailTips, Mailbox Full, Automatic Replies, and custom MailTips, are returned for all users. However, you may only want to allow these MailTips for a specific set of users. For example, if you set up an organization relationship with a partner, you may want to allow these MailTips only for the users that work with that partner.

To achieve this, you need to first create a group and add all users for whom you want to share recipient-specific MailTips to that group. You can then specify that group on the organization relationship.

After you implement this restriction, your Client Access servers will first verify whether the recipient for whom they received a MailTips query is part of this group. If the recipient is a member of this group, the Client Access servers will proxy back all MailTips including the recipient-specific MailTips. Otherwise they won't include the recipient-specific MailTips in their response.

For detailed steps about how to configure MailTips access levels, see Manage MailTips for organization relationships.

Manage MailTips for organization relationships

Clients and mobile > MailTips > MailTips over organization relationships >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-03*

You can use the Exchange Management Shell to configure custom settings for MailTips between various organizations.

By establishing an organizational relationship, you can enhance the user experience for both organizations by sharing free/busy data, configuring secure message flow, and enabling message tracking. For more information about organizational relationships, see MailTips over organization relationships.

You can use various settings to control how MailTips are used between two organizations that have established an organizational relationship. The procedures in this section illustrate these various controls. In all examples, the on-premises organization is contoso.com, the remote organization is online.contoso.com, and the organizational relationship is named Contoso Online.

You use the **Set-OrganizationRelationship** cmdlet to configure these settings.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "MailTips" entry in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable MailTips between two organizations

This example configures the organizational relationship so that MailTips are returned to senders in the remote organization when composing messages to recipients in your organization.

```
Set-OrganizationRelationship "Contoso Online" -
MailTipsAccessEnabled $true
```

This example configures the organizational relationship to prevent MailTips from being returned to senders in the remote organization when composing messages to recipients in your organization.

```
Set-OrganizationRelationship "Contoso Online" -
MailTipsAccessEnabled $false
```

For detailed syntax and parameter information, see Set-OrganizationRelationship.

Use the Shell to configure which MailTips are returned to the remote organization

For each organizational relationship, you can determine which set of MailTips are returned to senders in the other organization. This example configures the organizational relationship so that all MailTips are returned.

```
Set-OrganizationRelationship "Contoso Online" -
MailTipsAccessLevel All
```

This example configures the organizational relationship so that only the Automatic Replies, Oversize Message, Restricted Recipient, and Mailbox Full MailTips are returned.

```
Set-OrganizationRelationship "Contoso Online" -
MailTipsAccessLevel Limited
```

This example configures the organizational relationship so that no MailTips are returned.

Note:

Don't use this method to disable MailTips for this relationship. To disable MailTips, set the *MailTipsAccessEnabled* parameter to `$false`.

```
Set-OrganizationRelationship "Contoso Online" -
MailTipsAccessLevel None
```

For detailed syntax and parameter information, see Set-OrganizationRelationship.

Use the Shell to configure a specific group of users for whom recipient-specific MailTips are returned

You can restrict the return of recipient-specific MailTips to a specific group of users. By default, when you enable MailTips for an organizational relationship, the following recipient-specific MailTips are returned for all users:

- Automatic Replies
- Mailbox Full
- Custom MailTip

You can specify a MailTips access group on the organizational relationship. After you specify a group, the recipient-specific MailTips are returned only for mailboxes, mail contacts, and mail users that are members of that group. This example configures the organizational relationship to return recipient-specific MailTips only for members of the ShareMailTips@contoso.com group.

```
Set-OrganizationRelationship "Contoso Online" -
MailTipsAccessScope ShareMailTips@contoso.com
```

For detailed syntax and parameter information, see Set-OrganizationRelationship.

Group metrics and MailTips

Exchange Server 2013 > Clients and mobile > MailTips >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-16*

Group metrics is the collection of the following data about distribution groups and dynamic distribution groups in your organization:

- Number of members
- Number of members who are external to your organization

Group metrics data is used to support MailTips in Microsoft Exchange Server 2013. MailTips are informative messages that are displayed to senders while they're composing messages. For more information about MailTips,including a full list of MailTips available in Exchange 2013, see MailTips.

Group metrics data is used by the following MailTips:

- **Large Audience** This MailTip is displayed when a sender adds a distribution group whose membership count is considered a large audience as configured in your organization. By default, any message addressed to more than 25 recipients is considered a large audience.
- **External Recipients** This MailTip is displayed when a sender adds a distribution group that has members who are external to your organization.

MailTips are evaluated every time a sender adds a recipient to a message. To provide this information, Exchange calculates group metrics data as a background process that can be scheduled to run outside of your organization's regular business hours. When evaluating recipients for MailTips, Exchange reads group metrics data.

Group Metrics generation

In Exchange 2013, group metrics data is stored in the **msExchGroupMemberCount** and **msExchGroupExternalMemberCount** attributes on the group object in Active Directory. The following files in the %ExchangeInstallPath%GroupMetrics folder are also associated with group metrics:

- **Cookie_<*nnnnnnn*>.dsc** This text file contains information about the Mailbox server that is configured to generate group metrics data, and the last successful group metrics generation time. This allows group metrics to generate data for groups that were changed since the last group

metrics generation.

- **ChangedGroups.txt** This file contains the list of groups that were updated the last time group metrics data was generated.

Group metrics generation is handled by an arbitration mailbox, which is also called an organization mailbox. When the *GMGen* parameter on an arbitration mailbox is set to `$true`, the arbitration mailbox is responsible for generating the group metrics data.

The Mailbox server generates full group metrics data for all distribution groups and dynamic distribution groups the first time the Group Metrics mailbox assistant runs, and incremental updates for any groups that were modified since the last full generation. By default, group metrics data is generated daily at a random time when the Exchange server workload is light. If the workload is constantly high, group metrics generation may be skipped.

To configure group metrics generation, see Configure group metrics.

Configure group metrics

Clients and mobile > MailTips > Group metrics and MailTips >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-30*

MailTips that provide information about the size of distribution groups and dynamic distribution groups rely on group metrics data. Group metrics data is generated on designated Mailbox servers. For more information about group metrics, see Group metrics and MailTips.

You can enable or disable group metrics generation on a Mailbox server.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Group metrics" entry in the Recipients Permissions topic.
- Group metrics data is only used for MailTips. Make sure that group metrics MailTips are enabled in your organization. For detailed steps, see Manage MailTips for organization relationships.
- You can only use the Shell to perform this procedure.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to enable or disable group metrics generation

Note:
By default, group metrics data is generated on any server responsible for generating the offline address book (OAB). These examples are only necessary for organizations that don't use OABs.

To enable or disable group metrics generation on a Mailbox server, run the following command:

```
Set-MailboxServer <ServerIdentity> -
ForceGroupMetricsGeneration <$true | $false>
```

This example enables group metrics generation on a Mailbox server named MBX1.

```
Set-MailboxServer MBX1 -ForceGroupMetricsGeneration $true
```

How do you know this worked?

To verify that you have successfully enabled or disabled group metrics generation in an organization that doesn't use OABs, do the following:
1. Run the following command:

```
Get-MailboxServer <ServerIdentity> | Format-List
ForceGroupMetricsGeneration
```

2. Verify the setting displayed is the setting you configured.

Apps for Outlook

Exchange Server 2013 > Clients and mobile >

Topic Last Modified: *2013-06-17*

Apps for Outlook are applications that extend the usefulness of email by adding information or tools that your users can use without having to leave Outlook Web App. Outlook Web App supports a variety of apps. When apps for Outlook are installed and enabled for a user, the apps are added to messages that meet the criteria that trigger it.

Tip:
For information about apps for Outlook from an end-user perspective, check out the Help topic Installed Apps at Office.com. That topic provides an overview of apps for Outlook and

also shows you some of the apps for Outlook that may be installed by default.

Office Store apps and custom apps

Outlook Web App supports a variety of apps that are available through the Office Store. Outlook Web App also supports custom apps that you can create and distribute to users in your organization.

Note:
Access to the Office Store isn't supported for mailboxes or organizations in specific regions. If you don't see **Add from the Office Store** as an option in the **Exchange admin center** under **Organization** > **Apps** > **+**, you may be able to install an App for Outlook from a URL or file location. For more information, contact your service provider.

Note:
Some apps for Outlook are installed by default. Default apps for Outlook only activate on English language content. For example, German postal addresses in the message body won't activate the Bing Maps app.

App access and installation

You can specify the apps for Outlook that you want your users to be able to use. You can also specify which apps are enabled by default. For more information, see Install or remove apps for Outlook for your organization.

If required, you can limit availability of an app to specific users in your organization. For more information, see Manage user access to apps for Outlook.

Allow administrators and users to install apps

You can specify which administrators in your organization have permission to install and manage apps for Outlook. You can also specify which users in your organization have permission to install and manage apps for their own use. For more information, see Specify the Administrators and Users Who Can Install and Manage Apps for Outlook.

Install or remove apps for Outlook for your organization

Exchange Server 2013 > Clients and mobile > Apps for Outlook >

Topic Last Modified: *2013-06-17*

You can install or remove apps for Outlook for your organization by using the EAC or the Shell.

Note:

By default, after you install an app for your organization, the app is available for all users in your organization. After installation, you can use the EAC or the Shell to make the app optional or required for your users, and to specify whether you want the app to be enabled or disabled. For information about how to change the default settings for an app, see Manage user access to apps for Outlook. To limit availability of apps to specific users in your organization, you must use the Shell. For more information, see Manage user access to apps for Outlook.

For additional management tasks related to apps for Outlook, see Apps for Outlook.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Apps for Outlook" entry in the Recipients Permissions topic.
- You can assign administrators permission to install and manage apps for your organization. You can also assign users permission to install and manage apps for their own use. For more information, see Specify the Administrators and Users Who Can Install and Manage Apps for Outlook.
- Access to the Office Store isn't supported for mailboxes or organizations in specific regions. If you don't see **Add from the Office Store** as an option in the **Exchange admin center** under **Organization** > **Apps** > **+**, you may be able to install an App for Outlook from a URL or file location. For more information, contact your service provider.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Add an app for Outlook

Use the EAC to add an app

1. In the EAC, navigate to **Organization** > **Apps**.
2. Click **New +**, and then choose the location that you want to install the app from.
   - **Add from the Office Store**. At the Office Store, select the app you want to install, and then click **Add**. Apps that work with Outlook Web App are listed under **Apps for Office and SharePoint** > **Outlook**.

Note:

Access to the Office Store isn't supported for mailboxes or organizations in specific regions. If you don't see **Add from the Office Store** as an option in the **Exchange admin center** under **Organization** > **Apps** > **+**, you may be able to install an App for Outlook from a URL or file location. For more information, contact your service provider.

- **Add from URL**. In **URL**, enter the full URL for the app manifest file that you want to install.
- **Add from file**. Select **Browse**, and then navigate to the location of the app manifest file that you want to install.

3. Click **Save**.

Use the Shell to add an app

This example shows you how to add an app from a URL.

```
New-App -OrganizationApp -Url <URL location for App
Manifest file>
```

This example shows you how to add an app from a file.

```
New-App -OrganizationApp -FileData <File location for App
Manifest file>
```

Tip:

When you use the Shell to install an app for your organization, you can install the app and configure settings for it at the same time.

For syntax and parameters, see New-App.

Remove an app for Outlook

Use the EAC to remove an app

1. In the EAC, navigate to **Organization** > **Apps**.
2. In the list view, select the app that you want to remove, and then click **Delete** 🗑.

Use the Shell to remove an app

You can use the Shell to remove an app from your organization.

Note:

Run the following command to look up the display names and application IDs for all the apps for Outlook installed for your organization.

```
Get-App -OrganizationApp |FL DisplayName,AppID
```

Run the following command to remove the custom app Finance Test App from the organization.

```
Remove-App -OrganizationApp -Identity <GUID for Finance
Test App>
```

For syntax and parameters, see Remove-App.

How do you know this worked?

To view the apps for Outlook that are installed in your organization, do one the following:

- In the EAC, navigate to **Organization** > **Apps**, and then review the list of installed apps.
- From the Shell, run `Get-App`, and then review the list of installed apps.

Manage user access to apps for Outlook

Exchange Server 2013 > Clients and mobile > Apps for Outlook >

Topic Last Modified: *2013-11-27*

You can use the EAC or the Shell to manage user access to apps for Outlook.

- Using the EAC, you can manage basic app access settings for your users at an organizational level. For example, you can configure whether an app is enabled or disabled for your users. You can also specify whether an app is required or optional for your users.
- With the Shell, you can manage all the settings that you can with the EAC, as well as other settings. For example, you can limit availability to specific users in your organization.

For additional management tasks related to apps for Outlook, see Apps for Outlook.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Apps for Outlook" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Specify whether an app is available, enabled, or disabled

Use the EAC to specify whether an app is available, enabled, or disabled

1. In the EAC, navigate to **Organization** > **Apps**.
2. In the list view, select the app that you want to change settings for, and then click **Edit** .
3. If you don't want your users to use the app, clear the **Make this app available to users in your organization** check box, and then click **Save**.
4. If you want your users to be able to use the app, select **Make this app available to users in your organization**, and then select the option you want.
   - **Optional, enabled by default** Use this setting if you want to allow your users to turn off the app.
   - **Optional, disabled by default** Use this setting if you want to allow your users to turn on the app.
   - **Mandatory, always enabled. Users can't disable this app** Use this setting if you don't want your users to turn off the app.
5. Click **Save**.

Use the Shell to specify whether an app is available, enabled, or disabled

You can use the Shell to specify whether an app is available, enabled, or disabled.

Note:

Run the following command to look up the display names and application IDs for all the apps for Outlook installed for your organization.

```
Get-App -OrganizationApp | Format-List DisplayName,AppID
```

If you want an app to be disabled and hidden from all your users, run the following command.

```
Set-App <App ID> -OrganizationApp -Enabled $false
```

If you want the app to be enabled by default, but you want your users to be able to turn it off, run the following command.

```
Set-App <App ID> -OrganizationApp -Enabled $true -
DefaultStateForUser Enabled
```

If you want the app to be disabled by default, but you want your users to be able to turn it on, run the following command.

```
Set-App <App ID> -OrganizationApp -Enabled $true -
DefaultStateForUser Disabled
```

If you want the app to be required for your users, run the following command.

```
Set-App <App ID> -OrganizationApp -Enabled $true -
DefaultStateForUser AlwaysEnabled
```

For detailed syntax and parameters, see Set-App.

How do you know this worked?

1. In the EAC, navigate to **Organization** > **Apps**.
2. Review the values in the **User Default** and **Provided To** columns.

Or

1. From the Shell, run `Get-App -OrganizationApp | Format-List DisplayName,AppId,Enabled,Default*`.
2. Review the values for **DefaultStateForUser** and **Enabled**.

Limit availability to specific users

Use the Shell to limit availability to specific users

If you want only members of your Marketing team distribution group to be able to use the LinkedIn app, run the following commands.

```
$a = Get-DistributionGroupMember Marketing

Set-App <App ID for the LinkedIn app> -OrganizationApp -
ProvidedTo SpecificUsers -UserList $a -DefaultStateForUser
Enabled}
```

For detailed syntax and parameters, see Set-App.

How do you know this worked?

To verify that you've successfully limited access for specific users, do the following:

1. From the Shell, run `Get-App -OrganizationApp | Format-List DisplayName,AppId,Enabled,Default*,ProvidedTo,UserList`.
2. Review the value for **ProvidedTo**.

Specify the Administrators and Users Who Can Install and Manage Apps for Outlook

Topic Last Modified: *2013-06-17*

You can specify which administrators in your organization have permissions to install and manage apps for Outlook. You can also specify which users in your organization have permission to install and manage apps for their own use.

There are four built-in management roles that you use to assign these permissions to individual users or security groups.

Administrative roles

- **Org Marketplace Apps** Enables an administrator to install and manage apps that are available from the Office Store for their organization.
- **Org Custom Apps** Enables an administrator to install and manage custom apps for their organization.

User roles

- **My Marketplace Apps** Enables a user to install and manage Office Store apps for their own use.
- **My Custom Apps** Enables a user to install and manage custom apps for their own use.

For information about each of these roles, see Org Marketplace Apps role, Org Custom Apps role, My Marketplace Apps role, and My Custom Apps role.

For information about apps for Outlook, see Apps for Outlook.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role assignments" entry in the Role management permissions topic.
- Access to the Office Store isn't supported for mailboxes or organizations in specific regions. If you don't see **Add from the Office Store** as an option in the **Exchange admin center** under **Organization** > **Apps** > **+**, you may be able to install an App for Outlook from a URL or file location. For more information, contact your service provider.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Assign administrators the permissions required to install

and manage apps for your organization

Use the EAC to assign permissions to administrators

You can use the EAC to assign administrators the permissions required to install and manage apps that available from the Office Store for your organization. For detailed information about how to do this, see Manage role groups.

Use the Shell to assign permissions to administrators

This example assigns administrator Tony Smith (TonySmith) the permissions required to install and manage apps that are available from the Office Store for your organization.

```
New-ManagementRoleAssignment -Role "Org Marketplace Apps" -
User TonySmith
```

This example assigns members of the Store App Admins security group (StoreAppAdmins) the permissions required to install and manage Office Store apps for your organization.

```
New-ManagementRoleAssignment -Role "Org Marketplace Apps" -
SecurityGroup StoreAppAdmins
```

This example assigns administrator Tony Smith (TonySmith) the permissions required to install custom apps for your organization.

```
New-ManagementRoleAssignment -Role "Org Custom Apps" -User
TonySmith
```

This example assigns members of the Custom App Admins security group (CustomAppAdmins) the permissions required to install custom apps for your organization.

```
New-ManagementRoleAssignment -Role "Org Custom Apps" -
SecurityGroup CustomAppAdmins
```

For more information about syntax and parameters, see New-ManagementRoleAssignment.

Assign users the permissions required to install and manage apps for their own use

Use the EAC to assign permissions to users

You can use the EAC to assign users the permissions required to view and modify custom apps for their own use. For detailed information about how to do this, see Manage role groups.

Use the Shell to assign permissions to users

This example assigns users Tony Smith (TonySmith) and Adam Barr (AdamBarr) the permissions required to install apps from the Office Store for their own use.

```
New-ManagementRoleAssignment -Role "My Marketplace Apps" -
UserList TonySmith,AdamBarr
```

This example assigns members of the Store App Users security group (Store AppUsers) the permissions required to install apps from the Office Store for their own use.

```
New-ManagementRoleAssignment -Role "My Marketplace Apps" -
SecurityGroup StoreAppUsers
```

This example assigns user Tony Smith (TonySmith) the permissions required to install custom apps for his own use.

```
New-ManagementRoleAssignment -Role "My Custom Apps" -User
TonySmith
```

This example assigns users of the Custom App Users (CustomAppUsers) the permissions required to install custom apps for their own use.

```
New-ManagementRoleAssignment -Role "My Custom Apps" -
SecurityGroup CustomAppUsers
```

For more information about syntax and parameters, see New-ManagementRoleAssignment.

How do you know this worked?

To verify that you've successfully assigned permissions for a user, run a Shell command using the format `Get-ManagementRoleAssignment -Role <Role Name> -GetEffectiveUsers`, where `Role Name` is the role for which you want to verify assigned permissions.

This example shows you how to verify whom you've assigned permissions to install apps from the Office Store for the organization.

1. Run `Get-ManagementRoleAssignment -Role "Org Marketplace Apps" -GetEffectiveUsers`.
2. In the results, review the entries in the **Effective Users** column.

For more information about syntax and parameters, see Get-ManagementRoleAssignment.

Configuring SSL offloading in Exchange

2013

Exchange Server 2013 > Clients and mobile >

Topic Last Modified: *2014-04-04*

The following helps you in configuring SSL offloading for the protocols and related services on Exchange 2013 Client Access servers with Service Pack 1 (SP1) installed. If you have multiple Client Access servers, you must perform the required steps for each protocol or service on every Client Access server with SP1 installed in your on-premises organization. That is not to mention that each Client Access server in your organization must be configured identically. If you are upgrading to newer Cumulative Updates (CUs) or service packs and you want to continue to use SSL offloading, you must perform the following steps again after you have upgraded or applied those updates on your Exchange 2013 Client Access servers.

One of the biggest advantages to SSL offloading is having the ability to more easily manage certificates that are used. Instead of having separate SSL certificates for each Client Access server with SP1 installed, a single SSL certificate is used and imported to all Client Access servers. The certificate used can be an existing or newly created SSL certificate.

Caution:

When you use Internet Information Services (IIS) Manager, the Exchange Management Shell, or a command-line interface to configure SSL offloading, notice that there is a **Default Web Site** and an **Exchange Back End** site. For SSL offloading, only configure the **Default Web Site** and don't make any changes to the **Exchange Back End** site.

Contents

What do I need to know before I begin?

- Install all of the required Client Access and Mailbox servers in your organization.
- Install Service Pack 1 (SP1) on each Client Access and Mailbox server in your organization. To download SP1, see Updates for Exchange 2013.
- Determine the required permissions for Exchange 2013 by seeing Feature permissions.
- To see what permissions you need for Client Access servers, see "Client Access server permissions" in Clients and mobile devices permissions.
- To see what permissions you need for Client Access servers, see "Outlook Web App permissions" in Clients and mobile devices permissions.
- You might be able to use only the Shell to perform some procedures. To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell.
- To use an existing certificate on your Client Access servers and on the device you are terminating the SSL connections with, export the certificate with the private key on a Client Access server and import or install it on the device. For details, see Export-ExchangeCertificate.
- To use a new certificate, you must use EAC or the Shell to create, import, and enable the new certificate. For details, see Exchange 2013 certificate management UI.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Configure SSL offloading for Outlook Web App

To enable SSL offloading for Outlook Web App, you need to remove the SSL requirement on the **owa** virtual directory on the **Default Web Site**:

- **Step 1** You can use Internet Information Services (IIS) Manager or a command line to disable SSL on the **owa** virtual directory:
  - Using Internet Information Services (IIS) Manager, expand **Sites** > **Default Web Site**, and then select the **owa** virtual directory. In the results pane under **IIS**, double-click **SSL Settings**. In the **SSL Settings** results pane, clear the **Require SSL** check box, and then click **Apply** in the **Actions** pane.
  - Using the command line, type the following and then press Enter.

```
appcmd set config "Default Web Site/owa" /section:access /
sslFlags:None /commit:APPHOST
```

- **Step 2** You need to recycle the correct application pool or restart Internet Information Services by using one of the following methods:
  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
appcmd Recycle AppPool MSExchangeOWAAppPool
```

- Using a Windows PowerShell cmdlet, type the following and then press Enter.

```
IIS:\>Restart-WebAppPool MSExchangeOWAAppPool
```

- Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
iisreset /noforce
```

- Using Internet Information Services (IIS) Manager: In Internet Information Services (IIS) Manager, in the **Actions** pane, click **Restart**.

Configure SSL offloading for the Exchange Admin Center (EAC)

To enable SSL offloading for EAC, you need to remove the SSL requirement on the **ecp** virtual directory on the **Default Web Site**:

- **Step 1** You can use Internet Information Services (IIS) Manager or a command line to disable SSL on the **ecp** virtual directory:
  - Using Internet Information Services (IIS) Manager, expand **Sites** > **Default Web Site**, and then select the **ecp** virtual directory. In the results pane under **IIS**, double-click **SSL Settings**. In the **SSL Settings** results pane, clear the **Require SSL** check box, and then click **Apply** in the **Actions** pane.
  - Using the command line, type the following and then press Enter.

```
appcmd set config "Default Web Site/ecp" /section:access /
sslFlags:None /commit:APPHOST
```

- **Step 2** You need to recycle the correct application pool or restart Internet Information Services by using one of the following methods:
  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
appcmd Recycle AppPool MSExchangeECPAppPool
```

  - Using a Windows PowerShell cmdlet, type the following and then press Enter.

```
IIS:\>Restart-WebAppPool MSExchangeECPAppPool
```

  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
iisreset /noforce
```

- Using Internet Information Services (IIS) Manager: In Internet Information Services (IIS) Manager, in the **Actions** pane, click **Restart**.

Configuring SSL offloading for Outlook Anywhere

SSL offloading for Outlook Anywhere is enabled by default. Outlook Anywhere clients can get email from a private or public network. By default, the internal host name or FQDN of the server is used to enable internal Outlook clients to connect. However, if Outlook Anywhere isn't used internally, then you should remove the internal host name. To allow both internal and external access for Outlook clients, you must configure the internal and external host names, set the authentication method for each, and set both the internal and external clients to require SSL. To configure the authentication method for the external clients, you can use EAC or the Exchange Management Shell, but for internal clients, you must use the Shell:

- **Step 1** You can use EAC or the Shell if you haven't added an external host name for Outlook Anywhere:
  - Using EAC, go to **Servers**, select the name of the Client Access server in the list, and then click **Edit**. In the **Exchange Server** window, click **Outlook Anywhere**, and then in the **Specify the external host name (for example, contoso.com) that users will use to connect to your organization** box, enter the external host name. Verify that the **Allow SSL offloading** option is selected, and then click **Save**.
  - Using the Exchange Management Shell, click **Start**, and then on the **Start** menu, click **Exchange Management Shell**. In the window, type the following and then press Enter:

```
Set-OutlookAnywhere -Identity ClientAccessServer1\Rpc* -
Externalhostname ClientAccessServer1.contoso.com -
ExternalClientsRequireSsl:$True -
ExternalClientAuthenticationMethod Basic
```

- **Step 2** By default, SSL offloading is enabled. However, you can use EAC or the Exchange Management Shell if SSL offloading has been disabled and you want to enable it:
  - Using EAC, go to **Servers**, select the name of the Client Access server in the list, and then click **Edit**. In the **Exchange Sever** window, click **Outlook Anywhere**, click the **Allow SSL offloading** option, and then click **Save**.
  - Using the Shell, type the following and then press Enter.

```
Set-OutlookAnywhere -Identity ClientAccessServer1\Rpc* -
SSLOffloading $true
```

- **Step 3** By default, **Require SSL** is not selected on the **Rpc** virtual directory, but if you want to verify that SSL is disabled, you can use Internet Information Services (IIS) Manager.
  - Using Internet Information Services (IIS) Manager, expand **Sites** > **Default Web Site**, and then select the **Rpc** virtual directory. In the results pane under **IIS**, double-click **SSL Settings**. In the **SSL Settings** results pane, verify that the **Require SSL** check box is cleared, and then click **Apply** in the **Actions** pane.

- **Step 4** You need to recycle the correct application pool or restart Internet Information Services by using one of the following methods:
  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
appcmd Recycle AppPool MSExchangeRpcProxyFrontEndAppPool
```

  - Using a Windows PowerShell cmdlet, type the following and then press Enter.

```
IIS:\>Restart-WebAppPool MSExchangeRpcProxyFrontEndAppPool
```

  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
iisreset /noforce
```

  - Using Internet Information Services (IIS) Manager: In Internet Information Services (IIS) Manager, in the **Actions** pane, click **Restart**.

Important:
You must wait for the Service Host process to apply any changes from Active Directory to Internet Information Services (IIS) every 15 minutes even if you restart IIS on a Client Access server.

Configuring SSL offloading for the Offline Address Book (OAB)

To enable SSL offloading for the Offline Address Book (OAB), you need to remove the SSL requirement on the **OAB** virtual directory on the **Default Web Site**:

- **Step 1** You can use Internet Information Services (IIS) Manager or a command line to disable SSL on the **OAB** virtual directory:
  - Using Internet Information Services (IIS) Manager, expand **Sites** > **Default Web Site**, and then select the **OAB** virtual directory. In the results pane under **IIS**, double-click **SSL Settings**. In the **SSL Settings** results pane, clear the **Require SSL** check box, and then click **Apply** in the **Actions** pane.
  - Using the command line, type the following and then press Enter.

```
appcmd set config "Default Web Site/OAB" /section:access /
sslFlags:None /commit:APPHOST
```

- **Step 2** You need to recycle the correct application pool or restart Internet Information Services by using one of the following methods:
  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
appcmd Recycle AppPool MSExchangeOABAppPool
```

- Using a Windows PowerShell cmdlet, type the following and then press Enter.

```
IIS:\>Restart-WebAppPool MSExchangeOABAppPool
```

- Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
iisreset /noforce
```

- Using Internet Information Services (IIS) Manager: In Internet Information Services (IIS) Manager, in the **Actions** pane, click **Restart**.

Configuring SSL offloading for Exchange ActiveSync (EAS)

To enable SSL offloading for Exchange ActiveSync (EAS), you need to remove the SSL requirement on the **Microsoft-Server-ActiveSync** virtual directory on the **Default Web Site**:

- **Step 1** You can use Internet Information Services (IIS) Manager or a command line to disable SSL on the **Microsoft-Server-ActiveSync** virtual directory:
  - Using Internet Information Services (IIS) Manager, expand **Sites** > **Default Web Site**, and then select the **Microsoft-Server-ActiveSync** virtual directory. In the results pane under **IIS**, double-click **SSL Settings**. In the **SSL Settings** results pane, clear the **Require SSL** check box, and then click **Apply** in the **Actions** pane.
  - Using the command line, type the following and then press Enter.

```
appcmd set config "Default Web Site/
MSExchangeSyncAppPool" /section:access /sslFlags:None /
commit:APPHOST
```

- **Step 2** You need to recycle the correct application pool or restart the Internet Information Services by using one of the following methods:
  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
appcmd Recycle AppPool MSExchangeSyncAppPool
```

- Using a Windows PowerShell cmdlet, type the following and then press Enter.

```
IIS:\>Restart-WebAppPool MSExchangeSyncAppPool
```

- Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
iisreset /noforce
```

- Using Internet Information Services (IIS) Manager: In Internet Information Services (IIS) Manager, in the **Actions** pane, click **Restart**.

Configuring SSL offloading for Exchange Web Services (EWS)

To enable SSL offloading for Exchange Web Services (EWS), you need to remove the SSL requirement on the **EWS** virtual directory on the **Default Web Site**:

- **Step 1** You can use Internet Information Services (IIS) Manager or a command line to disable SSL on the **EWS** virtual directory:
  - Using Internet Information Services (IIS) Manager, expand **Sites** > **Default Web Site**, and then select the **EWS** virtual directory. In the results pane under **IIS**, double-click **SSL Settings**. In the **SSL Settings** results pane, clear the **Require SSL** check box, and then click **Apply** in the **Actions** pane.
  - Using the command line, type the following and then press Enter.

```
appcmd set config "Default Web Site/EWS" /section:access /
sslFlags:None /commit:APPHOST
```

- **Step 2** You need to recycle the correct application pool or restart Internet Information Services by using one of the following methods:
  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
appcmd Recycle AppPool MSExchangeServicesAppPool
```

  - Using a Windows PowerShell cmdlet, type the following and then press Enter.

```
IIS:\>Restart-WebAppPool MSExchangeServicesAppPool
```

  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
iisreset /noforce
```

  - Using Internet Information Services (IIS) Manager: In Internet Information Services (IIS) Manager, in the **Actions** pane, click **Restart**.

Configuring SSL offloading for the Autodiscover service

To enable SSL offloading for the Autodiscover service, you need to remove the SSL requirement on the **Autodiscover** virtual directory on the **Default Web Site**:

- **Step 1** You can use Internet Information Services (IIS) Manager or a command line to disable SSL on the **Autodiscover** virtual directory:
  - Using Internet Information Services (IIS) Manager, expand **Sites** > **Default Web Site**, and then select the **Autodiscover** virtual directory. In the results pane under **IIS**, double-click **SSL**

Settings. In the **SSL Settings** results pane, clear the **Require SSL** check box, and then click **Apply** in the **Actions** pane.

- Using the command line, type the following and then press Enter.

```
appcmd set config "Default Web Site/autodiscover" /
section:access /sslFlags:None /commit:APPHOST
```

- **Step 2** You need to recycle the correct application pool or restart Internet Information Services by using one of the following methods:
  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
appcmd Recycle AppPool MSExchangeAutodiscoverAppPool
```

  - Using a Windows PowerShell cmdlet, type the following and then press Enter.

```
IIS:\>Restart-WebAppPool MSExchangeAutodiscoverAppPool
```

  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
iisreset /noforce
```

  - Using Internet Information Services (IIS) Manager: In Internet Information Services (IIS) Manager, in the **Actions** pane, click **Restart**.

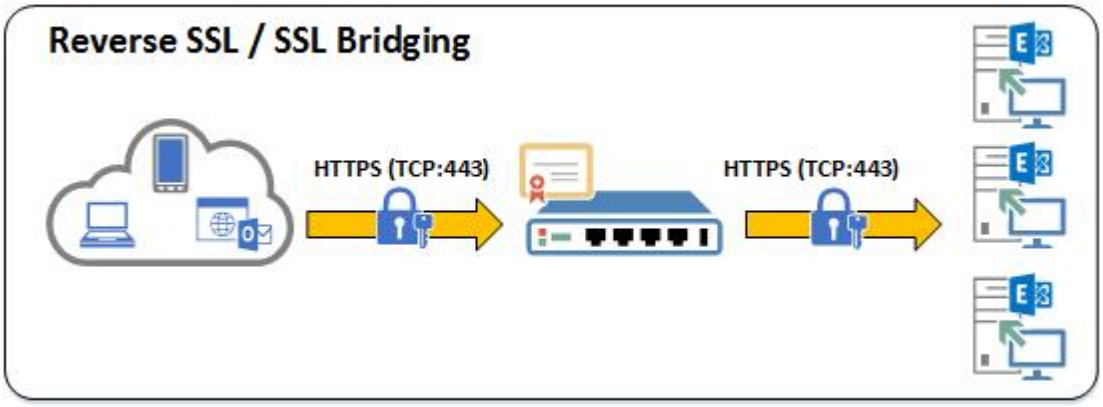
Configuring SSL Offloading for the Mailbox Replication Proxy Service (MRSProxy)

The Mailbox Replication Proxy (MRSProxy) service is installed on every Exchange 2013 Client Access server. MRSProxy helps you to make cross-forest move requests on-premises as well as moving on-premises mailboxes to Office 365. However, by default, MRSProxy is disabled. If you are enabling it, you should enable it in the remote Exchange forest for cross–forest, on-premises mailbox moves or in the on-premises Exchange forest for moving a mailbox to Office 365. Although the MRSProxy service runs under Exchange Web Services (EWS), it's not supported to configure SSL offloading.

The reason for this is that the MRSProxy service expects traffic to be signed/encrypted. Any hardware load balancer or firewall must reencrypt the MRSProxy traffic before sending it to Client Access servers. If this is the case, it is recommended that you configure SSL bridging for offloading to work.

Reverse SSL or SSL Bridging If you enable reverse SSL or SSL bridging on hardware load balancers, you won't need to perform the preceding steps on each CAS server. However, enabling reverse SSL on your hardware load balancers means that SSL encryption and decryption will stay with the Client Access servers. In this case, the SSL encryption and decryption will occur on both the

hardware load balancers and the Client Access servers. Choosing to use Exchange 2013 SSL offloading or reverse SSL (SSL bridging) is dependent on the organizational goals and the security practices that must be implemented. The following picture shows client connectivity with SSL bridging (reverse SSL) enabled.



Configuring SSL offloading for Outlook clients (MAPI virtual directory)

To enable SSL offloading for Outlook clients, you need to remove the SSL requirement on the **MAPI** virtual directory on the **Default Web Site**:

- **Step 1** You can use Internet Information Services (IIS) Manager or a command line to disable SSL on the **MAPI** virtual directory:
  - Using Internet Information Services (IIS) Manager, expand **Sites** > **Default Web Site**, and then select the **MAPI** virtual directory. In the results pane under **IIS**, double-click **SSL Settings**. In the **SSL Settings** results pane, clear the **Require SSL** check box, and then click **Apply** in the **Actions** pane.
  - Using the command line, type the following and then press Enter.

```
appcmd set config "Default Web Site/MAPI" /section:access /
sslFlags:None /commit:APPHOST
```

- **Step 2** You need to recycle the correct application pool or restart the Internet Information Services by using one of the following methods:
  - Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
appcmd Recycle AppPool MSExchangeMapiFrontEndAppPool
```

  - Using a Windows PowerShell cmdlet, type the following and then press Enter.

```
IIS:\>Restart-WebAppPool MSExchangeMapiFrontEndAppPool
```

- Using a command line: Go to **Start** > **Run**, type **cmd**, and then press Enter. In the Command Prompt window, type the following and then press Enter.

```
iisreset /noforce
```

- Using Internet Information Services (IIS) Manager: In Internet Information Services (IIS) Manager, in the **Actions** pane, click **Restart**.

Using a script to enable SSL offloading for all protocols and services

If you're working with a large organization with multiple Exchange 2013 Client Access servers, you might want to speed up the preceding steps that you went through. You can copy and paste the commands in either of the following scripts into Notepad, make any changes, save the file with a .ps1 extension, and then run it from the Exchange Management Shell. Depending on your needs, both of these scripts can be used to configure SSL offloading for all protocols and services for a single Client Access server or for multiple ones.

Note:
For the **Set-OutlookAnywhere** cmdlet entries, replace "MyServer" with the name of your Client Access server(s).

Using Set-WebConfigurationProperty

```
Set-OutlookAnywhere -Identity MyServer\Rpc* -
Externalhostname MyServer.mail.contoso.com -
ExternalClientsRequireSsl $True -
ExternalClientAuthenticationMethod Basic
Set-OutlookAnywhere -Identity MyServer\Rpc* -SSLOffloading
$true
Set-WebConfigurationProperty -Filter //security/access -
name sslflags -Value "None" -PSPath IIS:  -Location
"Default Web Site/OWA"
Set-WebConfigurationProperty -Filter //security/access -
name sslflags -Value "None" -PSPath IIS: -Location "Default
Web Site/ecp"
Set-WebConfigurationProperty -Filter //security/access -
name sslflags -Value "None" -PSPath IIS: -Location "Default
Web Site/EWS"
Set-WebConfigurationProperty -Filter //security/access -
```

```
name sslflags -Value "None" -PSPath IIS: -Location "Default
Web Site/Autodiscover"
Set-WebConfigurationProperty -Filter //security/access -
name sslflags -Value "None" -PSPath IIS: -Location "Default
Web Site/Microsoft-Server-ActiveSync"
Set-WebConfigurationProperty -Filter //security/access -
name sslflags -Value "None" -PSPath IIS: -Location "Default
Web Site/OAB"
Set-WebConfigurationProperty -Filter //security/access -
name sslflags -Value "None" -PSPath IIS: -Location "Default
Web Site/MAPI"
iisreset /noforce
```

Using appcmd

Note:

For the **Set-OutlookAnywhere** cmdlet entries, replace "MyServer" with the name of your Client Access server(s).

```
Set-OutlookAnywhere -Identity MyServer\Rpc* -
Externalhostname MyServer.mail.contoso.com -
ExternalClientsRequireSsl $True -
ExternalClientAuthenticationMethod Basic
Set-OutlookAnywhere -Identity MyServer\Rpc* -SSLOffloading
$true
&$env:systemroot\system32\inetsrv\appcmd set config
"Default Web Site/owa" /section:access /sslFlags:None /
commit:APPHOST
&$env:systemroot\system32\inetsrv\appcmd set config
"Default Web Site/ecp" /section:access /sslFlags:None /
commit:APPHOST
&$env:systemroot\system32\inetsrv\appcmd set config
"Default Web Site/EWS" /section:access /sslFlags:None /
commit:APPHOST
&$env:systemroot\system32\inetsrv\appcmd set config
"Default Web Site/Autodiscover" /section:access /
sslFlags:None /commit:APPHOST
&$env:systemroot\system32\inetsrv\appcmd set config
"Default Web Site/Microsoft-Server-ActiveSync" /
section:access /sslFlags:None /commit:APPHOST
```

```
&$env:systemroot\system32\inetsrv\appcmd set config
"Default Web Site/OAB" /section:access /sslFlags:None /
commit:APPHOST
&$env:systemroot\system32\inetsrv\appcmd set config
"Default Web Site/MAPI" /section:access /sslFlags:None /
commit:APPHOST
iisreset /noforce
```

Configuring coexistence with Exchange 2007 and Exchange 2010

During a coexistence scenario where you have a mix of Exchange 2003 and Exchange 2010 servers in the organization, one of the first steps you need to perform after deploying the Exchange 2010 Client Access Servers is to change DNS so that Exchange 2003 users access their mailboxes from a group of Exchange 2010 Client Access servers. In such a scenario, it's fully supported to enable SSL offloading on the load balancer used to distribute client traffic across the Client Access servers.

Coexistence with other versions of Outlook Web App

With SSL offloading configured on the Exchange 2013 Client Access servers, coexistence works with Exchange 2007 and Exchange 2010:

- To coexist with Exchange 2007, an earlier namespace is required, and redirection will happen to it only for Outlook Web App and Exchange Web Services. Autodiscover, Outlook Anywhere, and Exchange ActiveSync will be proxied over to the earlier versions.
- To coexist with Exchange 2010, if you have the external URL set, a redirect will be used. If not, a proxy will be used.

Unified Messaging

Exchange Server 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-23*

Unified Messaging (UM) enables users to use voice mail and other features, including Outlook Voice Access and Call Answering Rules. UM combines voice messaging and email messaging into one mailbox that can be accessed from many different devices. Users can listen to their messages from their email Inbox or by using Outlook Voice Access from any telephone. You have control over how users place outgoing calls from UM, and the experience people have when they call in to your organization.

Today, IT administrators frequently manage the voice mail or telephony networks and the email systems or data networks for their organizations as separate systems. Voice mail and email are located in separate inboxes that are hosted on separate servers accessed through the desktop for email and through the telephone for voice mail.

Unified Messaging makes it possible for Exchange administrators to combine voice messaging and email messaging into one mailbox so their users can listen to their voice mail messages in their Inbox or by using Outlook Voice Access from any telephone. It uses the Exchange store for both email and voice messages.

Contents

New features

Unified Messaging (UM) was first introduced in Microsoft Exchange Server 2007 and was also available in Exchange 2010. The Unified Messaging feature set in Exchange 2013 is similar to previous versions of Exchange. However, new features have been added and there have been architectural changes. Unified Messaging is now considered a component or sub feature of the voice-related features that are offered in Exchange 2013. The term *Unified Messaging* is still widely used in Exchange Management Shell cmdlets and UM-related services, and all Unified Messaging components—including dial plans, auto attendants, UM mailbox policies, and UM IP gateways—along with the ability to manage those UM components, are located within the Unified Messaging node in the navigation pane of the Exchange Administration Center (EAC).

The following topics are gateways to information about new or enhanced features found in Exchange 2013 Unified Messaging:

- Voice architecture changes
- IPv6 support in Unified Messaging
- Voice mail preview enhancements
- Unified Messaging cmdlet updates

Return to top

Unified Messaging features

The voice mail features found in Unified Messaging offer benefits for both end users and IT administrators.

Features for end users

When you deploy Unified Messaging, users can access voice mail, email, and calendar information that's located in their mailbox from an email client, for example, Outlook or Microsoft Outlook Web App, from a mobile phone with Microsoft Exchange ActiveSync set up, such as a Windows Phone, or from a telephone. Additionally, users can use the following features:

- **Access to Exchange information** UM-enabled users can access a full set of voice mail features from Internet-capable mobile phones, Microsoft Office Outlook 2007 or later versions, and Outlook Web App. These features include many voice mail configuration options and the ability to play a voice message from either the Reading Pane, using an integrated Windows Media Player, or the message list, using computer speakers.
- **Play on Phone** The Play on Phone feature lets UM-enabled users play voice messages over a telephone. If the user works in an office cubicle, is using a public computer or a computer that isn't enabled for multimedia, or is listening to a voice message that's confidential, they might not want to or be able to listen to a voice message through computer speakers. They can play the voice message using any telephone, including a home, office, or mobile telephone.
- **Voice mail form** The voice mail form resembles the default email form. It gives users an interface for performing actions such as playing, stopping, or pausing voice messages, playing voice messages on a telephone, and adding and editing notes.

The voice mail form includes the embedded Windows Media Player and an Audio notes field. The embedded Windows Media Player and notes field are displayed in either the Reading Pane when users preview a voice message or in a separate window when they open the voice message. If users aren't enabled for Unified Messaging, or if a supported email client hasn't been installed on the client computer, they view voice messages as email attachments, and the voice mail form isn't available.

- **User configuration** A user who's enabled for Unified Messaging can configure several voice mail options for Unified Messaging using Outlook Web App. For example, the user can configure telephone access numbers and the voice mail Play on Phone number, and can then reset a voice mail access PIN.
- **Call answering** Call answering includes answering incoming calls on behalf of users, playing their personal greetings, recording messages, and submitting them for delivery to their Inbox as an email message.
- **Call Answering Rules** Call Answering Rules lets users who are enabled for voice mail determine how their incoming call answering calls should be handled. The way call answering rules are applied to incoming calls is similar to the way Inbox rules are applied to incoming email messages. By default, no call answering rules are configured. If an incoming call is answered by the Mailbox server, the caller is prompted to leave a voice message for the called party. Using call answering rules, a caller can:
  - Leave a voice message for the UM-enabled user.
  - Transfer to an alternate contact of the UM-enabled user.
  - Transfer to the alternate contact's voice mail.

- Transfer to other phone numbers that the UM-enabled user has configured.
- Use the Find Me feature or locate the UM-enabled user via a transfer from an operator.

- **Voice Mail Preview** The Mailbox server uses Automatic Speech Recognition (ASR) on newly created voice mail messages. When users receive voice messages, the messages contain both a recording and text that's been created from the voice recording. Users see the voice message text displayed in an email message from within Outlook Web App or another supported email client.
- **Message Waiting Indicator** Message Waiting Indicator is a feature found in most legacy voice mail systems and can refer to any mechanism that indicates the existence of a new message. In Exchange 2007, this functionality was provided by a third-party application, which indicated receipt of a new voice message by lighting the lamp on the desk phone. This feature was added to Exchange 2010, and third-party software is no longer needed. Enabling or disabling Message Waiting Indicator is done on the user's mailbox or on a UM mailbox policy.
- **Missed call and voice mail notifications using SMS** When users are part of a hybrid or Office 365 deployment, and they configure their voice mail settings with their mobile phone number and configure call forwarding, they can receive notifications about missed calls and new voice messages on their cell phones in a text message via the Short Messaging Service (SMS). However, to receive these types of notifications, the users must first configure text messaging and also enable notifications on their account.
- **Protected Voice Mail** Protected Voice Mail is Unified Messaging functionality that enables users to send private mail. This mail is protected by Active Directory Rights Management Services (AD RMS), and users are restricted from forwarding, copying, or extracting the voice file from email. Protected Voice Mail increases the confidentiality of Unified Messaging, and lets users limit the audience for voice messages. This functionality is similar to the way private email messages were handled in Exchange 2007 but now it also applies to voice mail messages.
- **Outlook Voice Access** There are two Unified Messaging user interfaces available to UM-enabled users: the telephone user interface (TUI) and the voice user interface (VUI). These two interfaces together are called Outlook Voice Access. Outlook Voice Access users can use Outlook Voice Access when they access the voice mail system from an external or internal telephone. UM-enabled users who dial in to the voice mail system can access their mailbox using Outlook Voice Access. Using a telephone, a UM-enabled user can:
  - Access voice mail
  - Listen to, forward, or reply to email messages
  - Listen to calendar information
  - Access or dial contacts who are stored in the organization’s directory or a single contact or contact group located in their personal Contacts.
  - Accept or cancel meeting requests
  - Set a voice message to let callers know the called party is away
  - Set user security preferences and personal options
- **Group addressing using Outlook Voice Access** In Exchange 2007, users could use either the telephone user interface (TUI) or voice user interface (VUI) in Outlook Voice Access to send email and voice messages when they signed in to their mailbox. However, users could only send a single email message to a single user in their personal Contacts, to multiple recipients from the

directory by adding each recipient individually, or by adding the name of a distribution list from the directory for your organization. In Exchange 2013, when a user signs in to their mailbox using Outlook Voice Access, they can also send email and voice messages to users in a group stored in their personal Contacts.

Return to top

Administrative features

Currently, most users and IT departments manage their voice mail separately from their email. Voice mail and email exist as separate inboxes hosted on separate servers accessed through the desktop for email and through the telephone for voice mail. Unified Messaging offers an integrated store for all messages and access to content through the computer and the telephone.

Exchange administrators can manage Unified Messaging using the same interface they use to manage the rest of Exchange, using the Exchange Administration Center (EAC) and the Exchange Management Shell. They can:

- Manage voice mail and email from a single platform
- Manage Unified Messaging using scriptable commands
- Build highly available and reliable Unified Messaging infrastructures

Exchange 2013 Unified Messaging offers administrators:

- **A complete voice mail system** Unified Messaging offers a complete voice mail solution using a single store, transport, and directory infrastructure. The store is provided by a Mailbox server and forwarding of incoming calls from a VoIP gateway or IP PBX is handled by a Client Access server. All email and voice mail messages can be managed from a single management point, using a single administration interface and tool set.
- **An Exchange security model** The Microsoft Exchange Unified Messaging service on a Mailbox server and the Microsoft Exchange Unified Messaging Call Router service on a Client Access server run as a single Exchange server account.
- **Consolidation of voice mail systems** Currently, most voice messaging systems require that all the voice messaging components be installed in every physical office location in an organization. In this kind of arrangement, the voice messaging systems in branch offices are located outside the central office and must be administered onsite. This frequently results in increased administration costs and complexity. Unified Messaging lets you manage your voice mail system from a central location. To create a centralized management system for Unified Messaging, you can place some of your Exchange servers in a datacenter or other location, and the remainder of your Exchange servers on-premises and then deploy VoIP gateways, IP PBXs, or Session Border Controllers (SBCs) in each of your branch offices to replace the voice messaging system for each branch office. Deploying a centralized voice messaging system this way can result in a significant savings in hardware and administrative costs.
- **Built-in Unified Messaging administrative roles** The set of UM-specific administrative roles for managing Unified Messaging and voice mail features includes the following:
  - UM Mailboxes

    - UM Prompts
    - Unified Messaging
- **Incoming fax support** Exchange 2013 provides built-in incoming fax support for users who have a UM-enabled mailbox. They can receive fax messages via calls placed to their extension number.

Customers who require a fax solution will have to deploy a fax partner solution. Fax partner solutions are available from several fax partners. The fax partner solutions are designed to be tightly integrated with Exchange and enable UM-enabled users to receive incoming fax messages. You can find a fax partner solution by visiting Microsoft Pinpoint for Fax Partners.

- **Support for multiple languages** All available language packs contain the Text-to-Speech (TTS) engine and the prerecorded prompts for a specified language and ASR support. However, only some language packs contain support for Voice Mail Preview. The US English (en-US) language pack is included on the installation media and additional UM language packs can be downloaded from the Microsoft Download Center.
- **Auto attendant** An auto attendant is a set of voice prompts that gives external and internal users access to the voice mail system. Users can use the telephone keypad or speech inputs to move through the auto attendant menu, place a call to a user, or locate a user in your organization and then place a call to them. An auto attendant gives the administrator the ability to:
    - Create a customized menu for external users.
    - Define informational greetings, business hours greetings, and non-business hours greetings.
    - Define holiday schedules.
    - Describe how to search the organization's directory.
    - Describe how to connect to a user's extension so that external callers can call users by specifying their extension.
    - Describe how to search the organization's directory so that external callers can search the organization's directory and call a specific user.
    - Enable external users to call the operator.

Return to top

Planning and deploying UM

Unified Messaging requires that you integrate your Exchange Server deployment with the existing telephony system for your organization. A successful deployment requires you to make a careful analysis of your existing telephony infrastructure and to perform the correct planning steps to deploy and manage voice mail in Unified Messaging.

When you plan your Unified Messaging deployment, you must consider design and other issues that may affect your ability to reach your organizational goals when you deploy Unified Messaging. Generally, the simpler the Unified Messaging topology, the easier Unified Messaging is to deploy and maintain. Install as few Client Access and Mailbox servers and create as few Unified Messaging components like UM dial plans, auto attendants and UM mailbox policies as you need to support

your business and organizational goals. Large enterprises with complex network and telephony environments, multiple business units, or other complexities will require more planning than smaller organizations with relatively straightforward Unified Messaging needs.

There are many areas that you must consider and evaluate to be able to successfully deploy Unified Messaging. You must understand the different aspects of Unified Messaging and each component and feature so that you can plan your Unified Messaging infrastructure and deployment appropriately. Allocating time to plan and work through these issues will help prevent problems when you deploy Unified Messaging in your organization.The following are some of the areas that you should consider and evaluate when planning for Unified Messaging in your organization:

- The needs of your organization.
- The security requirements in your organization.
- Your existing telephony, circuit-switched network, and your current voice mail system.
- Your current packet-switched IP network design. This includes your local area network (LAN) and WAN connectivity points and devices.
- Your current Active Directory environment.
- The number of users that you'll have to support.
- The number of Client Access and Mailbox servers you'll need.
- Whether you'll be integrating UM with Microsoft Lync Server to enable Enterprise Voice.
- The placement of VoIP gateways, telephony equipment, and Client Access and Mailbox servers.
- The type of UM deployment: on-premises or hybrid.
- The storage requirements for voice mail users.

Return to top

Managing UM with the EAC and the Shell

EAC management

Exchange 2013 provides a single unified management console for your organization that includes all UM components and features. The Exchange Administration Center (EAC) provides a streamlined, optimized interface for management of on-premises, online, or hybrid deployments. The EAC in Exchange 2013 replaces the Exchange Management Console (EMC) and the Exchange Control Panel (ECP) in Exchange 2010. Some of the EAC features include:

- **List view** The list view in EAC has been designed to remove limitations that existed in ECP. ECP was limited to displaying up to 500 objects and, if you wanted to view objects that weren't listed in the details pane, you needed to use searching and filtering to find those specific objects. In Exchange 2013, the viewable limit from within the EAC list view is approximately 20,000 objects. In addition, paging has been added so that you can page to the results. You can also configure page size and export to a CSV file.
- **Add/Remove columns to the Recipient list view** You can choose which columns to view, and you can save your custom list views.
- **Secure the ECP virtual directory** You can partition access from the Internet and Intranets from within the ECP IIS virtual directory to allow or disallow management features. With this feature,

you can permit or deny access to users trying to access the EAC from the Internet outside of your organizational environment, while still allowing access to an end user's Outlook Web App Options.

- **Public Folder management** In Exchange 2010 and Exchange 2007, public folders were managed through the Public Folder administration console. Public folders are now in the EAC, and you don't need a separate tool to manage them.
- **Notifications** In Exchange 2013, the EAC now has a Notification viewer so that you can view the status of long-running processes and, if you choose, receive notification via an email message when the process completes.
- **Role Based Access Control (RBAC) user editor** In Exchange 2010 you could use the RBAC User Editor to add users to management role groups. In Exchange 2013, the RBAC User Editor functionality is now in the EAC, and you don't need a separate tool to manage RBAC.
- **Unified Messaging tools** In Exchange 2010 you could use the Call Statistics and User Call Logs tools to help provide UM statistics and information about specific calls for a UM-enabled user. In Exchange 2013, the Call Statistics and User Call Logs tools are now in the EAC, and you don't need a separate tool to manage them.

Shell management

The Exchange Management Shell, built on Windows PowerShell technology, is a powerful command-line interface that enables automation of administrative tasks. With the Shell, you can manage every aspect of Exchange. You can enable new email accounts, create Send and Receive connectors, configure database properties, manage all aspects of Unified Messaging, and more. The Shell can perform every task that can be performed by the EAC plus tasks that can't be done in the EAC. In fact, when you do something in the EAC, it's the Shell that's doing the work behind the scenes.

Return to top

Unified Messaging documentation

The following table contains links to topics that will help you learn about and manage Exchange Unified Messaging.

Topic	Description
New voice mail features	Learn about new features in Microsoft Exchange 2013.
Planning for Unified Messaging	Learn about the concepts and information you need to plan a Unified Messaging deployment.
Deploying voice mail and UM	Learn about the requirements and steps involved in deploying voice mail and UM.

UM languages, prompts, and greetings	Learn about UM language packs and language settings.
Telephone system integration with UM	Learn about integrating your telephony network with UM.
Connect your voice mail system to your telephone network	Learn how to use and configure UM components to connect your telephony network to Exchange UM.
Automatically answer and route incoming calls	Learn how to create UM auto attendants and manage settings for navigation menus, greetings, and business and non-business hours.
Set up voice mail for users	Learn how to create and manage UM mailbox policies and how to enable users for UM.
Set up client voice mail features	Learn how to set up client features to enable users to access and manage their voice mail messages.
Set Outlook Voice Access PIN security	Learn how to set PIN requirements for Outlook Voice Access users.
Protect voice mail	Learn how to use UM to protect voice messages.
Run reports for voice mail calls	Learn about UM call reports.

New voice mail features

Exchange Server 2013 > Unified Messaging >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-13*

Unified Messaging (UM) in Microsoft Exchange Server 2013 includes the same feature set as Exchange 2010 and Exchange 2007, with some enhancements and architectural changes. However, Unified Messaging is no longer a separate server role. It's now a component of the voice-related features offered in Exchange 2013.

Changes in the Voice architecture

The Voice architecture in Exchange 2013 is different than it was in Exchange 2010 and Exchange 2007. In previous versions of Exchange UM, all the components for Unified Messaging were included on a server that had the UM server role installed. In Exchange 2013, the Unified Messaging components are split between a Client Access server running the Microsoft Exchange Unified Messaging Call Router service and a Mailbox server running the Microsoft Exchange Unified Messaging service. Most of the functionality, including the services and worker processes for Unified Messaging, is located on each Mailbox server. The Client Access server, running the Microsoft Exchange Unified Messaging Call Router service, proxies incoming calls to the Mailbox server. For details, see Voice architecture changes.

Support for IPv6

Internet Protocol version 6 (IPv6) is the most recent version of the Internet Protocol (IP). IPv6 is intended to correct many of the shortcomings of IPv4, which was the previous version of the IP. Just as Exchange 2010 did, Exchange 2013 Client Access and Mailbox servers fully support IPv6 networks. For details, see IPv6 support in Unified Messaging.

Support for UCMA 4.0 API

Since Exchange 2010 Service Pack 1 (SP1), the Unified Messaging role has relied on Unified Communications Managed API v2.0 (UCMA) for signaling and media. Therefore, UCMA 2.0 was a prerequisite for Exchange 2010 UM setup. UCMA 2.0 is downloaded separately and deployed manually by administrators on UM servers running Exchange 2010 SP1 or a later version.

However, UCMA 2.0 has several limitations. Many of these shortcomings are corrected by UCMA 4.0, which is required for Exchange 2013. Now that the UM server is no longer a separate server role, it's the Client Access and Mailbox servers that require UCMA 4.0.

UCMA 4.0 supports new features in Unified Messaging, such as using the same version of the Speech Engine for both Text-to-Speech (TTS) and Automatic Speech Recognition (ASR). The platform that's used for Exchange 2013, .NET 4.0, includes a single installer file and enables backward compatibility with Exchange 2010 and Exchange 2007 UM servers.

In Exchange 2010 SP2 and SP1, UCMA 2.0 installation is required prior to installing the service pack on a Unified Messaging server.

Using UCMA 4.0 offers multiple benefits:

- It incorporates hotfixes and patches.
- It supports IPv6.
- Deployment of UCMA 4.0 has been automated and simplified.
- UCMA 4.0 setup includes all prerequisites for Exchange 2013.
- UCMA 4.0 provides more accurate speech engine translations and more scalable voice platform support across multiple products.

Note:

UCMA 4.0 is installed when you're installing Exchange 2013. For details about UCMA 4.0 and setup requirements, see Exchange 2013 prerequisites. To upgrade to the most recent version of UCMA, you must first uninstall any previous versions of UCMA that are installed using Add/Remove programs.

Improvements to Voice Mail Preview

Some enhancements to speech-related services are included in Exchange Server 2013 UM via the Speech Engine 11.0 and UCMA 4.0. There have been improvements in grammar generation, core voice services, and support for multiple languages. Exchange Server 2013 UM also includes several enhancements for transcription services that are delivered to end users and increased confidence and accuracy for Voice Mail Preview. For details, see Voice mail preview enhancements.

Enhanced caller ID support

In previous releases of Exchange Unified Messaging, a UM server that received a call used caller ID to look up the possible identity of the calling party. This search extended across Active Directory and the UM user's personal contacts stored in their mailbox.

In the past, users sometimes were frustrated by the voice mail system's failure to identify Exchange or personal contacts from their caller ID. Until now, only the default contact folder in a user's Exchange mailbox has been used for this search. But Exchange Server 2013 users are likely to have contacts aggregated from external social networks or contacts they added to unique folders when organizing their contacts. In Exchange 2013, UM extends the scope of the search to include the user's other Exchange and personal contact folders that were created manually. Exchange 2013 also supports contact aggregation from external social networks, provides intelligence to link multiple contacts that refer to the same person, and uses that data to present person-centric (rather than contact-centric) views. This means that contacts that are aggregated from external social networks can be placed in the contact folder stored in the user's mailbox in Microsoft Outlook Web App and Outlook. These contacts can now also be added to any additional contact folders that users create.

Caller ID look-up is integrated with contact aggregation, so that it searches across external contacts. The **PersonID** property, where present and set to a value other than Null, improves the user experience for caller ID resolution by suppressing duplicate matches to contacts that are associated with the same person. Because the PersonID property is the same on both results, UM treats this as a match to a single contact.

Enhancements to speech platform and speech recognition

Exchange Server 2013 UM introduces some enhancements to the speech platform and speech recognition, including the following:

- Enhancements and improved accuracy for Voice Mail Preview.
- Support for the Microsoft Speech Platform – Runtime (Version 11.0).
- Speech grammar generation using the system mailbox for an organization.

Exchange Unified Messaging uses static and dynamic speech grammars to recognize commands, names of contacts in the global address list (GAL), and names of personal contacts in the user's mailbox. Today, in Exchange Server 2013, every Mailbox server running the Microsoft Exchange Unified Messaging service generates grammars for all UM languages installed on it and stores them in directories. Every Mailbox server stores every possible grammar, which it generates based on the number of dial plans, auto attendants, and the UM languages that are installed.

Grammar files are used by UM to allow callers to use speech to locate users in your organization. The files are updated each 24 hours by the Mailbox Assistant. The GGG.exe command in Exchange 2007 and Exchange 2010 made it possible to manually update the grammar files without waiting for the scheduled update. In Exchange Server 2013, to address ASR grammar generation scalability issues for UM, the speech GAL grammar generation no longer happens on the server with the Unified Messaging server role installed. Instead, it happens periodically using the Mailbox Assistant, on the Mailbox server running the Microsoft Exchange Unified Messaging service that hosts the organization's arbitration mailbox. The GAL speech grammar file is stored in the arbitration mailbox for an organization and later downloaded to all Mailbox servers in the Exchange organization. By default, the Mailbox Assistant runs every 24 hours. You can adjust the frequency by using the **Set-MailboxServer** cmdlet.

Cmdlet updates

For Exchange 2013, many UM cmdlets have been brought over from Exchange 2010. However, there have been changes in some of those cmdlets, and new cmdlets have been added for new functionality. For details, see Unified Messaging cmdlet updates.

Voice architecture changes

Exchange Server 2013 > Unified Messaging > New voice mail features >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-04*

The Microsoft Exchange Server 2013 architecture is different than the architecture in Exchange

Server 2007 and Exchange Server 2010. In Exchange 2007 and Exchange 2010, the types of servers were separated into multiple server roles: Client Access, Mailbox, Hub Transport, and Unified Messaging. In Exchange 2013, the server roles are combined into two types of servers, and all components or services from those server roles are run on the same physical server or on two separate servers called Client Access and Mailbox. In the new model, the Client Access server running the Microsoft Exchange Unified Messaging Call Router service redirects Session Initialization Protocol (SIP) traffic that's generated from an incoming call to a Mailbox server. Then a media (Realtime Transport Protocol (RTP) or secure RTP (SRTP)) channel is established from the VoIP gateway or IP Private Branch eXchange (PBX) to the Mailbox server that hosts the user's mailbox. In Exchange 2013, the Mailbox server has the same processes as the Unified Messaging server role in Exchange 2007 and Exchange 2010. The Mailbox server runs both the Microsoft Exchange Unified Messaging service and UM worker processes. The Client Access server runs the Microsoft Exchange Unified Messaging Call Router service, which receives an incoming call and forwards it to the Mailbox server.

Contents

Support for the new Exchange architecture

In Exchange 2013, the Client Access server is responsible for Autodiscover, Secure Sockets Layer (SSL), authentication, redirection, and proxy. The Client Access server is the entry point for any inbound calls or SIP requests for Unified Messaging (UM). The routing logic and SIP REDIRECT is implemented as a service that's automatically included in a Client Access server. This service is known as the Microsoft Exchange Unified Messaging Call Router service. It's installed and runs on each Client Access server in your organization. When a Client Access server receives a SIP INVITE for an incoming call, the Microsoft Exchange Unified Messaging Call Router service redirects the incoming call to the Mailbox server. Then a media channel (RTP or SRTP) is created between the VoIP gateway, IP PBX, or session border controller (SBC) and the Mailbox server. Although the Client Access server acts as a SIP redirector, it only handles SIP requests from VoIP gateways, IP PBXs, or SBCs. It doesn't receive any media traffic. Media traffic that uses RTP or SRTP is only passed between the Mailbox server and SIP peers such as VoIP gateways, IP PBXs, or SBCs—not to the Client Access server. When you deploy Exchange 2013 and UM, you have to configure your VoIP gateways, IP PBXs, or SBCs to point to the Client Access servers that you've installed so that incoming calls will be routed correctly for UM.

In some cases, deploying multiple Client Access servers is a requirement, and the Client Access servers are deployed separately on different physical hardware from the Mailbox servers. Client Access servers can be grouped in an array by using an L4 or L5 hardware or software load balancer.

However, there are no Active Directory Exchange object-based Client Access server arrays. Using a hardware or software load balancer in front of Client Access servers is an accepted practice in larger Exchange deployments.

When a Client Access server is installed, the Microsoft Exchange Unified Messaging Call Router service is running. The service does the following:

- When initialized, it reads a local configuration file named msexchangeumcallrouter.config.
- Performs speech grammar generation using an arbitration mailbox for an organization.
- Supports Transmission Control Protocol (TCP) and/or Transport Layer Security (TLS) connections. This setting is configurable.
- Will only stop if there's a configuration error or if it can't register the required ports.

In Exchange 2013, the Mailbox server isn't responsible for answering SIP requests from incoming calls. It's only responsible for receiving the SIP traffic from a Client Access server and then establishing an RTP or SRTP connection to the VoIP gateway, IP PBX, or SBC.

After a Client Access server redirects an incoming call to a Mailbox server, a media channel is established between the VoIP gateway, IP PBX, or SBC and the Mailbox server. After the media channel is established, the Microsoft Exchange Unified Messaging service on the Mailbox server plays the user's voice mail greeting, processes call answering rules for the user, and invites the caller to leave a voice message. The Mailbox server then records the voice message, creates a transcription of the message, and deposits it in the user's mailbox. However, if you're integrating Exchange with Office Communications Server 2007 R2 or Lync Server, both the SIP and RTP or SRTP media channels for incoming calls are handled by Lync servers and the Mailbox server. In a Lync integrated environment, you don't have VoIP gateways, IP PBXs, or SBCs. To Lync, the Mailbox server that's running the Microsoft Exchange Unified Messaging service looks just like an Exchange 2010 UM server. The Mailbox server and the Client Access server that's running the Microsoft Exchange Unified Messaging Call Router service are considered trusted peers because both servers must be added to a SIP dial plan. Lync routes the incoming call using the Inbound Routing component, which uses SIP to communicate with the Client Access server and then route the call to a Mailbox server.

Exchange 2010 UM administrators can configure a set of properties for Unified Messaging on each UM server. In Exchange 2013, UM components and configuration settings for UM are found on both Client Access and Mailbox servers. All the configuration settings that applied to a single computer running the Unified Messaging server role in Exchange 2010 are still available. However, some of those properties and configuration settings are set on a Client Access server that's running the Microsoft Exchange Unified Messaging Call Router service, and others are available on a Mailbox server that's running the Microsoft Exchange Unified Messaging service. In some cases, the same setting is available on both. The following list shows the cmdlets and parameters that are available on Client Access servers and Mailbox servers and where changes were made to a cmdlet to support deployment scenarios with previous versions of Unified Messaging.

- **Set-UMService -DialPlans <MultiValuedProperty>** Available on Exchange 2013 Mailbox servers and also works on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Set-UMCallRouterSettings -DialPlans <MultiValuedProperty>** Available on Exchange 2013

Client Access servers but not available for Exchange 2007 and Exchange 2010 Unified Messaging servers.

- **Set-UMService -MaxCallsAllowed <Int32>** Available on Exchange 2013 Mailbox servers and also works on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Set-UMCallRouterSettings -MaxCallsAllowed <Int32>** Not available on Exchange 2013 Client Access servers and not available for Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Set-UMService -SipTcpListeningPort <Int32>** Not configurable on Exchange 2013 Mailbox servers but works on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Set-UMService -SipTlsListeningPort <Int32>** Not configurable on Exchange 2013 Mailbox servers but works on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Set-UMCallRouterSettings -SipTcpListeningPort <Int32>** Available on Exchange 2013 Client Access servers but doesn't work on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Set-UMCallRouterSettings -SipTlsListeningPort <Int32>** Available on Exchange 2013 Client Access servers but doesn't work on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Set-UMService - Status <Enabled | Disabled | NoNewCalls>** Not available on Exchange 2013 Mailbox servers but works on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Set-UMCallRouterSettings - Status <Enabled | Disabled | NoNewCalls>** Not available on Exchange 2013 Client Access servers and doesn't work on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Set-UMService -UMStartupMode <TCP | TLS | Dual>** Available on Exchange 2013 Mailbox servers and works on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Set-UMCallRouterSettings - UMStartupMode <TCP | TLS | Dual>** Available on Exchange 2013 Client Access servers but doesn't work on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Enable-UMService** Not available on Exchange 2013 Mailbox servers but works on Exchange 2007 and Exchange 2010 Unified Messaging servers.
- **Disable-UMService** Not available on Exchange 2013 Mailbox servers but works on Exchange 2007 and Exchange 2010 Unified Messaging servers.

For the Mailbox server, you'll use the **Set/Get/Enable/Disable-UMService** cmdlets to view or configure UM properties for the Microsoft Exchange Unified Messaging service on Exchange 2013 Mailbox servers or Exchange 2007 or Exchange 2010 Unified Messaging servers. A different set of cmdlets, **Set/Get-UMCallRouterSettings**, are used to view or configure the Microsoft Exchange Unified Messaging Call Router service properties on a Client Access server. This ensures that the existing **Get-UMServer**, **Set-UMServer**, **Enable-UMServer**, and **Disable-UMServer** cmdlets from Exchange 2007 and Exchange 2010 will work in a coexistence deployment with Exchange 2013 Mailbox servers. This also ensures that the cmdlets will work when the Mailbox and Client Access servers are installed on the same or different servers.

Return to top

UM ports

The Microsoft Exchange Unified Messaging Call Router service found on a Client Access server uses SIP over either Transmission Control Protocol (TCP) or mutual Transport Layer Security (mutual TLS) to communicate with Mailbox servers that are running the Microsoft Exchange Unified Messaging service. To avoid TCP/User Datagram Protocol (UDP) port conflicts, the Microsoft Exchange Unified Messaging Call Router service and the Microsoft Exchange Unified Messaging service default to and listen on different TCP ports. They can accept both unsecured and secured connections, depending on whether mutual TLS is used with SIP and RTP traffic. By default, a Client Access server listens for SIP requests on both TCP port 5060 in Unsecured mode and TCP port 5061 in SIP Secured mode when mutual TLS is used. These ports are configurable using the **Set-UMCallRouterSettings** cmdlet. The Microsoft Exchange Unified Messaging Call Router service on the Client Access server doesn't handle media (RTP or SRTP) traffic, so only TCP ports and no UDP ports are used. By default, a Mailbox server listens for SIP requests on both TCP port 5062 in Unsecured mode and TCP port 5063 in SIP Secured mode when mutual TLS is used. These ports aren't configurable using Exchange Management Shell cmdlets. On the Mailbox server that runs the Microsoft Exchange Unified Messaging service, TCP ports can't be configured on the Exchange server either by using the Shell or by configuring settings in the registry. The Microsoft Exchange Unified Messaging service on the Mailbox server will accept connections from a Client Access server on SIP ports 5062 and 5063. After the Client Access server redirects the SIP request to a Mailbox server, an RTP or SRTP media channel is created using a VoIP gateway, IP PBX, or SBC, and the Microsoft Exchange Unified Messaging worker process on the Mailbox server.

The following table summarizes the Exchange 2013 ports and protocols, and whether the ports can be changed.

UM listening ports

Protocol	TCP port	UDP port	Can the ports be changed?
SIP (Client Access server – Microsoft Exchange Unified Messaging Call Router service)	5060 (unsecured), 5061 (secured). The service listens on both ports.	Not applicable	Yes, using the **Set-UMCallRouterSettings** cmdlet.
SIP (Mailbox server – Microsoft Exchange Unified Messaging service)	5062 (unsecured), 5063 (secured). The service listens on both ports.	Not applicable	Ports can't be changed.
SIP (Mailbox server - UM worker process)	5065 and 5067 for TCP (unsecured). 5066 and	Not applicable	Ports can't be changed.

	5068 for mutual TLS (secured). This is the case when *UMStartupMode* is set to *Dual*. If *UMStartUpMode* is set to *TCP* or *TLS*, ports 5065 and 5066 are used. The default *UMStartupMode* is *TCP*.		
RTP (Mailbox server - UM worker process)	Not applicable	Ports between 1024 and 65535.	The range of ports can be changed through the registry (however, this isn't a supported configuration): HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft Speech Server\2.0\AudioConnectionMinPort HKLM\SOFTWARE\Microsoft\Microsoft Speech Server\2.0\AudioConnectionMaxPort

Return to top

UM dial plans

Mapping or associating UM dial plans to UM servers isn't required in Exchange 2013 the way it was in Exchange 2007 and Exchange 2010. Client Access or Mailbox servers running UM services don't need to be linked to a dial plan because all Client Access and Mailbox servers are expected to

receive all incoming calls from VoIP gateways, IP PBXs, or SBCs. The exception is that SIP dial plans that are used with Lync 2013, Lync Server 2010, and Office Communications Server 2007 R2 must be associated with Client Access and Mailbox servers that you've deployed. Both types of Exchange servers must be added to each SIP dial plan to be included as trusted peers from Communications Server 2007 R2 or Lync Server. Otherwise, Communications Server 2007 R2 or Lync Server will reject outbound calls from users.

The following table summarizes the relationship between Client Access and Mailbox servers and UM dial plans.

Linking UM dial plans

Topology	Dial plan
Client Access and Mailbox on the same server (without Communications Server 2007 R2 or Lync Server 2010 non-SIP dial plans)	Dial plans are no longer required to be associated with a Client Access or Mailbox server. You aren't allowed to add the Client Access or Mailbox servers to a dial plan. If you run the **Set-UMService** cmdlet, it will generate an error if you try to associate a Mailbox server with a non-SIP dial plan.
Client Access and Mailbox on different servers (without Communications Server 2007 R2 or Lync Server 2010 non-SIP dial plans)	Dial plans are no longer required to be associated with Client Access or Mailbox servers. You aren't allowed to add Client Access or Mailbox servers to a dial plan. If you run the **Set-UMService** cmdlet, it will generate an error if you try to associate a Mailbox server with a non-SIP dial plan.
Client Access and Mailbox server on the same physical server (with Communications Server 2007 R2 and Lync Server 2010 with SIP dial plans)	For a single SIP dial plan, add all Client Access and Mailbox servers to the SIP dial plan. For multiple SIP dial plans, add all Client Access and Mailbox servers to each SIP dial plan. This will make both servers trusted peers of Office Communications Server 2007 R2 or Lync Server. You must use the same certificate in your Office Communications Server 2007 R2 or Lync Server deployment as you do on each

	Client Access and Mailbox server.
Client Access and Mailbox server on different physical servers (with Communications Server 2007 R2 and Lync Server 2010 with SIP dial plans)	For a single SIP dial plan, add all Client Access and Mailbox servers to the SIP dial plan. For multiple SIP dial plans, add all Client Access and Mailbox servers to each SIP dial plan. This will make both servers trusted peers of Office Communications Server 2007 R2 or Lync Server. If the certificates being used on the Client Access and Mailbox servers are different, you must use the same certificate in your Office Communications Server 2007 R2 or Lync Server deployment as you do on each Client Access and Mailbox server in your organization.

Return to top

UM Call Router performance counters

Past versions of Exchange included the Unified Messaging server role, which ran the Microsoft Exchange Unified Messaging service. Because of the architecture changes in Exchange 2013, the Client Access server runs the Microsoft Exchange Unified Messaging Call Router service and the Mailbox server runs the Microsoft Exchange Unified Messaging service. The same performance counters for the Microsoft Exchange Unified Messaging service are available to administrators as in earlier versions of Exchange UM. However, there are also additional performance counters that you can use on the Client Access server to verify the status of the Microsoft Exchange Unified Messaging Call Router service and for troubleshooting.

To support the new Client Access Unified Messaging Call Router service in Exchange 2013, the following performance counters are now available.

Performance counters

Performance counter category	Counter name	Description	Threshold
MSExchangeUMRouter Availability	% of Inbound Calls Rejected by the Microsoft Exchange	Shows the percentage of inbound calls that were rejected by the	Should always be less than 5 percent, but should be 0 at all times.

	Unified Messaging Call Router Service over the Last Hour	Microsoft Exchange Unified Messaging Call router service over the last hour.	
MSExchangeUMRouter Availability	Calls Disconnected on Irrecoverable Internal Error for the Microsoft Exchange Unified Messaging Call Router Service	Shows the number of calls disconnected after an internal system error occurred.	Should be 0 at all times.
MSExchangeUMRouter Availability	Total Inbound Calls Rejected by the Microsoft Exchange Unified Messaging Call Router Service	Shows the total number of inbound calls that were rejected by the Microsoft Exchange Unified Messaging Call Router service since the service was started.	Should be 0 at all times.
MSExchangeUMRouter Availability	Total number of calls received by the Microsoft Exchange Unified Messaging Call Router Service	Shows the total number of inbound calls that were received by the Microsoft Exchange Unified Messaging Call Router service since the service was started.	Should be 0 or greater.
MSExchangeUMRouter Availability	% of Inbound Calls Rejected by the Microsoft Exchange Unified Messaging Call Router Service Over	Shows the percentage of inbound calls that were rejected by the Microsoft Exchange Unified Messaging Call	Should be less than 5 percent.

	the Last Hour	Router service over the last hour.	

Return to top

IPv6 support in Unified Messaging

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-12*

Internet Protocol version 6 (IPv6) is the most recent version of the Internet Protocol (IP). IPv6 is intended to correct many of the shortcomings of IPv4, which was the previous version of the IP. In Microsoft Exchange Server 2010, IPv6 is supported only when IPv4 is also used. A pure IPv6 Exchange environment isn't supported. The use of IPv6 addresses and IP address ranges is supported only when both IPv6 and IPv4 are enabled on the computer running Exchange 2010 and the network supports both IP address versions. However, because IPv4 and IPv6 are completely different protocols, an IPv4 network can't communicate directly with an IPv6 network, and vice versa. To handle this shortcoming, network administrators are required to deploy devices, such as routers, that can route information between IPv4 networks and IPv6 networks. If Exchange 2010 is deployed using both IPv4 and IPv6, all server roles except Unified Messaging (UM) can send data to and receive data from devices, servers, and clients that use IPv6 addresses. With Exchange 2013, Unified Messaging is no longer a separate server role like the Transport, Client Access, and Mailbox server roles in Exchange 2007 and Exchange 2010. UM-related components and speech services run on only Client Access and Mailbox servers.

In Exchange 2013, because UM architecture has changed and now requires Unified Communications Managed API (UCMA) v4.0 to support both IPv4 and IPv6 as well as other Exchange features, both Client Access and Mailbox servers that have Unified Messaging components and services will fully support IPv6 networks.

IPv6 support

Starting with Exchange 2010 Service Pack 1 (SP1), the Unified Messaging server role relied on UCMA 2.0 for its underlying Session Initiation Protocol (SIP) signaling and speech processing. UCMA 2.0 is the main component for speech features in UM. UCMA 2.0 contains a SIP stack, a media stack, and speech engines for Automatic Speech Recognition (ASR), in addition to speech synthesis that's generated by Text-to-Speech (TTS).

In Exchange 2010, running a dual stack (IPv4 and IPv6) was required by all server roles except for

UM because UM required UCMA 2.0 but supported only IPv4, not IPv6. For Exchange 2013, UCMA 4.0 is used by UM and is required for installing Exchange 2013 on Client Access and Mailbox servers. UCMA 4.0 is required to support new features and to support IPv6.

Some of the reasons why UM now uses UCMA 4.0 to support the new features in Exchange 2013, including IPv6, are as follows:

- Some government agencies require IPv6 support for products that they use.
- UM now requires compatibility with hardware devices such as routers, IP gateways, IP PBXs, and session border controllers (SBCs) that run either a dual stack (IPv4 and IPv6) or IPv6 only.
- In Exchange 2013, the Microsoft Exchange Unified Messaging service runs on the Mailbox server and the Microsoft Exchange Unified Messaging Call Router service runs on the Client Access server. The Mailbox and Client Access server roles in Exchange 2013 require both IPv4 and IPv6.
- Online services allow clients to connect to their service using either IPv4 orIPv6.
- Public IPv4 address space is running out. For Exchange Server 2013 Enterprise, this isn't really an issue for UM, since UM always communicates with internal SIP peers that can be deployed with a private IPv4 address space. However, for hosted Exchange UM, the customer's equipment must support hosted UM using IPv4 and IPv6.

With the exception of UM and a small part of Transport, Exchange 2013 can connect to Exchange 2010 servers in an organization when either a Client Access or Mailbox server runs in dual-stack mode with IPv4 and IPv6 enabled. This means that customers can install Exchange 2013 on computers that are running with both IPv4 and IPv6 stack addresses configured. This allows IPv6 clients and other Exchange servers, including Exchange Server 2010, to connect directly to Exchange 2013.

UM works on Windows servers running in dual-stack mode. This is because protocols such as HTTP ignore the transport type, and UM uses voice over IP (VoIP) protocols (including SIP/RTP/STUN/TURN/ICE), which aren't dependent on one another. This includes media negotiation (RTP/SRTP), in which UM advertises and communicates a list of IP addresses to SIP peers, such as IP gateways, IP PBXs, or SBCs.

What does it mean to support IPv6 for UM?

To enable Exchange 2013 UM to support IPv6, both enterprise and online UM administrators must be able to take advantage of IPv6 when they connect UM to IPv6-capable devices, including devices such as routers, IP gateways, IP PBXs, and Office Communications Server 2007 R2 and Microsoft Lync servers. However, if IPv6 isn't available for interoperability and backward compatibility with previous versions of Exchange, administrators don't need to make additional configuration changes, and IPv4 can be used instead.

For Exchange 2013 Enterprise, UM must communicate directly with SIP peers (IP gateways, IP PBXs, and SBCs) that may not support IPv6 in their software or firmware. Therefore, UM must be able to communicate directly with SIP peers that support IPv4 and, more important, with IPv6. For hosted Exchange 2013, UM communicates with customer equipment through SBCs or Lync Server 2010 or

Lync Server 15. In hosted Exchange 2013 environments, IPv6 SIP-aware clients such as SBCs and Lync servers can potentially be deployed and thus handle the IPv6-to-IPv4 conversion process.

UM device support for IPv6

Because Exchange 2013 Mailbox and Client Access servers that run UM components and services support IPv6, IP gateway, IP PBX, and SBC vendors must also be able to support IPv6. There are several issues that affect device support for IPv6:

- There are IP gateways, IP PBXs, and SBCs that may be able to support IPv6 but haven't yet been tested with IPv6 and UM. This support may be added in the future, but it's dependent on the hardware vendor.
- Some IP gateways currently have no IPv6 support.
- Some SBCs have IPv4-IPv6 functionality, but they don't currently work for UM because they don't support SRTP (Secure Real-time Transport Protocol)/SDES (Session Description Protocol Security).
- There are IP PBXs that don't support a dual stack and pure IPv6, but these devices haven't been tested to work with Exchange 2013.

Currently UCMA 4.0 is IPv6-enabled, meaning that it can accept IPv6 connections but that IPv4 also can be accepted, when operating in dual mode or when making outbound connections. Running in dual mode allows IPv4 connections to be made when they're needed to connect to previous versions of Exchange UM. For Lync installations, this is done by Lync Server, which obtains the version information from Active Directory for the latest version of Exchange Server. For traditional telephony devices—including IP gateways, IP PBXs, and SBCs—to support IPv6 connections along with IPv4, they must listen for both types of connections. This is because each SIP peer must be able to accept both types of connections for backward compatibility with previous versions of Exchange UM. This is also necessary to support outdialing for both types of connections.

UM configuration for supporting IPv6

After you install your Client Access and Mailbox servers, you need to create Unified Messaging dial plans, auto attendants, IP gateways, and hunt groups. To allow UM to support IPv6, you must:

- Create a new UM IP gateway or configure an existing UM IP gateway with an IPv6 address for each of the IP gateways, IP PBXs, or SBCs on your network. When you're creating and configuring the required UM IP gateways, you must add the IPv6 address or the Fully Qualified Domain Name (FQDN) for the UM IP gateway. If you're adding the FQDN to the UM IP gateway, you must have created the correct DNS records to resolve the UM IP gateway FQDN to the IPv6 address. If you have an existing UM IP gateway, you can use the **Set-UMIPgateway** cmdlet to configure the IPv6 address or FQDN. After you create or configure the UM IP gateways, you can use the **Get-UMIPgateway** cmdlet to view the properties of the UM IP gateway to ensure that the IPv6 settings are correct.
- Configure the *IPAddressFamily* parameter on each UM IP gateway. To enable the IP gateway to accept IPv6 packets, you must set the UM IP gateway to either accept both IPv4 and IPv6

connections, or accept only IPv6 connections, by using the **Set-UMIPgateway** cmdlet and setting the *IPAddressFamily* parameter to one of the following:
  - *IPv4* – This is the default and is used if no other value is configured.
  - *IPv6* - This enables IPv6 to be used. However, IPv4 isn't used.
  - *Any* – This allows IPv6 to be used, but if the device doesn't support IPv6, then IPv4 is used instead.
- After you've configured your UM IP gateways, you must also configure the IP gateways, IP PBXs, and SBCs on your network to support IPv6. For details, see your hardware vendor for a list of devices that support IPv6 and how to correctly configure them.
- Optionally, you may need to set the Client Access and Mailbox servers to accept IPv6 traffic if either of the servers are only set to receive IPv4 traffic. However, the default setting is for both Client Access servers running the Microsoft Exchange Unified Messaging Call Router service and Mailbox servers running the Microsoft Exchange Unified Messaging service to accept IPv4 and IPv6 traffic. For details about configuring the IPv6 settings on Client Access and Mailbox servers, see Set-UMCallRouterSettings and Set-UMService.

There are two parameters that may need to be configured on Client Access and Mailbox servers to support IPv6: *IPAddressFamily* and *IPAddressFamilyConfigurable*. To enable a Client Access and a Mailbox server to accept IPv6 packets, you must set the Client Access and Mailbox server either to accept both IPv4 and IPv6 connections, or accept only IPv6 connections. To configure the *IPAddressFamily* parameter, the *IPAddressFamilyConfigurable* parameter must be set to `$true`.

UM IP addressing logic

The logic behind IPv6 support for UM in Exchange 2013 is as follows:

- Client Access and Mailbox servers listen on both IPv4 and IPv6 interfaces when the dual stack is enabled and the Client Access and Mailbox servers are set to *IPv6* or *Any*. Otherwise, only IPv4 is used.
- For outgoing calls, UM uses dual mode if the *IPAddressFamily* parameter for the UM IP gateways, Client Access servers, and Mailbox servers is set to *IPv6* or *Any*. Otherwise, only IPv4 is used.

When making outgoing calls in dual mode, if the *IPAddressFamily* parameter is set to *IPv6* or *Any*:

- UCMA will obtain a list of addresses in the FQDN for a SIP peer that it's trying to reach.
- UCMA will try all IPv6 addresses, if any.
- If UCMA determines that an address isn't available, it will include the address in a list and not try it again based on a configured interval. This prevents UM from needlessly retrying known bad addresses.
- If no IPv6 addresses are available, UCMA will fall back to IPv4 addresses in the list of addresses for SIP peers.

Voice mail preview enhancements

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-07-05*

Voice Mail Preview is a feature that's available to users who receive their voice mail messages using Microsoft Exchange Server 2010 or Exchange Server 2013 Unified Messaging (UM). Voice Mail Preview enhances UM voice mail functionality by providing a text version of audio recordings. The voice mail text is displayed in an email message within Microsoft Office Outlook Web App, Outlook 2010, and other email programs.

Voice Mail Preview enhancements

In Exchange 2013, UM includes several enhancements to the user interface for Outlook Web App and Outlook clients and improvements to increase confidence and accuracy for Voice Mail Preview. Also, some enhancements to the speech-related services are offered through the Microsoft Speech Platform (Version 11.0) and Unified Communications Managed API (UCMA) 4.0 to enhance grammar generation and language support.

Unified Messaging introduced the Voice Mail Preview feature in Exchange 2010. Voice Mail Preview uses automatic speech recognition (ASR) to add a text version of a voice mail audio file to a voice message. ASR isn't entirely accurate, especially when it's used to record audio over a phone that includes unknown voices and noises.

Some organizations require consistently error-free or near-error-free transcripts of voice mail messages for some, if not all, of their users. The Voice Mail Preview Partner Program helps such organizations meet those requirements. The Voice Mail Preview Partner Program was designed for Exchange 2010 to improve Voice Mail Preview results, but it wasn’t used by Exchange 2010 customers because of the overhead and cost. To address these issues, Exchange 2013 includes the following enhancements for Voice Mail Preview:

- **Improved audio normalization** Audio normalization is the process of uniformly increasing (or decreasing) the amplitude of an entire audio signal so that the resulting peak amplitude matches a specified target or the norm. UM normalizes the audio recording before compressing it and sending to the user.
- **Enhanced speech recognition** By collecting voice mail messages (only if the Exchange customer chooses to share this information), the results of the voice mail previews can be used to add words and phrases to the speech engine. This is done by setting the *VoiceMailAnalysisEnabled* parameter to `$true` using the **Set-UMMailbox** cmdlet or by setting the *AllowVoiceMailAnalysis* parameter to `$true` on the **Set-UMMailboxPolicy** cmdlet. Also, Exchange 2013 UM uses information more efficiently from email threads created by a user using Outlook Voice Access. This includes information about the participant’s (Active Directory or personal contact) information (country, city, company), and the phone number of the Outlook Voice Access user.
- **Voice Mail Preview confidence** The confidence score is the number assigned by Unified Messaging that is directly related to the overall accuracy of the transcription. The confidence calculations that are used by UM have been adjusted to be more accurate and represent the actual accuracy of a transcribed message.

- **Filtering** Offensive words are detected and filtered and the results are cached and stored in the user's mailbox.
- **Hiding the text preview** If the confidence score of a voice mail preview is below a given threshold, the Voice Mail Preview text will be hidden. If the text is hidden, the voice message will include text stating that the confidence of the voice mail was too low for results to be displayed.
- **Transcription performance** Voice Mail Preview is a CPU-intensive operation that requires roughly twice the time it takes to process an audio file. If generating the voice mail preview text takes too long, CPU throttling stops processing the preview. In Exchange 2010, UM didn't try to transcribe any voice message that was longer than 75 seconds. In Exchange 2013, the entire voice message is transcribed, but the text for the message isn't included if it extends past 75 seconds.
- **Color schemes** Because of the confusion over the colors that were used to distinguish between low, medium, and high confidence for a voice mail preview, the color scheme has been removed in Exchange 2013 for Outlook Web App and Outlook.

Unified Messaging cmdlet updates

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-10-03*

Many of the Unified Messaging (UM) cmdlets that existed in Exchange Server 2010 have been brought over for Exchange Server 2013, but there have been changes to some of those cmdlets. In addition, new cmdlets were added for Exchange 2013.

Updated parameters and new UM cmdlets

The following is a list of the updated parameters and new cmdlets for Exchange 2013.

Cmdlet	Parameters
New-UMIPGateway	`[-IPAddressFamily <IPv4Only | IPv6Only | Any>]`
Set-UMIPGateway	`[-IPAddressFamily <IPv4Only | IPv6Only | Any>]`
Get-UMMailbox	`[-AccountPartition <AccountPartitionIdParameter>]`
Set-UMMailbox	`[-ImListMigrationCompleted <$true | $false> -VoiceMailAnalysisEnabled <$true | $false>]`
Test-Connectivity	`[-CallRouter <SwitchParameter>]`
New-UMCallAnsweringRule	`[-Name <String> [-CallerIds <MultiValuedProperty>] [-`

	CallersCanInterruptGreeting <$true | $false>] [-CheckAutomaticReplies <$true | $false>] [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-ExtensionsDialed <MultiValuedProperty>] [-KeyMappings <MultiValuedProperty>] [-Mailbox <MailboxIdParameter>] [-Organization <OrganizationIdParameter>] [-Priority <Int32>] [-ScheduleStatus <Int32>] [-TimeOfDay <TimeOfDay>] [-WhatIf [<SwitchParameter>]]
Remove-UMCallAnsweringRule	[-Identity <UMCallAnsweringRuleIdParameter> [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
Get-UMCallAnsweringRule	[-Identity <UMCallAnsweringRuleIdParameter>] [-DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>]
Set-UMCallAnsweringRule	[-Identity <UMCallAnsweringRuleIdParameter> [-CallerIds <MultiValuedProperty>] [-CallersCanInterruptGreeting <$true | $false>] [-CheckAutomaticReplies <$true | $false>] [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-ExtensionsDialed <MultiValuedProperty>] [-KeyMappings <MultiValuedProperty>] [-Mailbox <MailboxIdParameter>] [-Name <String>] [-Priority <Int32>] [-ScheduleStatus <Int32>] [-TimeOfDay <TimeOfDay>] [-WhatIf [<SwitchParameter>]]
Enable-UMCallAnsweringRule	[-Identity <UMCallAnsweringRuleIdParameter> [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
Disable-UMCallAnsweringRule	[-Identity <UMCallAnsweringRuleIdParameter> [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
Get-UMCallRouterSettings	[-DomainController <Fqdn>] [-Server <ServerIdParameter>]
Set-UMCallRouterSettings	Set-UMCallRouterSettings [-Confirm [<SwitchParameter>]] [-DialPlans <MultiValuedProperty>] [-DomainController <Fqdn>] [-IPAddressFamily <IPv4Only | IPv6Only | Any>] [-IPAddressFamilyConfigurable <$true | $false>] [-Server <ServerIdParameter>] [-SipTcpListeningPort <Int32>] [-SipTlsListeningPort <Int32>] [-UMPodRedirectTemplate <String>] [-UMStartupMode <TCP | TLS | Dual>] [-WhatIf [<SwitchParameter>]]

Disable-UMService	`-Identity <UMServerIdParameter> [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Immediate <$true | $false>] [-WhatIf [<SwitchParameter>]]` **Note:** This cmdlet only works with Exchange 2007 and 2010 UM servers.
Enable-UMService	`-Identity <UMServerIdParameter> [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]` **Note:** This cmdlet only works with Exchange 2007 and 2010 UM servers.
Get-UMService	`[-Identity <UMServerIdParameter>] [-DomainController <Fqdn>]`
Set-UMService	`Set-UMService -Identity <UMServerIdParameter> [-Confirm [<SwitchParameter>]] [-DialPlans <MultiValuedProperty>] [-DomainController <Fqdn>] [-GrammarGenerationSchedule <ScheduleInterval[]>] [-IPAddressFamily <IPv4Only | IPv6Only | Any>] [-IPAddressFamilyConfigurable <$true | $false>] [-IrmLogEnabled <$true | $false>] [-IrmLogMaxAge <EnhancedTimeSpan>] [-IrmLogMaxDirectorySize <Unlimited>] [-IrmLogMaxFileSize <ByteQuantifiedSize>] [-IrmLogPath <LocalLongFullPath>] [-MaxCallsAllowed <Int32>] [-SIPAccessService <ProtocolConnectionSettings>] [-UMStartupMode <TCP | TLS | Dual>] [-WhatIf [<SwitchParameter>]]`

For details about all UM cmdlets, see Unified Messaging cmdlets.

Planning for Unified Messaging

Exchange Server 2013 > Unified Messaging >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-10*

When you plan your Unified Messaging (UM) deployment, there are many factors that you must consider to be able to successfully deploy UM. You must understand the different elements of Unified Messaging and each component and feature so that you can plan your Unified Messaging infrastructure and deployment appropriately. Allocating time to plan and work through these issues will help prevent problems when you deploy Unified Messaging in your organization.

You may be deploying Unified Messaging in a new Exchange organization or upgrading from a legacy or third-party voice mail solution. If you're upgrading, you need to decide whether to convert the devices that are accepting telephony circuit-based protocols to a data network using IP or whether to deploy an Enterprise Voice solution like Microsoft Lync Server. This is just the first step in preparing yourself to understand and deploy UM for your organization.

Planning your voice mail system

UM provides voice mail, fax, and email messaging in one store that can be accessed from a telephone, a user's computer, or a mobile device. Users can access voice messages, email, calendar information, and personal contacts that are located in their Exchange mailbox from email clients such as Outlook and Outlook Web App.

Client Access servers running the Microsoft Exchange Unified Messaging Call Router service and Mailbox servers running the Microsoft Exchange Unified Messaging service are designed to provide voice mail features for users in your organization.

Mailbox servers rely on Client Access servers to forward SIP traffic from incoming calls and then establish a connection with a VoIP gateway, IP PBX, or Session Border Controller (SBC) and accept the RTP/SRTP media traffic. All voice mail and fax messages are submitted from the Microsoft Exchange Unified Messaging service on a Mailbox server to be delivered to the user's mailbox. For a user to use the voice mail features with Unified Messaging, they must have an Exchange mailbox.

Planning your UM deployment

Generally, the simpler the Unified Messaging topology, the easier UM is to deploy and maintain. Install as few Client Access and Mailbox servers and create as few Unified Messaging components —like UM dial plans, auto attendants, and UM mailbox policies—as you need to support your business and organizational goals. Large enterprises with complex network and telephony environments, multiple business units, or other complexities will require more planning than smaller organizations with relatively straightforward Unified Messaging needs.

The following are some of the areas that you should consider when planning for Unified Messaging in your organization:

- **Organizational requirements** Evaluate your business needs, the usefulness of deploying a voice mail system, your physical network and business topology, and security requirements for your organization.
- **Telephony requirements** Review your existing telephony, circuit-switched network, and voice mail system.
- **Network requirements** Analyze your network topology, your current packet-switched IP network design including your LAN and WAN connectivity points, and devices.
- **Active Directory (directory service)** Inspect your current implementation and design and think about how to integrate UM.
- **Deployment model** Decide whether you want to have a hybrid, online-only, or on-premises UM

deployment.

- **Exchange requirements** Determine the following:
  - How many users will be using voice mail.
  - Which UM features and services you want to deploy, such as concurrent calls, internal and external access for users, incoming faxing, Voice Mail Preview, and so on.
  - The number of Client Access and Mailbox servers you'll need to deploy.
  - The storage requirements and quotas for voice mail users.
  - The best design for high availability and site resiliency. This includes UM system requirements, providing a highly available and scalable UM deployment, and system hardware requirements to ensure performance.
- **Integration with telephony components and devices** Decide whether to use traditional telephony equipment or Microsoft Lync Server. Consider where to place VoIP gateways, telephony equipment, and Client Access and Mailbox servers, and whether you want to enable Enterprise Voice in your organization.

Connecting your telephony network

Unified Messaging requires that you integrate your Exchange Server deployment with your existing telephony system or integrate it with Microsoft Lync Server. You need to make a careful analysis of your existing telephony infrastructure and Microsoft Lync Server, and then follow the correct planning steps so you can deploy and manage UM voice mail successfully.

VoIP gateways Choosing the correct VoIP gateway, IP PBX, SIP-enabled PBX, or SBC is just the first step in integrating your telephony network with UM. You must configure those devices to work with UM, deploy the required Client Access and Mailbox servers, and create and configure all necessary UM components. These components allow you to make the connection from your circuit-based protocol network to your IP data network and enable voice mail for your users.

Microsoft Lync Server Unified Messaging can use Microsoft Lync Server to combine voice messaging, instant messaging, enhanced presence, audio/video conferencing, and email into a familiar, integrated communications experience. Integrating UM and Microsoft Lync Server has the following benefits:

- Enhanced presence notifications across a variety of applications that keep users informed of the availability of contacts.
- Integration of instant messaging, voice messaging, conferencing, email, and other communication methods, which enables users to select the most appropriate method for the task. Users can also switch from one method to another as needed.
- Availability of communications alternatives from any location where an Internet connection is available.
- A smart client (Microsoft Lync) for telephony, instant messaging, and conferencing.
- Continuity of the user experience across multiple devices.

For more information about Microsoft Lync Server, see Microsoft Lync Server.

Note:

Planning and deploying Unified Messaging can pose certain challenges. Depending on your technical experience with Exchange and voice mail systems, you might want to obtain the assistance of a Unified Messaging specialist. An Exchange Unified Messaging specialist will help make sure that there's a smooth transition from a legacy or third-party voice mail system to Exchange Unified Messaging. Performing a new deployment or upgrading a legacy voice mail system requires significant knowledge about VoIP gateways, PBXs, and Unified Messaging. For more information about how to contact a Unified Messaging specialist, see Microsoft Exchange Server 2013 Unified Messaging (UM) Specialists.

Deployment steps

Many deployment options are available for Unified Messaging. Each option has several steps in common that are required to create a scalable and highly-available system to support large numbers of users. These steps are as follows:

1. Deploy and configure your telephony components or Microsoft Lync Server with Unified Messaging.
2. Verify that you've correctly installed the Client Access and Mailbox servers that are required by Unified Messaging.
3. Create and configure the required Unified Messaging components, including UM dial plans, UM IP gateways, UM hunt groups, and UM mailbox policies.
4. Perform post-deployment tasks, including obtaining certificates for mutual TLS, creating UM auto attendants, and configuring faxing.

For details about deploying Unified Messaging, see Deploy Exchange 2013 UM.

If you're integrating your Unified Messaging environment with Microsoft Lync Server, there are additional planning considerations. For details, see Deploying Exchange 2013 UM and Lync Server overview.

Deploying voice mail and UM

Exchange Server 2013 > Unified Messaging >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-08-05*

Exchange Unified Messaging (UM) enables you to provide voice mail services to users in your organization. When you deploy Unified Messaging, you must either integrate your Exchange Server deployment with the existing telephony system for your organization or integrate it with Microsoft Lync Server. A successful deployment requires you to make a careful analysis of your existing telephony infrastructure and perform the correct planning steps to deploy and manage voice mail in Unified Messaging. If you’re integrating Exchange with Lync Server, you must also familiarize

yourself with that product.

When you're deploying Unified Messaging, you have multiple options depending on the telephony hardware found in your organization. If you're connecting UM to your telephony network, you may have one of the following telephony configurations in your organization:

- One or multiple VoIP gateways with one or multiple PBXs
- One or multiple IP PBXs
- One or multiple SIP-enabled PBXs
- Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server 2010 or 2013

⚠**Warning:**
When you're deploying Exchange UM in a hosted or hybrid environment, you must deploy session border controllers (SBCs). SBCs don't enable UM to connect to a telephony network or provide a dial tone for an organization. However, they do connect your on-premises UM deployment to a datacenter using the IP protocol over a public or private WAN.

Telephony Hardware Choosing the correct VoIP gateway, IP PBX, or SBC is only the first step in integrating your telephony network with UM. You must configure those devices to work with UM, deploy the required Client Access and Mailbox servers, and create and configure all needed UM components. These components allow you to connect your circuit-based protocol network to your IP data network and enable voice mail for users in your organization. For details, see Telephony advisor for Exchange 2013.

Microsoft Lync Server Unified Messaging can use Microsoft Lync Server to combine voice messaging, instant messaging, enhanced presence, audio/video conferencing, and email into a familiar, integrated communications experience. Integrating UM and Lync Server has the following benefits:

- Enhanced presence notifications across a variety of applications that keep users informed of the availability of contacts.
- Integration of instant messaging, voice messaging, conferencing, email, and other communication methods to enable users to select the most appropriate method for the task. Users can also switch from one method to another as needed.
- Availability of communications alternatives from any location where an Internet connection is available.
- A smart client (Microsoft Lync) for telephony, instant messaging, and conferencing.
- Continuity of the user experience across multiple devices.

For more information about Lync Server, see Microsoft Lync Server.

Deployment steps

Many deployment options are available for Unified Messaging. Each option has several steps in common that are required to create a scalable and highly available system to support large numbers of users. These steps are as follows:

1. Deploy and configure your telephony components or Microsoft Lync Server with Unified

Messaging.

2. Verify that you've correctly installed the Client Access and Mailbox servers that are required by Unified Messaging.

⚠**Warning:**
You must deploy at least one Exchange 2013 Mailbox server in your organization before you configure the VoIP gateways or IP PBXs to send UM SIP and RTP traffic to the Exchange 2013 Client Access servers.

3. Create and configure the required Unified Messaging components including UM dial plans, UM IP gateways, UM hunt groups, and UM mailbox policies.
4. Perform post-deployment tasks including deploying certificates for mutual TLS, creating UM auto attendants, and client features.

For details about deploying Unified Messaging, see Deploy Exchange 2013 UM.

If you're integrating your Unified Messaging environment with Microsoft Lync Server, there are additional planning considerations. For details, see Deploying Exchange 2013 UM and Lync Server overview.

Deploy Exchange 2013 UM

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-08-05*

Unified Messaging (UM) requires that you integrate your Exchange Server deployment with the existing telephony system for your organization. A successful deployment requires you to make a careful analysis of your existing telephony infrastructure and perform the correct planning steps to deploy and manage voice mail in Unified Messaging.

Contents

Before you deploy

Before you deploy Unified Messaging, we recommend that you familiarize yourself with the concepts in the following topics:

- UM dial plans
- UM IP gateways

- UM services
- UM hunt groups
- Automatically answer and route incoming calls
- UM mailbox policies
- Voice mail for users

Deploying Unified Messaging

Whether you're deploying UM using IP Private Branch eXchanges (IP PBXs), VoIP gateways, or Microsoft Lync Server, all the deployment options for Unified Messaging have several steps in common. These steps are required to create a scalable and highly available system to support large numbers of Unified Messaging users. These steps are as follows:

1. Deploy and configure your telephony components for Unified Messaging.
2. Verify that you've correctly installed the Client Access server running the Microsoft Exchange Unified Messaging Call Router service and the Mailbox server running the Microsoft Exchange Unified Messaging service.
3. Create and configure the required Unified Messaging components.
4. Perform any post-deployment tasks for Unified Messaging.

Deploy and configure telephony components

To successfully deploy Unified Messaging in an Exchange organization, the Exchange administrator needs to become knowledgeable about data networking concepts and telephony terminology and concepts and be able to correctly configure the telephony components that UM requires. Performing a new deployment or upgrading a legacy voice mail system requires significant knowledge about telephony networks and Unified Messaging.

Generally, you need to complete three tasks to successfully configure the telephony components that UM requires:

1. **Provision PBX lines** The first step in deploying a scalable UM solution is to provision PBX lines.
2. **Organize channels** After you provision PBX-based voice channels, you can organize the channels into hunt groups.
3. **Deploy VoIP gateways** After you organize your voice channels as hunt groups, you need to end these channels at VoIP gateways. VoIP gateways are used with a legacy PBX to convert the circuit-switched protocols found on a telephony network to IP-based packet-switched protocols.

When you integrate your organization's telephony and data networks during the deployment of Unified Messaging, you need to configure the telephony and data networking components correctly. You also need to configure the following components or interfaces to successfully deploy Unified Messaging:

- **Configure the connection from the PBXs in your organization to communicate with your VoIP gateways.** For details, see Connect a VoIP gateway to communicate with a PBX.
- **Configure the connection from the VoIP gateway interface to the PBX.** For more information about how to configure your PBX interface to communicate with your supported

VoIP gateway, see the product documentation that's specific to your PBX or see Connect a VoIP gateway to communicate with a PBX.

- **Configure the connection from the VoIP gateway interface to the Client Access and Mailbox servers.** For details, see Connect a VoIP gateway, IP PBX, or session border controller to UM.
- **Configure the connection from Client Access and Mailbox servers to the VoIP gateway interface.** For details, see Connect UM to a supported VoIP gateway.

Return to top

Install the Mailbox and Client Access servers

Different deployment paths are available for organizations that plan to deploy Exchange Unified Messaging. Although these paths all lead to the same end—a successful deployment of Unified Messaging—each path is slightly different because each customer's needs and starting points are different. However, generally there are common starting points and paths that cover all supported deployment scenarios, including new installations and upgrades. Follow these steps to deploy your Client Access and Mailbox servers:

1. Verify that your existing infrastructure meets certain prerequisites. For details, see Exchange 2013 prerequisites.
2. Deploy your new Exchange 2013 organization. For details, see Install Exchange 2013 using the Setup wizard.

⚠**Warning:**
You must deploy at least one Exchange 2013 Mailbox server in your organization before you configure the VoIP gateways or IP PBXs to send UM SIP and RTP traffic to the Exchange 2013 Client Access servers.

3. Verify that you've correctly installed the Client Access and Mailbox servers. After you install the servers, we recommend that you verify the installation and review the server setup logs. For details, see Verify an Exchange 2013 installation.

Add the required UM language packs

UM language packs enable callers and Outlook Voice Access users to interact with the voice mail system in multiple languages. After you install an additional language pack on a Mailbox server, callers and Outlook Voice Access users can hear email messages and interact with the voice mail system in that language.

When you first install Exchange, U.S. English will be the default language, and the only available language option for your dial plan. After you install a UM language pack on a Mailbox server, the language associated with the language pack will be listed as an available option when you configure the default language for the dial plan. By default, because UM auto attendants are associated with a UM dial plan when they're created, they use the default language setting of the associated UM dial plan. However, this setting can be changed after the UM auto attendant is created.

You can add UM language packs by using the Setup.exe command or by running the *<UMLanguagePack>*.exe installation program after you've downloaded the UM language pack from Exchange Server 2013 UM Language Packs. However, you have to use the Setup.exe command to remove a UM language pack. There's no Exchange Management Shell cmdlet that you can use to add or remove languages from a Mailbox server. For more information about how to install a UM language pack, see Install a UM language pack.

Note:
By default, when you install a Mailbox server, the U.S. English language (en-US) is installed. It can't be removed unless you remove the Mailbox server from the computer.

Return to top

Create and configure UM components

Several UM components are required for the deployment and operation of Unified Messaging. Unified Messaging components connect the telephony infrastructure with the Unified Messaging environment. After you've successfully installed the Client Access and Mailbox servers, follow these steps.

Step 1: Create and configure UM dial plans

UM dial plans are important to the operation of Unified Messaging and are required to successfully deploy Unified Messaging on your network. After you've successfully installed your Client Access and Mailbox servers, a UM dial plan will be the first component that you'll create.

By default, UM dial plans and Client Access and Mailbox servers that are associated with the dial plan send and receive data without using encryption. In Unsecured mode, the VoIP and SIP traffic won't be encrypted. When you create the dial plan or after you've created the dial plan, you can configure the dial plan to encrypt the VoIP and SIP traffic by using Mutual Transport Layer Security (mutual TLS). If you will be using mutual TLS, you will set the dial plan to SIP secured or Secured, set the UM start mode to TLS or Dual, and create and distribute a trusted certificate be the Exchange servers and the VoIP gateways, IP PBXs or session border controllers (SBCs). After you configure the VoIP security setting, you'll have to configure the startup mode for the Client Access and Mailbox servers. For details, see Configure the startup mode on a Mailbox server or Configure the startup mode on a Client Access server.

Perform the following procedure to create a new UM dial plan.

Create a UM dial plan

1. In the Exchange admin center (EAC), navigate to **Unified Messaging** > **UM dial plans**, and then click **Add +**.
2. On the **New UM Dial Plan** page, complete the following boxes:
   - **Name** Type the name of the dial plan. A UM dial plan name is required and must be unique. The name you type is used only for display purposes in the EAC and the Shell. The maximum length of a UM dial plan name is 64 characters, and it can include spaces. However, it can't

include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

Important:

Although the box for the name of the dial plan can accept 64 characters, the name of the dial plan can't be longer than 49 characters. This is because, when you create a dial plan, a default UM mailbox policy is also created that has the name <*DialPlanName*> Default Policy. The *name* parameter for both the UM dial plan and UM mailbox policy can be 64 characters long.

- **Extension length (digits)** Enter the number of digits for extension numbers in the dial plan. The number of digits for extension numbers is based on the telephony dial plan created on a PBX. For example, if a user associated with a telephony dial plan dials a four-digit extension to call another user in the same telephony dial plan, you select 4 as the number of digits in the extension.

This is a required box that has a value range from 1 through 20. The typical extension length is from 3 through 7 numbers. If your existing telephony environment includes extension numbers, you must specify a number of digits that matches the number of digits in those extensions.

When you create a Telephone extension dial plan, you're required to enter an extension number for the user when they're linked to a Telephone extension dial plan. An extension number is also required with Session Initiation Protocol (SIP) dial plans or E.164 dial plans when a UM-enabled user is linked to a SIP URI or E.164 dial plan. This extension number is used by Outlook Voice Access users when they access their Exchange mailbox.

- **Dial plan type** A Uniform Resource Identifier (URI) is a string of characters that identifies or names a resource. The main purpose of this identification is to enable VoIP devices to communicate with other devices over a network using specific protocols. URIs are defined in schemes that define a specific syntax and format and the protocols for the call. In simple terms, this format is passed from the IP PBX or PBX. After you create a UM dial plan, you won't be able to change the URI type without deleting the dial plan, and then re-creating the dial plan to include the correct URI type. You can select one of the following URI types for the dial plan:
  - **Telephone extension** This is the most common URI type. The calling and called party information from the VoIP gateway or IP Private Branch eXchange (PBX) is listed in one of the following formats: Tel:512345 or 512345@<*IP address*>. This is the default URI type for dial plans.
  - **SIP URI** Use this URI type if you must have a Session Initiation Protocol (SIP) URI dial plan such as an IP PBX that supports SIP routing or if you're integrating Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server and Unified Messaging. The calling and called party information from the VoIP gateway. IP PBX, or Communications Server 2007 R2 or Lync Server is listed as a SIP address in the following format: sip:<*username*>@<*domain* or *IP address*>:Port.
  - **E.164** E.164 is an international numbering plan for public telephone systems in which each assigned number contains a country code, a national destination code, and a subscriber number. The calling and called party information sent from the VoIP gateway or IP PBX is listed in the following format: Tel:+14255550123.

Warning:

After you create a dial plan, you will be unable to change the URI type without deleting the dial plan, and then re-creating the dial plan to include the correct URI type.

- **VoIP security mode** Use this drop-down list to select the VoIP security setting for the UM dial plan. You can select one of the following security settings for the dial plan:
  - **Unsecured** By default, when you create a UM dial plan, it is set to not encrypt the SIP signaling or RTP traffic. In unsecured mode, the Client Access and Mailbox servers associated the UM dial plan send and receive data from VoIP gateways, IP PBXs, SBCs and other Client Access and Mailbox servers using no encryption. In unsecured mode, neither the Realtime Transport Protocol (RTP) media channel nor the SIP signaling information is encrypted.
  - **SIP secured** When you select **SIP secured**, only the SIP signaling traffic is encrypted, and the RTP media channels still use TCP, which isn't encrypted. With SIP secured, Mutual Transport Layer Security (TLS) is used to encrypt the SIP signaling traffic and VoIP data.
  - **Secured** When you select **Secured**, both the SIP signaling traffic and the RTP media channels are encrypted. Both the secure signaling media channel that uses Secure Realtime Transport Protocol (SRTP) and the SIP signaling traffic use mutual TLS to encrypt the VoIP data.
- **Country/Region code** Use this box to type the country/region code number to be used for outgoing calls. This number will automatically be prepended to the telephone number that's dialed. This box accepts from 1 through 4 digits. For example, in the United States, the country/region code is 1. In the United Kingdom, it's 44.

3. Click **Save**.

Important:

In previous versions of Exchange, the Unified Messaging server had to be added to a UM dial plan. In Exchange 2013, Client Access and Mailbox servers can’t be associated with a Telephone extension or E.164 dial plan. Client Access and Mailbox servers will answer all incoming calls for all types of dial plans. However, if you’re integrating UM with Microsoft Lync Server, you must add all Client Access and Mailbox servers to all SIP URI dial plans to enable call routing to work correctly with Lync Server.

Return to top

Step 2: Create and configure your UM IP gateways

A UM IP gateway represents either a VoIP gateway hardware device or an IP PBX. The combination of the UM IP gateway and a UM hunt group establishes a link between a VoIP gateway or IP PBX and a UM dial plan.

If you've created or enabled VoIP security on a dial plan, the UM IP gateway that you create by using one of the following procedures will be associated with a UM dial plan that uses VoIP security. In that case, you must use a fully qualified domain name (FQDN) to create the UM IP gateway, and not an IP address. You must also configure the UM IP gateway to listen on TCP port 5061. To configure a UM IP gateway to listen on TCP port 5061, run the following command: `Set-`

`UMIPGateway -identity MyUMIPGateway -Port 5061`. You must also verify that any VoIP gateways or IP PBXs have also been configured to listen on port 5061 for mutual TLS.

Perform the following procedure to create a new UM IP gateway.

Create a UM IP gateway

1. In the EAC, navigate to **Unified Messaging** > **UM IP Gateways**, and then click **Add +**.
2. On the **New UM IP gateway** page, enter the following information:
   - **Name** Use this box to specify a unique name for the UM IP gateway. This is a display name that appears in the EAC. If you have to change the display name of the UM IP gateway after it's been created, you must first delete the existing UM IP gateway, and then create another UM IP gateway that has the appropriate name. The UM IP gateway name is required, but it's used for display purposes only. Because your organization may use multiple UM IP gateways, we recommend that you use meaningful names for your UM IP gateways. The maximum length of a UM IP gateway name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
   - **Address** You can configure a UM IP gateway with either an IP address or a fully qualified domain name (FQDN). Use this box to specify the IP address configured on the VoIP gateway, SIP-enabled PBX, IP PBX, or SBC, or an FQDN. This box accepts only FQDNs that are valid and formatted correctly.

You can enter alphabetical and numeric characters in this box. IPv4 addresses, IPv6 addresses, and FQDNs are supported. If you want to use mutual TLS between a UM IP gateway and a dial plan operating in either SIP secured or Secured mode, you must configure the UM IP gateway with an FQDN. You must also configure it to listen on port 5061 and verify that any VoIP gateways or IP PBXs have also been configured to listen for mutual TLS requests on port 5061. To configure a UM IP gateway, run the following command: `Set-UMIPGateway -identity MyUMIPGateway -Port 5061`.
If you use an FQDN, you must also make sure that you've correctly configured a DNS host record for the VoIP gateway so that the host name will be correctly resolved to an IP address. Also, if you use an FQDN instead of an IP address, and the DNS configuration for the UM IP gateway is changed, you must disable and then enable the UM IP gateway to make sure that configuration information for the UM IP gateway is updated correctly

   - **UM dial plan** Click **Browse** to select the UM dial plan that you want to associate with the UM IP gateway. When you select a UM dial plan to associate with a UM IP gateway, a default UM hunt group is also created and associated with the UM dial plan that you selected. If you don't select a UM dial plan, you must manually create a UM hunt group and then associate that UM hunt group with the UM IP gateway that you create.
3. Click **Save**.

Step 3: Create and configure your UM hunt groups (optional)

Hunt group is a term that's used to describe a group of PBX or IP PBX resources or extension numbers that are shared by users. Hunt groups are used to efficiently distribute calls into or out of a given business unit.

If you've created a UM IP gateway and associated the UM IP gateway with a UM dial plan, a default UM hunt group has been created. You can associate another UM hunt group with the same or a different UM IP gateway, depending on the number of UM IP gateways that you've created.

When you create a UM hunt group, you enable all Mailbox servers that are specified within the UM dial plan to communicate with a VoIP gateway. For details, see UM hunt groups.

Create a UM hunt group

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** ✎.
2. On the **UM Dial Plan** page, under **UM Hunt Groups**, click **Add +**.
3. On the **New UM Hunt Group** page, complete the following boxes:
   - **Associated UM IP gateway** This display-only box shows the name of the UM IP gateway that will be associated with the UM hunt group.
   - **Name** Use this box to create the display name for the UM hunt group. A UM hunt group name is required and must be unique, but it's used only for display purposes in the EAC and the Shell. If you have to change the display name of the hunt group after it's been created, you must first delete the existing hunt group and then create another hunt group that has the appropriate name.

If your organization uses multiple hunt groups, we recommend that you use meaningful names for your hunt groups. The maximum length of a UM hunt group name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

   - **Dial plan** Click **Browse** to select the dial plan that will be associated with the UM hunt group. Associating a hunt group with a dial plan is required. A UM hunt group can be associated with only one UM IP gateway and one UM dial plan.
   - **Pilot identifier** Use this box to specify a string that uniquely identifies the pilot identifier or pilot ID configured on the PBX or IP PBX.

An extension number or a Session Initiation Protocol (SIP) Uniform Resource Identifier (URI) can be used in this box. Alphanumeric characters are accepted in this box. For legacy PBXs, a numeric value is used as a pilot identifier. However, some IP PBXs can use SIP URIs.

4. Click **Save**.

Return to top

Step 4: Create and configure a UM mailbox policy

UM mailbox policies are required when you enable users for Unified Messaging. The mailbox of each UM-enabled user must be linked to a single UM mailbox policy. After you create a UM mailbox policy, you link one or more UM-enabled mailboxes to the UM mailbox policy. This lets you control PIN security settings such as the minimum number of digits in a PIN or the maximum number of failed sign-in attempts for the UM-enabled users who are associated with the UM mailbox policy.

Every time that you create a UM dial plan, a UM mailbox policy is also created. The UM mailbox policy will be named <*DialPlanName*> Default Policy. However, if you have to create a new UM

mailbox policy, perform the following procedure.

Create a UM mailbox policy

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then click **Edit** ✎.
2. On the **UM dial plan** page, under **UM Mailbox Policies**, click **Add +**.
3. On the **New UM Mailbox Policy** page, in the **Name** text box, enter the name of the new UM mailbox policy.

Use this box to specify a unique name for the UM mailbox policy. This is a display name that appears in the EAC. If you must change the display name of the UM mailbox policy after it's been created, you must first delete the existing UM mailbox policy, and then create another UM mailbox policy that has the appropriate name. You can't delete a UM mailbox policy if any UM-enabled users are associated with it.

The UM mailbox policy name is required, but it is used for display purposes only. Because your organization may use multiple UM mailbox policies, we recommend that you use meaningful names for your UM mailbox policies. The maximum length of a UM mailbox policy name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

4. Click **Save** to save the new UM mailbox policy. When you save the UM mailbox policy, all of the default settings including PIN policies, voice mail features, and Protected Voice Mail settings are enabled. If you want to customize or change any default settings, use the **Set-UMMailbox** cmdlet to change the settings for the UM mailbox policy you just created.

Step 5: Create and configure UM auto attendants (optional)

Unified Messaging enables you to create one or more UM auto attendants, depending on the needs of your organization. When you create a UM auto attendant, you create a voice menu system for your organization. Callers from outside or inside your organization can then move through the menu system to locate and place or transfer calls to users or departments in your organization.

Callers can move through the menu system by using dual tone multi-frequency (DTMF), also known as touchtone, or voice inputs. For Automatic Speech Recognition (ASR) to work, so users can use voice inputs, you must speech-enable the UM auto attendant.

Creating and using auto attendants is optional in Unified Messaging. However, if you want to create a new UM auto attendant, perform the following procedure.

Create a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**, select the UM dial plan for which you want to add an auto attendant, and then click **Edit** ✎.
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, click **Add +**.
3. On the **New UM auto attendant** page, complete the following boxes:
   - **Name** Use this box to create the display name for the UM auto attendant. A UM auto attendant name is required and must be unique. However, it's used only for display purposes in the EAC and the Shell.

If you have to change the display name of the auto attendant after it's created, you must first delete the existing UM auto attendant and then create another auto attendant that has the appropriate name. If your organization uses multiple UM auto attendants, we recommend that you use meaningful names for your UM auto attendants. The maximum length of a UM auto attendant name is 64 characters, and it can include spaces.

Although you can name a new UM auto attendant to include spaces, if you integrate Unified Messaging with Office Communications Server 2007 R2 or Microsoft Lync Server, the name of the auto attendant can't include spaces. Therefore, if you created an auto attendant that has spaces in the display name and you're integrating with Office Communications Server 2007 R2 or Lync Server, you must first delete that auto attendant and then create another auto attendant that doesn't include spaces in the display name.

- **Create this auto attendant as enabled** Select this check box to enable the auto attendant to answer incoming calls when you finish creating the UM auto attendant. By default, a new auto attendant is created as disabled.

If you decide to create the UM auto attendant as disabled, you can use the EAC or the Shell to enable the auto attendant after you finish creating the auto attendant.

- **Set the auto attendant to respond to voice commands** Select this check box to speech-enable the UM auto attendant. If the auto attendant is speech-enabled, callers can respond to the system or custom prompts used by the UM auto attendant using touchtone or voice inputs. By default, the auto attendant won't be speech-enabled when it's created.

For callers to use a speech-enabled auto attendant in a language other than U.S. English (en-US), you must install the appropriate UM language pack and configure the properties of the auto attendant to use this language. The en-US UM language pack is installed by default when you install a Mailbox server.

- **Access numbers** Use this box to enter the extension or telephone numbers that callers will use to reach the auto attendant. Type an extension number or telephone number in the box, and then click **Add +** to add the number to the list. The number of digits in the extension number or telephone number that you provide doesn't have to match the number of digits for an extension number configured on the associated UM dial plan. This is because direct calls are allowed to UM auto attendants.

The number of extension numbers or pilot identifiers you can enter is unlimited. However, you may create a new auto attendant without listing an extension number or telephone number. An extension number or telephone number isn't required.

You can edit or remove an existing extension number or pilot identifier. To edit an existing extension number or telephone number, click **Edit** ✎. To remove an existing extension number or telephone number from the list, click **Remove −**.

4. Click **Save**.

Return to top

Post-deployment tasks for Unified Messaging

After you complete a new installation of the Client Access and Mailbox servers and have successfully deployed Unified Messaging, you should complete the post-deployment tasks. The post-deployment tasks will help you enable users for Unified Messaging, secure your UM deployment, and deploy incoming faxing for UM-enabled users.

Enable users for voice mail

After you've deployed your VoIP gateways or IP PBXs, installed the Client Access and Mailbox servers, and created the components required for Unified Messaging, you need to enable your users for Unified Messaging. For details, see Enable a user for voice mail.

Protect voice mail

Unified Messaging can be configured to use Active Directory Rights Management Services (AD RMS) to protect voice messages for an organization. This feature is known as Protected Voice Mail. When a voice message is protected, the recipient is not only blocked from forwarding the message, but UM also assures that only the intended recipient or recipients of the message can access its content. Protected voice messages can be accessed by using Microsoft Outlook 2010 or later, Outlook Web App, or Outlook Voice Access. For details, see Protect voice mail.

Mutual TLS for UM

To use mutual TLS to encrypt SIP and Realtime Transport Protocol (RTP) traffic that's sent and received by your Client Access and Mailbox servers, perform the following tasks:

- Run the Exchange Certificate wizard. For details, see Deploying certificates for UM.
- Import the certificate on the Client Access and Mailbox servers.
- Import the required certificates on the VoIP gateways and the IP PBX and Client Access and Mailbox servers in your organization.
- Configure VoIP security on the UM dial plans. For details, see Configure the VoIP security setting.
- Configure the startup mode on the Client Access and Mailbox servers. For details, see Configure the startup mode on a Mailbox server and Configure the startup mode on a Client Access server.
- Configure the UM IP gateways to listen on port 5061. For details, see Configure the listening port.

PIN policies for UM-enabled users

In Unified Messaging, PIN policies are defined and configured on a UM mailbox policy. When you enable a user for Unified Messaging, you associate the user with an existing UM mailbox policy. The UM PIN policies that are configured on the UM mailbox policy should be based on the security requirements of your organization. For more information about how to configure PIN settings for UM-enabled users, see Set Outlook Voice Access PIN security.

Set up client voice mail features

After you've deployed your servers and the required UM components, there are several optional

voice mail-related features that you can configure. For more information, see the following:

- Setting up Outlook Voice Access
- Allow voice mail users to forward calls
- Allow users to see a voice mail transcript
- Enable voice mail users to receive faxes

Important:
If you're integrating your Unified Messaging environment with Microsoft Lync Server, there are additional planning considerations. For details, see Deploying Exchange 2013 UM and Lync Server overview.

Return to top

Checklist: Deploy Exchange 2013 UM

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-08-05*

Use this checklist to help you install and deploy Unified Messaging (UM) in your organization.

Before you start working with this checklist, make sure you're familiar with the concepts in:

- Unified Messaging
- New voice mail features
- Planning for Unified Messaging

For step-by-step guidance about how to deploy UM and Microsoft Lync Server, see Checklist: Integrate Exchange 2013 UM with Lync Server.

Checklist for deploying Unified Messaging

Done?	Tasks	Topic
	Deploy and configure telephony components.	Connect UM to your telephone system
	Review the system requirements before installing Exchange 2013.	Exchange 2013 system requirements
	Verify that you meet the prerequisites for installation.	Exchange 2013 prerequisites

	Install the required Client Access and Mailbox servers. **⚠Warning:** You must deploy at least one Exchange 2013 Mailbox server in your organization before you configure the VoIP gateways or IP PBXs to send UM SIP and RTP traffic to the Exchange 2013 Client Access servers.	Install Exchange 2013 using the Setup wizard
	Verify the installation and review the server setup logs.	Verify an Exchange 2013 installation
	If required, install the required UM language packs.	Install a UM language pack
	Create the number of dial plans required for your organization.	Create a UM dial plan
	Configure the dial plan security setting.	Configure the VoIP security setting
	Configure the UM startup mode for each Client Access and Mailbox server.	Configure the startup mode on a Mailbox server Configure the startup mode on a Client Access server
	Configure the number of concurrent calls on your Mailbox servers.	Configure the number of incoming calls on a Mailbox server
	Configure Outlook Voice Access numbers and other settings.	Manage a UM dial plan
	Configure outbound dialing for Unified Messaging.	Authorize calls using dialing rules Authorize calls for users in a

		dial plan Authorize calls for a group of users
	Create the required number of auto attendants.	Create a UM auto attendant
	Set up and configure each of the UM auto attendants.	Set up a UM auto attendant
	Create, import, and enable a new Exchange certificate for UM or enable a mutually-trusted third-party certificate. Also, import the certificate on all VoIP gateways, IP PBXs, and SBCs.	Add Mailbox and Client Access servers to a SIP URI dial plan
	Restart the Microsoft Exchange Unified Messaging service and the Unified Messaging Call Router service on all Exchange servers to load the required certificates.	Stop the Microsoft Exchange Unified Messaging service Start the Microsoft Exchange Unified Messaging service Stop the Microsoft Exchange Unified Messaging Call Router service Start the Microsoft Exchange Unified Messaging Call Router service
	Create a UM mailbox policy or configure the default UM mailbox policy.	Create a UM mailbox policy Manage a UM mailbox policy
	Enable users for Unified Messaging with an extension number and an E.164 number, if	Enable a user for voice mail

	required.	
	Enable incoming faxing (Optional).	Enable voice mail users to receive faxes
	Set up Protected Voice Mail (Optional).	Protect voice mail

Return to top

Upgrade Exchange 2010 UM to Exchange 2013 UM

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-04*

When you're upgrading a Microsoft Exchange 2010 organization with Unified Messaging (UM) to Exchange 2013 Unified Messaging, there are steps that are required and other steps that were already completed as part of your Exchange 2010 UM deployment. Depending on your telephony environment and the UM components that were created and configured to support Unified Messaging in Exchange 2010, you may need to deploy additional telephony equipment including Voice over IP (VoIP) gateways, IP Private Branch eXchanges (PBXs), or traditional or SIP-enabled PBXs and then create and configure any additional UM components that will be required for Exchange 2013 UM.

What do you need to know before you begin?

- Estimated time to complete this task: 45-90 minutes.
- Verify that you have the appropriate permissions in the Exchange 2010 and Exchange 2013 organization to create and configure all the required components.
- Verify that you've deployed and correctly configured your telephony components, including VoIP gateways and PBXs, IP PBXs, or Session Initiation Protocol (SIP)-enabled PBXs.
- Verify that you've correctly installed and configured the Client Access servers running the Microsoft Exchange Unified Messaging Call Router (UM Call Router) service and Mailbox servers running the Microsoft Exchange Unified Messaging (UM) service. To learn more about UM services, see UM services.

Warning:

You must deploy at least one Exchange 2013 Mailbox server in your organization before you configure the VoIP gateways or IP PBXs to send UM SIP and RTP traffic to the Exchange 2013 Client Access servers.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

How do you do this?

Step 1: Download and install the required UM language packs

UM language packs enable callers and Outlook Voice Access users to interact with the voice mail system in multiple languages. After you install an additional language on an Exchange 2013 Mailbox server, callers and Outlook Voice Access users can hear email messages and interact with the voice mail system in that language. However, to make the language available for all incoming calls, you must install the required UM language packs on all Exchange 2013 Mailbox servers. This is because every Exchange 2013 Mailbox server can answer incoming calls for Unified Messaging.

By default, when you install an Exchange 2013 Mailbox server, the U.S. English (en-US) language pack is installed. It's the only available language option for your dial plan unless you install another UM language pack. (U. S. English can't be removed unless you remove the Mailbox server from the computer.) After you install a UM language pack on an Exchange 2013 Mailbox server, the language associated with the language pack will be listed as an available option when you configure the default language for the dial plan. By default, because a UM auto attendant is linked to a UM dial plan when the auto attendant is created, it uses the default language setting of the linked UM dial plan. However, this setting can be changed after the UM auto attendant is created.

Note:

If U.S. English is the only language that you want to provide for your dial plan, you can skip this step and go to step 2.

You can add UM language packs by using the setup.exe command or by running the *<UMLanguagePack>*.exe installation program after you've downloaded the UM language pack from Exchange Server 2013 UM Language Packs. For more information, see Install a UM language pack.

This example uses setup.exe to install the Japanese (ja-JP) UM language pack.

```
setup.exe /AddUmLanguagePack:ja-JP /s:d:\Exchange
\UMLanguagePacks /IAcceptExchangeServerLicenseTerms
```

Step 2: Move the Exchange 2010 system mailbox used for UM custom greetings, announcements, menus, and prompts to Exchange 2013

Custom greetings, announcements, menus, and prompts are used by Unified Messaging dial plans and auto attendants. The system mailbox named {e0dc1c29-89c3-4034-b678-e6c29d823ed9} is created when you install Exchange 2010 or Exchange 2013 and is used to support features such as Message Approval and Multi-Mailbox Search. This system mailbox is also used to store dial plan and auto attendant custom greetings, announcements, menus, and prompts. If the system mailbox doesn't exist, you can use the **Setup /PrepareAD** command to create it.

By default, system mailboxes aren't visible in the Exchange admin center (EAC). You can get a list of the system mailboxes by running one the following:

This command returns a list of all the system mailboxes.

```
Get-Mailbox -Arbitration
```

This command returns a list of system mailboxes and their individual properties or settings.

```
Get-Mailbox -Arbitration |fl
```

By using this system mailbox, custom greetings, announcements, menus, and prompts can be backed up and restored along with other mailboxes in a database. This reduces the amount of resources that are needed. Storing custom greetings, announcements, menus, and prompts in a system mailbox removes any possible inconsistencies that may have occurred. To learn more about mailbox moves, see Mailbox moves in Exchange 2013.

Optional: Manually export and import dial plan and auto attendant custom greetings, announcements, menus, and prompts

There could be situations that the Exchange 2010 system mailbox has been moved and you still need to export and import the custom greetings, announcements, menus, and prompts that are used with UM dial plans and auto attendants from the Exchange 2010 system mailbox to the Exchange 2013 system mailbox. The system mailbox in both versions is named {e0dc1c29-89c3-4034-b678-e6c29d823ed9}.

Custom greetings, announcements, menus, and prompts are audio files (in .wav or .wma format) that are used by UM for the following purposes:

- On UM dial plans, the audio files are used for customized welcome greetings and informational announcements. They're played when Outlook Voice Access users call in to an Outlook Voice Access number.
- On UM auto attendants, the audio files are used for customized non-business and business hours greetings, informational announcements, menu prompts, and navigation menus. They're played when callers call in to a UM auto attendant.

When you're exporting and importing custom greetings, announcements, menus, and prompts from Exchange 2010 to Exchange 2013, you must use the **Export-UMPrompt** and **Import-UMPrompt** cmdlets. You can't use the EAC to export or import custom prompts. On an Exchange 2010 server, use the **Export-UMPrompt** cmdlet to export the Exchange 2010 dial plan and auto attendant prompts. After you've exported the prompts, you can import them to the Exchange 2013 Mailbox server. When you run the **Export-UMPrompt** cmdlet from your Exchange 2010 server, the command performs a GUID or object identifier lookup for the dial plan or auto attendant in Active Directory and queries it to determine if there are any custom greetings, announcements, menus, or prompts. If found, the custom greetings, announcements, menus, or prompts will be saved to the directory that you specify. After you've exported all custom greetings, announcements, menus, and prompts, use the **Import-UMPrompt** cmdlet to import the prompts into your Exchange 2013 system mailbox.

This example exports the welcome greeting for the UM dial plan `MyUMDialPlan` and saves it as the file `welcomegreeting.wav`.

```
$prompt = Export-UMPrompt -PromptFileName
"customgreeting.wav" -UMDialPlan MyUMDialPlan
set-content -Path "d:\DialPlanPrompts\welcomegreeting.wav"
-Value $prompt.AudioData -Encoding Byte
```

This example imports the welcome greeting `welcomegreeting.wav` from d:\UMPrompts into the UM dial plan `MyUMDialPlan`.

```
[byte[]]$c = Get-content -Path "d:\UMPrompts
\welcomegreeting.wav" -Encoding Byte -ReadCount 0
Import-UMPrompt -UMDialPlan MyUMDialPlan -PromptFileName
"welcomegreeting.wav" -PromptFileData $c
```

This example exports a custom greeting for the UM auto attendant `MyUMAutoAttendant` and saves it to the file `welcomegreetingbackup.wav`.

```
Export-UMPrompt -PromptFileName "welcomegreeting.wav" -
UMAutoAttendant MyUMAutoAttendant
```

```
set-content -Path "e:\UMPromptsBackup\welcomegreeting.wav"
-Value $prompt.AudioData -Encoding Byte
```

This example imports the welcome greeting `welcomegreeting.wav` from d:\UMPrompts into the UM auto attendant `MyUMAutoAttendant`.

```
[byte[]]$c = Get-content -Path "d:\UMPrompts
\welcomegreeting.wav" -Encoding Byte -ReadCount 0
Import-UMPrompt -UMAutoAttendant MyUMAutoAttendant -
PromptFileName "welcomegreeting.wav" -PromptFileData $c
```

To learn more about custom prompts for UM, see:

- Import and export custom greetings, announcements, menus, and prompts
- Import-UMPrompt
- Export-UMPrompt
- UM languages, prompts, and greetings

Step 4: Export and import certificates

If you're using SIP secured or Secured dial plans in your Exchange 2010 organization, you'll need to export and import the certificates that were used to your Exchange 2013 Client Access and Mailbox servers. Mutual Transport Layer Security (mutual TLS) is used to encrypt data sent between your Exchange 2013 servers and the VoIP gateways, IP PBXs, and SIP-enabled PBXs. Certificates bind the identity of the certificate owner to a pair of electronic keys (public and private) that are used to encrypt and sign information digitally. You can use one of the following certificates for the UM and UM Call Router services:

- A self-signed (Exchange) certificate
- An internal public key infrastructure (PKI) certificate
- A third-party commercial certificate

By default, when you install Exchange 2013, two self-signed certificates are created: **Microsoft Exchange Server Auth Certificate** and **Microsoft Exchange**. The **Microsoft Exchange** self-signed certificate can be used for UM to encrypt data, but you must assign the certificate to the UM and UM Call Router services. This self-signed certificate can be copied and then imported on the VoIP gateways, IP PBXs, and SIP-enabled PBXs. However, it can't be used when you're integrating UM with Microsoft Lync Server.

To enable UM to encrypt data that's sent between your Exchange 2013 servers and VoIP gateways, IP PBXs, and SIP-enabled PBXs, you need to do the following:

- Use an existing self-signed UM certificate, create a new self-signed Exchange certificate, submit a certificate request to an internal certification authority for a PKI certificate, or purchase a third-party commercial certificate that you can use for mutual TLS between your Exchange 2013 Mailbox and Client Access servers and VoIP gateways, IP PBXs, and SIP-enabled PBXs.

Create an Exchange self-signed certificate by using the EAC, as follows:

1. In the EAC, navigate to **Servers** > **Certificates**, and then click **Add +**.
2. On the **New Exchange certificate** page, choose **Create a self-signed certificate**, and then select **Next**.
3. Enter a friendly name for the certificate, and then select **Next**.
4. Click **Add +** to select the Exchange servers that you want to apply this certificate to, and then select **Next**.
5. Specify the domains that you want to be included in your certificate, and then select **Next**. If you want to add a domain for a service, click **Edit** ✎.
6. Verify that the domains you included are correct, and then select **Finish**.

Important:

When you use the EAC to create a certificate, you won't be prompted to enable the services for the certificate. After the certificate has been created, you can use the EAC to enable the services. For more information about how to enable a certificate for services, see Assign a certificate to the UM and UM Call Router services.

Create an Exchange self-signed certificate by running the following command in the Shell.

```
New-ExchangeCertificate -Services 'UM, UMCallRouter' -
DomainName '*.northwindtraders.com' -FriendlyName
'UMSelfSigned' -SubjectName
'C=US,S=WA,L=Redmond,O=Northwindtraders,OU=Servers,CN=
Northwindtraders.com' -PrivateKeyExportable $true
```

Note:

If you specify the services you want to enable by using the *Services* parameter, you will be prompted to enable the services for the certificate you created. In this example, you will be prompted to enable the certificate for the Unified Messaging and Unified Messaging Call Router services. For more information about how to enable a certificate for services, see Assign a certificate to the UM and UM Call Router services.

- Import the certificate that will be used on all Exchange 2013 Client Access and Mailbox servers in your organization. If you use the Exchange 2013 self-signed certificate, you'll need to copy the certificate, then import it on the VoIP gateways, IP PBXs, or SIP-enabled PBXs. If you use the self-signed certificate from Exchange 2010, the Subject Alternative Name (SAN) must contain the machine names of all the Exchange 2013 servers. If you have Exchange 2010 Unified Messaging servers in your organization, you can use the Exchange 2013 self-signed certificate, but you must add the machine names of the Exchange 2010 UM servers to the SAN in the Exchange 2013 certificate.
- Enable or assign the certificate to be used to the UM and UM Call Router services on the Client Access and Mailbox servers in your organization.

Enable the UM service and UM Call Router service on all Exchange 2013 servers to use the Exchange self-signed certificate by using the EAC, as follows:

1. In the EAC, navigate to **Servers** > **Certificates**, select the certificate you want to enable services on, and then click **Edit** ✎.

2. On the **Procedure** page, select **Services**, select **Unified Messaging**, and then select **Unified Messaging call router**.

Enable an Exchange self-signed certificate by running the following command in the Shell.

```
Enable-ExchangeCertificate -Thumbprint
5113ae0233a72fccb75b1d0198628675333d010e -Services 'UM,
UMCallRouter'
```

- Configure any new or existing UM dial plans as SIP secured or Secured.
- Configure the UM startup mode to TLS or Dual on the Client Access and Mailbox servers in your organization.
- Create and configure new or existing UM IP gateways with a fully qualified domain name (FQDN).
- Configure the listening port on the UM IP gateways to use TLS port 5061.
- Restart the UM Call Router service on all Exchange 2013 Client Access servers and restart the UM service on all Exchange 2013 Mailbox servers. To learn more about UM services, see UM services.

Step 5: Configure the UM startup mode on all Exchange 2013 Client Access servers

If you're using SIP secured or Secured dial plans, you must configure the UM startup mode on your Exchange 2013 Client Access servers. You can specify the UM startup mode for the UM Call Router service on an Exchange 2013 Client Access server by using the EAC or the Exchange Management Shell. By default, the Client Access server will start up in TCP mode, but if you're using Transport Layer Security (TLS) to encrypt Voice over IP (VoIP) traffic, you must configure the Exchange 2013 Client Access server to use TLS or Dual mode. We recommend that all Exchange 2013 Client Access servers be configured to use Dual as the UM startup mode. This is because all Exchange 2013 Client Access servers can answer incoming calls for all UM dial plans, and those dial plans can have different security settings. If you change the UM startup mode, you must restart the UM Call Router service for the change to take effect. To learn more about UM services, see UM services.

Configure the UM startup mode on an Exchange 2013 Client Access server by using the EAC, as follows:

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Exchange server that you want to modify, and then click **Edit** ✎.
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Call Router settings** > **UM startup mode**, select one of the following from the drop-down list:
   - **TCP** Use this option if you aren't using mTLS and are using only Unsecured dial plans.
   - **TLS** Use this option if you're using mTLS and are using only SIP Secured or Secured dial plans.
   - **DUAL** Use this option if you're using mTLS and are using Unsecured, SIP Secured, and Secured dial plans.
5. After you select the UM startup mode, click **Save**.

Configure the UM startup mode on an Exchange 2013 Client Access server by running the following command in the Shell.

```
Set-UMCallRouterSettings -Server
MyUMCallRouter.northwindtraders.com -UMStartupMode Dual
```

Step 6: Configure the UM startup mode on all Exchange 2013 Mailbox servers

If you're using SIP secured or Secured dial plans, you must configure the UM startup mode on your Exchange 2013 Mailbox servers. You can specify the UM startup mode for the UM service on an Exchange 2013 Mailbox server by using the EAC or the Shell. By default, an Exchange 2013 Mailbox server will start up in TCP mode, but if you're using Transport Layer Security (TLS) to encrypt Voice over IP (VoIP) traffic, you must configure the Exchange 2013 Mailbox server to use TLS or Dual mode. We recommend that all Exchange 2013 Mailbox servers be configured to use Dual as the UM startup mode. This is because all Exchange 2013 Mailbox servers can answer incoming calls for all UM dial plans, and those dial plans can have different security settings. If you change the UM startup mode, you must restart the UM service for the change to take effect. To learn more about UM services, see UM services.

Configure the UM startup mode on an Exchange 2013 Mailbox server by using the EAC, as follows:

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Exchange server that you want to modify, and then click **Edit** ✎.
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Service settings** > **UM startup mode**, select one of the following from the drop-down list:
   - **TCP** Use this option if you aren't using mTLS and are using only Unsecured dial plans.
   - **TLS** Use this option if you're using mTLS and are using only SIP Secured or Secured dial plans.
   - **DUAL** Use this option if you're using mTLS and are using Unsecured, SIP Secured, and Secured dial plans.
5. After you select the UM startup mode, click **Save**.

Configure the UM startup mode on an Exchange 2013 Mailbox server by running the following command in the Shell.

```
Set-UMService -Identity MyUMServer -ExternalHostFqdn
host.external.contoso.com -IPAddressFamily Any -
UMStartupMode Dual
```

Step 7: Create or configure existing UM dial plans

Depending on your existing Exchange 2010 deployment, you may be required to create new UM

dial plans or configure your existing dial plans. A UM dial plan represents a set of traditional or SIP-enabled Private Branch eXchanges (PBXs) or IP PBXs that share common user extension numbers. All users' extensions hosted on traditional or SIP-enabled PBXs or IP PBXs within a dial plan contain the same number of digits. Users can dial one another's telephone extensions without appending a special number to the extension or dialing a full telephone number.

UM dial plans are used in Unified Messaging to make sure that user telephone extensions are unique. In some telephony networks, multiple PBXs or IP PBXs exist. In these telephony networks, there could be two users who have the same telephone extension number. UM dial plans resolve this situation. Putting the two users into two separate UM dial plans makes their extensions unique. For more information, see UM dial plans.

If required, you can create a UM dial plan by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**, and then click **New +**.
2. On the **New UM Dial Plan** page, complete the following boxes:
   - **Name** Type the name of the dial plan. A UM dial plan name is required and must be unique. However, the name you type is used only for display purposes in the EAC and the Shell. The maximum length of a UM dial plan name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

Although the box for the name of the dial plan can accept 64 characters, the name of the dial plan can't be longer than 49 characters. If you try to create a dial plan name that contains more than 49 characters, you'll receive an error message. The message will say that the UM mailbox policy couldn't be generated because the UM dial plan name is too long. This happens because, when you create a dial plan a default UM mailbox policy named <*DialPlanName*> **Default Policy** is also created. When the 15 characters in the default policy are added to the name of the dial plan, the total characters exceed the limit. The *name* parameter for both the UM dial plan and UM mailbox policy can be 64 characters. However, if the name of the dial plan is longer than 49 characters, the name of the default UM mailbox policy will be longer than 64 characters, and this isn't allowed by the system.

   - **Extension length (digits)** Enter the number of digits for extension numbers in the dial plan. The number of digits for extension numbers is based on the telephony dial plan created on a PBX. For example, if a user associated with a telephony dial plan dials a four-digit extension to call another user in the same telephony dial plan, you select 4 as the number of digits in the extension.

This is a required box that has a value range from 1 through 20. The typical extension length is from 3 through 7 numbers. If your existing telephony environment includes extension numbers, you must specify a number of digits that matches the number of digits in those extensions.

When you create a Telephone extension dial plan, you're required to enter an extension number for the user if they're linked to a Telephone extension dial plan. An extension number is also required with Session Initiation Protocol (SIP) dial plans or E.164 dial plans when a UM-enabled user is linked to a SIP URI or E.164 dial plan. The extension number is used by Outlook Voice Access users when they access their Exchange mailbox.

   - **Dial plan type** A Uniform Resource Identifier (URI) is a string of characters that identifies or

names a resource. The main purpose of this identification is to enable VoIP devices to communicate with other devices over a network using specific protocols. URIs are defined in schemes that define a specific syntax and format and the protocols for the call. In simple terms, this format is passed from the IP PBX or PBX. After you create a UM dial plan, you won't be able to change the URI type without deleting the dial plan, and then re-creating the dial plan to include the correct URI type. You can select one of the following URI types for the dial plan:

- **Telephone extension** This is the most common URI type. The calling and called party information from the VoIP gateway or IP Private Branch eXchange (PBX) is listed in one of the following formats: Tel:512345 or 512345@<*IP address*>. This is the default URI type for dial plans.
- **SIP URI** Use this URI type if you must have a Session Initiation Protocol (SIP) URI dial plan such as an IP PBX that supports SIP routing or if you're integrating Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server and Unified Messaging. The calling and called party information from the VoIP gateway. IP PBX, or Communications Server 2007 R2 or Lync Server is listed as a SIP address in the following format: sip:<*username*>@<*domain* or *IP address*>:Port.
- **E.164** E.164 is an international numbering plan for public telephone systems in which each assigned number contains a country code, a national destination code, and a subscriber number. The calling and called party information sent from the VoIP gateway or IP PBX is listed in the following format: Tel:+14255550123.

⚠Warning:

After you create a dial plan, you will be unable to change the URI type without deleting the dial plan, and then re-creating the dial plan to include the correct URI type.

- **VoIP security mode** Use this drop-down list to select the VoIP security setting for the UM dial plan. You can select one of the following security settings for the dial plan:
  - **Unsecured** By default, when you create a UM dial plan, it is set to not encrypt the SIP signaling or RTP traffic. In unsecured mode, the Client Access and Mailbox servers associated the UM dial plan send and receive data from VoIP gateways, IP PBXs, SBCs and other Client Access and Mailbox servers using no encryption. In unsecured mode, neither the Realtime Transport Protocol (RTP) media channel nor the SIP signaling information is encrypted.
  - **SIP secured** When you select **SIP secured**, only the SIP signaling traffic is encrypted, and the RTP media channels still use TCP, which isn't encrypted. With SIP secured, Mutual Transport Layer Security (TLS) is used to encrypt the SIP signaling traffic and VoIP data.
  - **Secured** When you select **Secured**, both the SIP signaling traffic and the RTP media channels are encrypted. Both the secure signaling media channel that uses Secure Realtime Transport Protocol (SRTP) and the SIP signaling traffic use mutual TLS to encrypt the VoIP data.
- **Country/Region code** Use this box to type the country/region code number to be used for outgoing calls. This number will automatically be prepended to the telephone number that's dialed. This box accepts from 1 through 4 digits. For example, in the United States, the

country/region code is 1. In the United Kingdom, it's 44.

3. Click **Save**.

If required, you can create a UM dial plan by running the following command in the Shell.

```
New-UMDialplan -Name MyUMDialPlan -URIType E164 -
NumberOfDigitsInExtension 5 -VoIPSecurity Secured
```

If required, you can configure an existing UM dial plan by using the EAC, as follows:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to view or modify, and then click **Edit** ✎.
3. On the **UM Dial Plan** page, click **Configure**. Use the configuration options to view specific dial plan settings and to enable or disable features.

If required, you can configure an existing UM dial plan by using the Shell:

```
Set-UMDialplan -Identity MyDialPlan -AccessTelephoneNumbers
4255551234 -AudioCodec Wma -CallAnsweringRulesEnabled
$false -OutsideLineAccessCode 9 -VoIPSecurity SIPSecured
```

When you deployed Exchange 2010 Unified Messaging, you were required to add a Unified Messaging server to a UM dial plan for it to answer incoming calls. This is no longer required. In Exchange 2013, Client Access and Mailbox servers can't be linked with a Telephone extension or E.164 dial plan, but must be linked to SIP URI dial plans. Client Access and Mailbox servers will answer all incoming calls for all types of dial plans.

Step 8: Create or configure existing UM IP gateways

Depending on your existing Exchange 2010 deployment, you may be required to create new UM IP gateways or configure your existing ones. If you're using SIP secured or Secured dial plans, you must create a UM IP gateway with an FQDN and use the Shell to configure it to listen on port 5061. For existing UM IP gateways, verify that they're configured with an FQDN and are listening on port 5061. If the UM IP gateway doesn't use an FQDN, use the EAC or the Shell to change the address. If the UM IP gateway doesn't use port 5061, use the Shell to change the port. You can view the settings of a UM IP gateway by using the **Get-UMIPGateway** cmdlet.

A UM IP gateway represents a physical Voice over IP (VoIP) gateway, IP PBX, or SIP-enabled PBX. Before a VoIP gateway, IP PBX, or SIP-enabled PBX can be used to answer incoming calls and send outgoing calls for voice mail users, a UM IP gateway must be created in the directory service.

The combination of the UM IP gateway and a UM hunt group establishes a link between a VoIP gateway, IP PBX, or SIP-enabled PBX and a UM dial plan. By creating multiple UM hunt groups, you can associate a single UM IP gateway with multiple UM dial plans. For more information, see UM IP gateways.

If required, you can create a UM IP gateway by using the EAC, as follows:

1. In the EAC, navigate to **Unified Messaging** > **UM IP gateways**, and then click **Add +**.
2. On the **New UM IP Gateway** page, enter the following information:
   - **Name** Use this box to specify a unique name for the UM IP gateway. This is a display name that appears in the EAC. If you have to change the display name of the UM IP gateway after it's been created, you must first delete the existing UM IP gateway, and then create another UM IP gateway that has the appropriate name. The UM IP gateway name is required, but it's used for display purposes only. Because your organization may use multiple UM IP gateways, we recommend that you use meaningful names for your UM IP gateways. The maximum length of a UM IP gateway name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
   - **Address** You can configure a UM IP gateway with either an IPv4 or IPv6 address or an FQDN. Use this box to specify the IP address or FQDN configured on the VoIP gateway, IP PBX, or SIP-enabled PBX. This box accepts only FQDNs that are valid and formatted correctly.

You can enter alphabetical and numeric characters. IPv4 addresses, IPv6 addresses, and FQDNs are supported. If you want to use mutual TLS between a UM IP gateway and a dial plan operating in either SIP secured or Secured mode, you must configure the UM IP gateway with an FQDN. You must also configure it to listen on port 5061 and verify that any VoIP gateways or IP PBXs have also been configured to listen for mutual TLS requests on port 5061. To configure a UM IP gateway, run the following command in the Shell: `Set-UMIPGateway -identity MyUMIPGateway -Port 5061`

If you use an FQDN, you must also make sure that you've correctly configured a DNS host record for the VoIP gateway, IP PBX, or SIP-enabled PBX so that the host name will be correctly resolved to an IP address. Also, if you use an FQDN instead of an IP address, and the DNS configuration for the UM IP gateway is changed, you must disable and then enable the UM IP gateway to make sure that configuration information for the UM IP gateway is updated correctly.

   - **UM Dial Plan** Click **Browse** to select the UM dial plan that you want to associate with the UM IP gateway. When you select a UM dial plan to associate with a UM IP gateway, a default UM hunt group is also created and associated with the UM dial plan that you selected. If you don't select a UM dial plan, you must manually create a UM hunt group and then associate that UM hunt group with a UM IP gateway that you have created.
3. Click **Save**.

If required, you can create a UM IP gateway by run the following command.

```
New-UMIPGateway -Identity MyUMIPGateway -Address
"MyUMIPGateway.contoso.com"
```

To configure an existing UM IP gateway by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM IP gateways**, and then click **Edit** ✎.
2. On the **UM IP gateway** page, click **Configure**. Use the configure options to view specific UM IP gateway settings and to enable or disable features.

To configure an existing UM IP gateway in the Shell, running the following command in the Shell.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
```

```
fe80::39bd:88f7:6969:d223%11 -IPAddressFamily Any -Status
Disabled -OutcallsAllowed $false
```

Step 9: Create a UM hunt group

Depending on your existing Exchange 2010 deployment, you may be required to create new UM hunt groups. A telephony hunt group provides a way to distribute telephone calls from a single number to multiple extensions or telephone numbers. In Unified Messaging, a UM hunt group is a logical representation of a telephony hunt group, and it links a UM IP gateway to a UM dial plan.

You need to have at least one UM hunt group for every IP PBX or PBX hunt group. When you complete the following procedure, one UM hunt group is created by default. If you have more than one IP PBX or PBX hunt group, you need to create additional UM hunt groups. To learn more about UM hunt groups, see UM hunt groups.

If required, you can To create a UM hunt group by using the EAC, as follows:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then click **Edit** ✎.
2. On the **UM Dial Plan** page, under **UM Hunt Groups**, click **New +**.
3. On the **New UM Hunt Group** page, enter the following information:
   - **Name** Use this box to create the display name for the UM hunt group. A UM hunt group name is required and must be unique, but it's used only for display purposes in the EAC and the Shell. If you have to change the display name of the hunt group after it's been created, you must first delete the existing hunt group and then create another hunt group that has the appropriate name. If your organization uses multiple hunt groups, we recommend that you use meaningful names for your hunt groups. The maximum length of a UM hunt group name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
   - **UM IP Gateway** Use this box to select a UM IP gateway. This box shows the name of the UM IP gateway that will be linked with the UM hunt group. To link a UM IP gateway to the UM hunt group, click **Browse**.
   - **Pilot Identifier** Use this box to specify a string that uniquely identifies the pilot identifier configured on the PBX or IP PBX. An extension number or a SIP Uniform Resource Identifier (URI) can be used in this box. Alphanumeric characters are accepted in this box. For legacy PBXs, a numeric value is used as a pilot identifier. However, some IP PBXs and SIP-enabled PBXs can use SIP URIs.
4. Click **Save**.

If required, you can create a UM hunt group by running the following command in the Shell.

```
New-UMHuntGroup -Name MyUMHuntGroup -PilotIdentifier
5551234,55555 -UMDialPlan MyUMDialPlan -UMIPGateway
MyUMIPGateway
```

Tip:

You can't configure or change settings for a UM hunt group. If you want to change the configuration settings for a UM hunt group, you must delete it and add a new UM hunt group with the correct settings.

Step 10: Create or configure UM auto attendants

Depending on your existing Exchange 2010 deployment, you may be required to create new UM auto attendants. You can use UM auto attendants to create a voice menu system that lets external and internal callers use the UM auto attendant menu system to locate people and place or transfer calls to company users or departments in an organization. For more information, see Automatically answer and route incoming calls.

In smaller deployments you may only want to deploy UM so that callers can leave voice mail for users. In these deployments, creating an auto attendant isn't required. However, in most cases, using auto attendants is very useful for external callers when they call in to your organization.

If required, you can create a UM auto attendant by using the EAC, as follows:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. Select the UM dial plan for which you want to add an auto attendant, and then click **Edit** ✎.
2. On the **UM dial plan** page, under **UM Auto Attendants**, click **Add +**.
3. On the **New UM Auto Attendant** page, complete the following boxes:
   - **Name** Use this box to create the display name for the UM auto attendant. A UM auto attendant name is required and must be unique. However, it's used only for display purposes in the EAC and the Shell. If you have to change the display name of the auto attendant after it's created, you must first delete the existing UM auto attendant and then create another auto attendant that has the appropriate name. If your organization uses multiple UM auto attendants, we recommend that you use meaningful names for your UM auto attendants. The maximum length of a UM auto attendant name is 64 characters, and it can include spaces.

Although you can name a new UM auto attendant to include spaces, if you integrate Unified Messaging with Lync Server, the name of the auto attendant can't include spaces. Therefore, if you created an auto attendant that has spaces in the display name and you're integrating with Lync Server, you must first delete that auto attendant and then create another auto attendant that doesn't include spaces in the display name.

   - **Create this auto attendant as enabled** Select this check box to enable the auto attendant to answer incoming calls when you finish creating the UM auto attendant. By default, a new auto attendant is created as disabled. If you decide to create the UM auto attendant as disabled, you can use the EAC or the Shell to enable the auto attendant after you finish creating it.
   - **Set the auto attendant to respond to voice commands** Select this check box to speech-enable the UM auto attendant. If the auto attendant is speech-enabled, callers can respond to the system or custom prompts used by the UM auto attendant using touchtone or voice inputs. By default, the auto attendant isn't speech-enabled when it's created. For callers to use a speech-enabled auto attendant in a language other than U.S. English (en-US), you must install

the appropriate UM language pack and configure the properties of the auto attendant to use this language. The en-US UM language pack is installed by default when you install an Exchange 2013 Mailbox server.

- **Access numbers** Enter the extension or telephone numbers that callers will use to reach the auto attendant. Type an extension number or telephone number in the box, and then click **Add** to add the number to the list. The number of digits in the extension number or telephone number that you provide doesn't have to match the number of digits for an extension number configured on the associated UM dial plan. This is because direct calls are allowed to UM auto attendants.

The number of extension or access numbers you can enter is unlimited. However, you may create a new auto attendant without listing an extension number or telephone number. An extension number or telephone number isn't required. You can edit or remove an existing extension number or pilot identifier. To edit an existing extension number or telephone number, click **Edit**. To remove an existing extension number or telephone number from the list, click **Remove**.

4. Click **Save**.

If required, you can create a UM auto attendant by running the following command in the Shell.

```
New-UMAutoAttendant -Name MyUMAutoAttendant -UMDialPlan
MyUMDialPlan -PilotIdentifierList 56000,56100 -
SpeechEnabled $true -Status Enabled
```

If required, you can configure an existing auto attendant by using the EAC, as follows:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**, and then click **Edit** ✎.
2. On the **UM dial plan** page, under **UM Auto Attendants**, select the UM auto attendant that you want to view or configure, and then click **Edit** ✎. Use the configuration options to view specific auto attendant settings and to enable or disable features.

If required, you can configure an existing auto attendant by running the following command in the Shell.

```
Set-UMAutoAttendant -Identity MySpeechEnabledAA -
DTMFFallbackAutoAttendant MyDTMFAA -OperatorExtension 50100
-AfterHoursTransferToOperatorEnabled $true -
StaroutToDialPlanEnabled $true
```

Step 11: Create or configure UM mailbox policies

Depending on your existing Exchange 2010 deployment, you may be required to create new UM mailbox policies or configure existing UM mailbox policies. UM mailbox policies are required when you enable users for Unified Messaging. The mailbox of each UM-enabled user must be linked to a single UM mailbox policy. After you create a UM mailbox policy, you link one or more UM-enabled mailboxes to the UM mailbox policy. This lets you control PIN security settings such as the minimum number of digits in a PIN or the maximum number of logon attempts for the UM-enabled users

who are linked to the UM mailbox policy. For more information, see UM mailbox policies.

If required, you can create a UM mailbox policy by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan that you want to modify, and then click **Edit** .
2. On the **UM dial plan** page, under **UM Mailbox Policies**, click **Add** .
3. On the **New UM Mailbox Policy** page, in the **Name** box, enter the name of the new UM mailbox policy.

Note:

Use this box to specify a unique name for the UM mailbox policy. This is a display name that appears in the EAC. If you must change the display name of the UM mailbox policy after it's been created, you must first delete the existing UM mailbox policy, and then create another UM mailbox policy that has the appropriate name. You can't delete a UM mailbox policy if any UM-enabled users are associated with it.The UM mailbox policy name is required, but it's used for display purposes only. Because your organization may use multiple UM mailbox policies, we recommend that you use meaningful names for your UM mailbox policies. The maximum length of a UM mailbox policy name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

4. Click **Save**.

Note:

When you save the UM mailbox policy, all the default settings including PIN policies, voice mail features, and Protected Voice Mail settings are enabled. If you want to customize or change any default settings for the UM mailbox policy you just created, use the **Set-UMMailbox** cmdlet or the EAC.

If required, you can create a UM mailbox policy by running for following command in the Shell.

```
New-UMMailboxPolicy -Name MyUMMailboxPolicy -UMDialPlan
MyUMDialPlan
```

If required, you can configure an existing UM mailbox policy by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**, and then click **Edit** .
2. On the **UM dial plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to view or configure, and then click **Edit** . Use the configuration options to view specific UM mailbox policy settings and to enable or disable features.

If required, you can configure an existing UM mailbox policy by running the following command in the Shell.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
LogonFailuresBeforePINReset 8 -MaxLogonAttempts 12 -
MinPINLength 8 -PINHistoryCount 10 -PINLifetime 60 -
ResetPINText "The PIN used to allow you access to your
mailbox using Outlook Voice Access has been reset."
```

Step 12: Move existing UM-enabled mailboxes to Exchange 2013

In Exchange 2010 Unified Messaging, after you've enabled users within the organization to use voice mail, a default set of UM properties is applied to the user so they can use UM features. Learn more at Voice mail for users.

During the process of upgrading, there will be a period of time during which you'll have mailboxes that are UM enabled both on Exchange 2010 Mailbox servers and on Exchange 2013 Mailbox servers. However, if you're moving all UM-enabled users over to Exchange 2013 Mailbox servers, you must use the EAC or the **New-MoveRequest** cmdlet in the Shell from an Exchange 2013 server to retain all of the properties and settings, including the user's PIN.

A move request is the process of moving a mailbox from one mailbox database to another. A local move request is a mailbox move that occurs within a single forest. For more information about mailbox moves, see:

- Mailbox moves in Exchange 2013
- New-MoveRequest
- New-MigrationBatch
- Managing Move Requests

To move an Exchange 2010 mailbox to an Exchange 2013 Mailbox server by using the EAC:

1. In the EAC, click **Recipients** > **Migration**, and then click **Add** .
2. In the **New local mailbox move** wizard, select the user you want to move, click **OK**, and then click **Next**.
3. On the **Move configuration** page, specify a name for the new batch. Select which options you want for the archive mailbox and the mailbox database location, and then click **New**.

To move an Exchange 2010 mailbox to an Exchange 2013 Mailbox server by using the Shell, run the following command.

```
New-MoveRequest -Identity 'tony@alpineskihouse.com' -
TargetDatabase "DB01"
```

Step 13: Enable new users for UM or configure settings for an existing UM-enabled user

A user must have a mailbox before they can be enabled for Unified Messaging. But, by default, a user who has a mailbox isn't enabled for UM. After the user is UM-enabled, you can manage, modify, and configure the UM properties and voice mail features for them. You can enable a user for UM by using the EAC or the Shell. Learn more at Voice mail for users.

When you enable a user for UM, you must define at least one extension number that UM will use

when voice mail is submitted to the user's mailbox and to allow the user to use Outlook Voice Access. After you enable the user for UM, you can add secondary extension numbers to the user's mailbox, modify or remove them by configuring the Exchange Unified Messaging (EUM) proxy address on the user's mailbox, or add or remove additional or secondary extensions for the user in the EAC. To add, modify, or remove extension numbers, E.164 numbers, or SIP addresses, see Voice mail-enabled user procedures.

To enable a user for Unified Messaging by using the EAC:

1. In the EAC, click **Recipients**.
2. In the list view, select the user whose mailbox you want to enable for Unified Messaging.
3. In the Details pane, under **Phone and Voice Features**, click **Enable**.
4. On the **Enable UM mailbox** page, click the **Browse** button next to **UM mailbox policy**, locate the UM mailbox policy to assign the user from the list, and then click **OK**.
5. On the **Enable UM mailbox** page, complete the following boxes:
   - **SIP address or E.164 number** Enter the SIP address or E.164 number for the user. These options are available if the user that you enable for Unified Messaging is assigned to a UM mailbox policy that's linked to either a SIP URI or an E.164 dial plan. Adding a SIP address or E.164 number for a user isn't available if the user is associated with a telephone extension dial plan. When you assign the user to a UM mailbox policy that's linked to a SIP URI or E.164 dial plan, you must still also enter an extension number for the user. This extension number is used when users access their mailbox using Outlook Voice Access. The number of digits that you configure in this box must match the number of digits configured on the SIP URI or E.164 dial plan.
   - **Extension number** Enter the extension number for the user you're enabling for UM.

You must provide a valid extension number for the user, and it must match the number of digits specified on the dial plan. You can only enter numeric characters or digits from 1 through 20. The typical extension number is 3 to 7 digits long. The number of digits in the extension is set on the dial plan that's linked to the UM mailbox policy that's assigned to the user.

   - Under **PIN settings**, complete the following:
     - **Automatically generate PIN** Click this button to automatically generate a PIN for the UM-enabled user to use for voice mail access via Outlook Voice Access. This is the default setting. When you click this button, a PIN is automatically generated based on the PIN policies configured on the UM mailbox policy assigned to the user. We recommend that you use this setting to help protect the user's PIN. The PIN is sent to the user in the welcome message they receive after they're enabled for UM. By default, they'll have to change this PIN when they first sign in to their mailbox to get their voice mail.
     - **Type a PIN** Click this button to manually specify a PIN that the user will use to access the voice mail system.The PIN must comply with the PIN policy settings configured on the UM mailbox policy associated with this UM-enabled user. For example, if the UM mailbox policy is configured to accept only PINs that contain seven or more digits, the PIN you enter in this box must be at least seven digits long.
     - **Require the user to reset their PIN the first time they sign in** Select this check box to

force the user to reset their voice mail PIN when they access the voice mail system from a telephone using Outlook Voice Access for the first time. They will be prompted to enter a PIN that's more familiar to them.It's a security best practice to force UM-enabled users to change their PIN when they first sign in to help protect against unauthorized access to their data and Inbox. This check box is selected by default.

6. On the **Enable UM mailbox** page, review your settings. Click **Finish** to enable the user for Unified Messaging. Click **Back** to make configuration changes.

To enable a user for Unified Messaging by using the Shell, run the following command.

```
Enable-UMMailbox -Identity tonysmith@contoso.com -
UMMailboxPolicy MyUMMailboxPolicy -Extensions 51234 -PIN
5643892 -NotifyEmail administrator@contoso.com -PINExpired
$true
```

If required, you can configure a user that's been enabled for UM by using the EAC:

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox for which you want to change the UM mailbox policy.
3. In the details pane, under **Phone and Voice Features** > **Unified Messaging**, click **View details**.
4. On the **UM Mailbox** page, click **UM mailbox settings** to view or change the following UM properties for an existing UM-enabled user:
   - **PIN Status** This display-only box shows the status of the user's mailbox. By default, when a user is UM-enabled, the PIN status is listed as **Not locked out**. However, if the user has input an incorrect Outlook Voice Access PIN multiple times, the status is listed as **Locked Out**.
   - **UM mailbox policy** This box shows the name of the UM mailbox policy associated with the UM-enabled user. You can click **Browse** to locate and specify the UM mailbox policy to be associated with this UM mailbox.
   - **Personal operator extension** Use this box to specify the operator extension number for the user. By default, an extension number isn't configured. The length of the extension number can be from 1 through 20 characters. This enables incoming calls for the UM-enabled user to be forwarded to the extension number that you specify in this box.

You can configure other types of operator extension numbers on dial plans and auto attendants. However, those extensions are generally meant for company-wide receptionists or operators. The personal operator extension setting could be used when an administrative assistant or personal assistant answers incoming calls for a specific user.

5. On the **UM Mailbox** page, under **Other extensions**, you can add, change, and view extension numbers for the user.
   - To add an extension number, click **Add +**. On the **Add another extension** page, use **Browse** to select the UM dial plan, and then enter the extension number in the **Extension number** box.
   - To remove an extension number, select the extension number you want to remove, and then click **Remove −**.
6. If you make any changes, click **Save**.

If required, you can configure a user that's been enabled for UM in the Shell by running the

following command.

```
Set-UMMailbox -Identity tony@contoso.com -
CallAnsweringAudioCodec Wma -CallAnsweringRulesEnabled
$false -FaxEnabled $false -UMSMSNotificationOption
VoiceMail
```

Step 14: Configure your VoIP gateways, IP PBXs, and SIP-enabled PBXs to send all incoming calls to the Exchange 2013 Client Access servers

When Exchange 2013 Client Access and Mailbox servers are installed, they're automatically enabled so they can answer incoming and outgoing voice calls and route voice mail messages to the intended recipients. When you're installing your Exchange 2013 Client Access and Mailbox servers and deploying Unified Messaging, you don't have to link or add Exchange 2013 Client Access or Mailbox servers to UM dial plans. Exchange 2013 Client Access and Mailbox servers answer all incoming calls and then use the UM dial plans to locate users.

The Exchange 2013 Client Access server is the entry point for any inbound calls or Session Initiation Protocol (SIP) requests for Unified Messaging. The service that handles the SIP requests on an Exchange 2013 Client Access server is the UM Call Router service and it runs on each Exchange 2013 Client Access server in your organization.

When you're upgrading to Exchange 2013 UM, you should have already installed and configured single or multiple VoIP gateways to connect to the PBXs in your telephony network, or installed and configured Session Initiation Protocol (SIP)-enabled PBXs or IP PBXs.

The last step in the process of upgrading to Exchange 2013 UM is to configure the VoIP gateways, IP PBXs, or SIP-enabled PBXs to send incoming calls—including callers who want to leave voice mail for a user, calls from UM-enabled users calling in to Outlook Voice Access, and calls from callers that dial in to a UM auto attendant—to your Exchange 2013 Client Access servers. All these calls are received first by a VoIP gateway, IP PBX, or SIP-enabled PBX and then forwarded on to the Exchange 2013 Client Access servers in your Exchange 2013 organization. For more information, see the following resources:

UM services

Configuration notes for supported VoIP gateways, IP PBXs, and PBXs

Telephony advisor for Exchange 2013

Step 15: Disable call answering on an Exchange 2010 Unified Messaging server

You can disable call answering by disabling UM on an Exchange 2010 Unified Messaging server or by removing the UM server from a dial plan. When you disable UM, you prevent the UM server from answering incoming calls. You can choose to disconnect all calls immediately or wait for existing calls to be processed before disabling the Unified Messaging server. You'll want to disable call answering before removing the server from a dial plan so that it will finish processing any incoming calls.

To disable Unified Messaging on an Exchange 2010 UM server by using the Exchange Management Console:

1. In the console tree of the EMC, navigate to **Server Configuration** > **Unified Messaging**.
2. In the result pane, select the Unified Messaging server to disable.
3. In the action pane, click one of the following:
   - When you select the **Disable Immediately** option, the Unified Messaging server disconnects all calls connected to the Unified Messaging server.
   - When you select the **Disable After Completing Calls** option, the Unified Messaging server won't accept new calls and won't be disabled until all calls have been processed.
4. In the confirmation dialog box, click **Yes** to continue.

To disable Unified Messaging on an Exchange 2010 UM server by using the Shell, run the following command.:

```
Disable-UMServer -Identity MyUMServer -Immediate $true
```

Tip:

You can use the **Disable-UMServer** cmdlet from an Exchange 2010 UM server or the **Disable-UMService** cmdlet from an Exchange 2013 Mailbox server to disable call answering.

Step 16: Remove an Exchange 2010 Unified Messaging server from a dial plan

To process calls, an Exchange 2010 UM server must be added to at least one UM dial plan. A UM server can be added to multiple UM dial plans. You can remove an Exchange 2010 UM server from a UM dial plan. When you remove a UM server from a dial plan, the UM server will no longer answer calls or process UM calls for UM-enabled users.

To remove an Exchange 2010 UM server from a dial plan by using the Exchange Management Console:

1. In the console tree of the EMC, navigate to **Server Configuration** > **Unified Messaging**.
2. In the result pane, select the Unified Messaging server.

3. In the action pane, click **Properties**.
4. On the **UM Settings** tab, in the **Associated Dial Plans** section, click **Remove**.
5. In the confirmation dialog box, click **Yes** to confirm the deletion of the Exchange 2010 server from the UM dial plan.
6. Click **OK** to close the properties window.

To remove an Exchange 2010 UM server from a dial plan using the Shell, run the following command.

```
$dp= Get-UMDialPlan "MySIPDialPlan"
$s=Get-UMServer -id MyUMServer
$s.dialplans-=$dp.identity
Set-UMServer -id MyUMServer -dialplans:$s.dialplans
```

In this example, there are three SIP URI dial plans: SipDP1, SipDP2 and SipDP3. This example removes the UM server named `MyUMServer` from the SipDP3 dial plan.

```
Set-UMServer -id MyUMServer -DialPlans SipDP1,SipDP2
```

In this example, there are two SIP URI dial plans: SipDP1 and SipDP2. This example removes the UM server named `MyUMServer` from the SipDP2 dial plan.

```
Set-UMServer -id MyUMServer -DialPlans SipDP1
```

Tip:

You can use the **Set-UMServer** cmdlet in the Shell on an Exchange 2010 Unified Messaging server or the **Set-UMService** cmdlet on an Exchange 2013 Mailbox server to remove an Exchange 2010 UM server from a single or multiple dial plans. For example, to remove a UM server from all dial plans, run the following command: `Set-UMServer -identity MyUMServer -DialPlans $null`

How do you know this worked?

After you've set up Unified Messaging, verify the following to ensure it's working correctly:

- A user you've enabled for voice mail can sign in to Outlook Web App or Outlook and see a Welcome Message for Unified Messaging.
- UM users can receive voice messages.
- UM users can call in to an Outlook Voice Access number to listen to email, calendar items, and voice mail.
- UM is routing calls from outside of your organization, and you can place a call.

Checklist: Upgrade Exchange 2010 UM

to Exchange 2013 UM

Topic Last Modified: *2014-04-04*

Use this checklist to help you upgrade Exchange 2010 Unified Messaging (UM) to Exchange 2013 UM. Be sure to refer to this information when you're upgrading your Exchange 2010 organization and your UM deployment to Exchange 2013. For step-by-step instructions for upgrading to Exchange 2013 UM, see Upgrade Exchange 2010 UM to Exchange 2013 UM.

Before you start working with this checklist, make sure you're familiar with the concepts in:

- Planning for Unified Messaging
- Telephone system integration with UM
- Connect your voice mail system to your telephone network

For step-by-step guidance about how to upgrade from Exchange 2007 UM to Exchange 2013 UM, see Upgrade Exchange 2007 UM to Exchange 2013 UM.

Checklist for upgrading Exchange 2010 UM to Exchange 2013 UM

Done?	Tasks	Topic
	Deploy and configure telephony components.	Connect UM to your telephone system
	Review the system requirements before installing Exchange 2013.	Exchange 2013 system requirements
	Verify that you meet the prerequisites for installation.	Exchange 2013 prerequisites
	Install the required Client Access and Mailbox servers. **⚠Warning:** You must deploy at least one Exchange 2013 Mailbox server in your organization before you configure the VoIP gateways or IP PBXs to send UM SIP and RTP	Install Exchange 2013 using the Setup wizard

	traffic to the Exchange 2013 Client Access servers.	
	Verify the installation and review the server setup logs.	Verify an Exchange 2013 installation
	If required, install the required UM language packs.	Install a UM language pack
	Move the Exchange 2010 system mailbox used for UM custom prompts to Exchange 2013.	Mailbox moves in Exchange 2013 **Note:** If the system mailbox has already been moved, you can still manually import/export custom prompts from Exchange 2010 using Import and export custom greetings, announcements, menus, and prompts.
	Export and import certificates.	Deploying certificates for UM
	Configure the UM startup mode on all Exchange 2013 Client Access servers.	Configure the startup mode on a Client Access server
	Configure the UM startup mode on all Exchange 2013 Mailbox servers.	Configure the startup mode on a Mailbox server
	Create or configure existing UM dial plans.	Create a UM dial plan Manage a UM dial plan
	Create or configure existing UM IP gateways.	Create a UM IP gateway Manage a UM IP gateway
	Create a UM hunt group.	Create a UM hunt group
	Create or configure UM auto	Create a UM auto attendant

	attendants.	Manage a UM auto attendant
	Create or configure UM mailbox policies.	Create a UM mailbox policy Manage a UM mailbox policy
	Move existing UM-enabled mailboxes to Exchange 2013.	Mailbox moves in Exchange 2013
	Enable new users for UM or configure settings for an existing UM-enabled user.	Enable a user for voice mail Manage voice mail settings for a user
	Configure your VoIP gateways, IP PBXs, and SIP-enabled PBXs to send all incoming calls to the Exchange 2013 Client Access servers.	Configuration notes for supported VoIP gateways, IP PBXs, and PBXs Connect a VoIP gateway, IP PBX, or session border controller to UM
	Disable call answering on the Exchange 2010 UM servers.	Disable Unified Messaging on Exchange 2010
	Remove an Exchange 2010 UM server from a dial plan.	Remove a UM Server from a Dial Plan

Upgrade Exchange 2007 UM to Exchange 2013 UM

Exchange Server 2013 > Unified Messaging > Deploying voice mail and UM >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-19*

When you're upgrading a Microsoft Exchange 2007 organization with Unified Messaging (UM) to Exchange 2013 Unified Messaging, there are steps that are required and other steps that were already completed as part of your Exchange 2007 UM deployment. Depending on your telephony environment and the UM components that were created and configured to support Unified

Messaging in Exchange 2007, you may need to deploy additional telephony equipment including Voice over IP (VoIP) gateways, IP Private Branch eXchanges (PBXs), or traditional or SIP-enabled PBXs and then create and configure any additional UM components that will be required for Exchange 2013 UM.

What do you need to know before you begin?

- Estimated time to complete this task: 45-90 minutes.
- Verify that you have the appropriate permissions in the Exchange 2007 and Exchange 2013 organization to create and configure all the required components.
- Verify that you've deployed and correctly configured your telephony components, including VoIP gateways and PBXs, IP PBXs, or Session Initiation Protocol (SIP)-enabled PBXs.
- Verify that you've correctly installed and configured the Client Access servers running the Microsoft Exchange Unified Messaging Call Router (UM Call Router) service and Mailbox servers running the Microsoft Exchange Unified Messaging (UM) service. To learn more about UM services, see UM services.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

How do you do this?

Step 1: Download and install the required UM language packs

UM language packs enable callers and Outlook Voice Access users to interact with the voice mail system in multiple languages. After you install an additional language on an Exchange 2013 Mailbox server, callers and Outlook Voice Access users can hear email messages and interact with the voice mail system in that language. However, to make the language available for all incoming calls, you must install the required UM language packs on all Exchange 2013 Mailbox servers. This is because every Exchange 2013 Mailbox server can answer incoming calls for Unified Messaging.

By default, when you install an Exchange 2013 Mailbox server, the U.S. English (en-US) language pack is installed. It's the only available language option for your dial plan unless you install another UM language pack. (U. S. English can't be removed unless you remove the Mailbox server from the computer.) After you install a UM language pack on an Exchange 2013 Mailbox server, the language associated with the language pack will be listed as an available option when you configure the default language for the dial plan. By default, because a UM auto attendant is linked to a UM dial plan when the auto attendant is created, it uses the default language setting of the

linked UM dial plan. However, this setting can be changed after the UM auto attendant is created.

Note:
If U.S. English is the only language that you want to provide for your dial plan, you can skip this step and go to step 2.

You can add UM language packs by using the setup.exe command or by running the *<UMLanguagePack>*.exe installation program after you've downloaded the UM language pack from Exchange Server 2013 UM Language Packs. For more information, see Install a UM language pack.

This example uses setup.exe to install the Japanese (ja-JP) UM language pack.

```
setup.exe /AddUmLanguagePack:ja-JP /s:d:\Exchange
\UMLanguagePacks /IAcceptExchangeServerLicenseTerms
```

Step 2: Move the Exchange 2007 custom greetings, announcements, menus, and prompts to the Exchange 2013 system mailbox

Custom greetings, announcements, menus, and prompts are used by Unified Messaging dial plans and auto attendants. The system mailbox named {e0dc1c29-89c3-4034-b678-e6c29d823ed9} is created when you install Exchange 2007 or Exchange 2013 and is used to support features such as Message Approval and Multi-Mailbox Search. This system mailbox is also used to store dial plan and auto attendant custom greetings, announcements, menus, and prompts. If the system mailbox doesn't exist, you can use the **Setup /PrepareAD** command to create it.

By default, system mailboxes aren't visible in the Exchange admin center (EAC). You can get a list of the system mailboxes by running one the following:

This command returns a list of all the system mailboxes.

```
Get-Mailbox -Arbitration
```

This command returns a list of system mailboxes and their individual properties or settings.

```
Get-Mailbox -Arbitration |fl
```

When you're importing custom greetings, announcements, menus, and prompts from Exchange 2007 to Exchange 2013, you must use the MigrateUMCustomPrompts.ps1 script. You can't use the EAC to import custom greetings, announcements, menus, and prompts. The MigrateUMCustomPrompts.ps1 script migrates a copy of all Exchange Server 2007 UM custom greetings, announcements, menus, and prompts to Exchange 2013 UM. By default, the MigrateUMCustomPrompts.ps1 script is located in the *<Program Files>*\Microsoft\Exchange Server

\V15\Scripts folder on an Exchange 2013 Mailbox server and must be run from an Exchange 2013 Mailbox server. To run the script:

1. Click **Start** > **All Programs** > **Microsoft Exchange Server 2013** > **Exchange Management Shell**.
2. In the Exchange Management Shell, at the prompt, type the path to the script. For example, type **cd "D:\Program Files\Microsoft\Exchange Server\V15\Scripts"**, and then press Enter.
3. At the Shell prompt, type **".\MigrateUMCustomPrompts"**, and then press Enter.

Note:
Custom prompts can also be imported individually by using the **Import-UMPrompt** cmdlet. The Exchange 2007 UM **Copy-UMCustomPrompt** cmdlet isn't supported for copying custom prompts to Exchange 2013 UM.

When you run the MigrateUMCustomPrompts.ps1 script from your Exchange 2013 server, the script performs a GUID or object identifier lookup for the dial plan or auto attendant in Active Directory and queries it to determine if there are any custom greetings, announcements, menus, or prompts. If found, the custom greetings, announcements, menus, and prompts will be imported into the system mailbox named {e0dc1c29-89c3-4034-b678-e6c29d823ed9}.

By using this system mailbox, custom greetings, announcements, menus, and prompts can be backed up and restored along with other mailboxes in a database. This reduces the amount of resources that are needed. Storing custom greetings, announcements, menus, and prompts in a system mailbox removes any possible inconsistencies that may have occurred. To learn more about mailbox moves, see Mailbox moves in Exchange 2013.

Step 3: Export and import certificates

If you're using SIP secured or Secured dial plans in your Exchange 2007 organization, you'll need to export and import the certificates that were used to your Exchange 2013 Client Access and Mailbox servers. Mutual Transport Layer Security (mutual TLS) is used to encrypt data sent between your Exchange 2013 servers and the VoIP gateways, IP PBXs, and SIP-enabled PBXs. Certificates bind the identity of the certificate owner to a pair of electronic keys (public and private) that are used to encrypt and sign information digitally. You can use one of the following certificates for the UM and UM Call Router services:

- A self-signed (Exchange) certificate
- An internal public key infrastructure (PKI) certificate
- A third-party commercial certificate

By default, when you install Exchange 2013, two self-signed certificates are created: **Microsoft Exchange Server Auth Certificate** and **Microsoft Exchange**. The **Microsoft Exchange** self-signed certificate can be used for UM to encrypt data, but you must assign the certificate to the UM and UM Call Router services. This self-signed certificate can be copied and then imported on the VoIP gateways, IP PBXs, and SIP-enabled PBXs. However, it can't be used when you're integrating UM with Microsoft Lync Server.

To enable UM to encrypt data that's sent between your Exchange 2013 servers and VoIP gateways, IP PBXs, and SIP-enabled PBXs, you need to do the following:

- Use an existing self-signed UM certificate, create a new self-signed Exchange certificate, submit a certificate request to an internal certification authority for a PKI certificate, or purchase a third-party commercial certificate that you can use for mutual TLS between your Exchange 2013 Mailbox and Client Access servers and VoIP gateways, IP PBXs, and SIP-enabled PBXs.

Create an Exchange self-signed certificate by using the EAC, as follows:

1. In the EAC, navigate to **Servers** > **Certificates**, and then click **Add +**.
2. On the **New Exchange certificate** page, choose **Create a self-signed certificate**, and then select **Next**.
3. Enter a friendly name for the certificate, and then select **Next**.
4. Click **Add +** to select the Exchange servers that you want to apply this certificate to, and then select **Next**.
5. Specify the domains that you want to be included in your certificate, and then select **Next**. If you want to add a domain for a service, click **Edit** .
6. Verify that the domains you included are correct, and then select **Finish**.

Important:

When you use the EAC to create a certificate, you won't be prompted to enable the services for the certificate. After the certificate has been created, you can use the EAC to enable the services. For more information about how to enable a certificate for services, see Assign a certificate to the UM and UM Call Router services.

Create an Exchange self-signed certificate by running the following command in the Shell.

```
New-ExchangeCertificate -Services 'UM, UMCallRouter' -
DomainName '*.northwindtraders.com' -FriendlyName
'UMSelfSigned' -SubjectName
'C=US,S=WA,L=Redmond,O=Northwindtraders,OU=Servers,CN=
Northwindtraders.com' -PrivateKeyExportable $true
```

Tip:

If you specify the services you want to enable by using the *Services* parameter, you will be prompted to enable the services for the certificate you created. In this example, you will be prompted to enable the certificate for the Unified Messaging and Unified Messaging Call Router services. For more information about how to enable a certificate for services, see Assign a certificate to the UM and UM Call Router services.

- Import the certificate that will be used on all Exchange 2013 Client Access and Mailbox servers in your organization. If you use the Exchange 2013 self-signed certificate, you'll need to copy the certificate, then import it on the VoIP gateways, IP PBXs, or SIP-enabled PBXs. If you use the self-signed certificate from Exchange 2007, the Subject Alternative Name (SAN) must contain the machine names of all the Exchange 2013 servers. If you have Exchange 2007 Unified Messaging servers in your organization, you can use the Exchange 2013 self-signed certificate, but you must add the machine names of the Exchange 2007 UM servers to the SAN in the Exchange 2013

certificate.

- Enable or assign the certificate to be used to the UM and UM Call Router services on the Client Access and Mailbox servers in your organization.

Enable the UM service and the UM Call Router service on all Exchange 2013 servers to use the Exchange self-signed certificate by using the EAC, as follows:

1. In the EAC, navigate to **Servers** > **Certificates**, select the certificate you want to enable services on, and then click **Edit** ✎.
2. On the **Procedure** page, select **Services**, select **Unified Messaging**, and then select **Unified Messaging call router**.

Enable an Exchange self-signed certificate by running the following command in the Shell.

```
Enable-ExchangeCertificate -Thumbprint
5113ae0233a72fccb75b1d0198628675333d010e -Services 'UM,
UMCallRouter'
```

- Configure any new or existing UM dial plans as SIP secured or Secured.
- Configure the UM startup mode to TLS or Dual on the Client Access and Mailbox servers in your organization.
- Create and configure new or existing UM IP gateways with a fully qualified domain name (FQDN).
- Configure the listening port on the UM IP gateways to use TLS port 5061.
- Restart the UM Call Router service on all Exchange 2013 Client Access servers and restart the UM service on all Exchange 2013 Mailbox servers. To learn more about UM services, see UM services.

Step 4: Configure the UM startup mode on all Exchange 2013 Client Access servers

If you're using SIP secured or Secured dial plans, you must configure the UM startup mode on your Exchange 2013 Client Access servers. You can specify the UM startup mode for the UM Call Router service on an Exchange 2013 Client Access server by using the EAC or the Shell. By default, the Exchange 2013 Client Access server will start up in TCP mode, but if you're using Transport Layer Security (TLS) to encrypt Voice over IP (VoIP) traffic, you must configure the Exchange 2013 Client Access server to use TLS or Dual mode. We recommend that all Exchange 2013 Client Access servers be configured to use Dual as the UM startup mode. This is because all Exchange 2013 Client Access servers can answer incoming calls for all UM dial plans, and those dial plans can have different security settings. If you change the UM startup mode, you must restart the UM Call Router service for the change to take effect. To learn more about UM services, see UM services.

Configure the UM startup mode on an Exchange 2013 Client Access server by using the EAC, as follows:

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Exchange server that you want to modify, and then click **Edit** ✎.
3. On the **Exchange Server** page, click **Unified Messaging**.

4. Under **UM Call Router settings** > **UM startup mode**, select one of the following from the drop-down list:
   - **TCP** Use this option if you aren't using mTLS and are using only Unsecured dial plans.
   - **TLS** Use this option if you're using mTLS and are using only SIP secured or Secured dial plans.
   - **DUAL** Use this option if you're using mTLS and are using Unsecured, SIP secured, and Secured dial plans.
5. After you select the UM startup mode, click **Save**.

Configure the UM startup mode on an Exchange 2013 Client Access server by running the following command in the Shell.

```
Set-UMCallRouterSettings -Server
MyUMCallRouter.northwindtraders.com -UMStartupMode Dual
```

Step 5: Configure the UM startup mode on all Exchange 2013 Mailbox servers

If you're using SIP secured or Secured dial plans, you must configure the UM startup mode on your Exchange 2013 Mailbox servers. You can specify the UM startup mode for the UM service on an Exchange 2013 Mailbox server by using the EAC or the Shell. By default, an Exchange 2013 Mailbox server will start up in TCP mode, but if you're using Transport Layer Security (TLS) to encrypt Voice over IP (VoIP) traffic, you must configure the Exchange 2013 Mailbox server to use TLS or Dual mode. We recommend that all Exchange 2013 Mailbox servers be configured to use Dual as the UM startup mode. This is because all Exchange 2013 Mailbox servers can answer incoming calls for all UM dial plans, and those dial plans can have different security settings. If you change the UM startup mode, you must restart the UM service for the change to take effect. To learn more about UM services, see UM services.

Configure the UM startup mode on an Exchange 2013 Mailbox server by using the EAC, as follows:

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Exchange server that you want to modify, and then click **Edit** ✎.
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Service settings** > **UM startup mode**, select one of the following from the drop-down list:
   - **TCP** Use this option if you aren't using mTLS and are using only Unsecured dial plans.
   - **TLS** Use this option if you're using mTLS and are using only SIP secured or Secured dial plans.
   - **DUAL** Use this option if you're using mTLS and are using Unsecured, SIP secured, and Secured dial plans.
5. After you select the UM startup mode, click **Save**.

Configure the UM startup mode on an Exchange 2013 Mailbox server by running the following command in the Shell.

```
Set-UMService -Identity MyUMServer -ExternalHostFqdn
host.external.contoso.com -IPAddressFamily Any -
UMStartupMode Dual
```

Step 6: Create or configure existing UM dial plans

Depending on your existing Exchange 2007 deployment, you may be required to create new UM dial plans or configure your existing dial plans. A UM dial plan represents a set of traditional or SIP-enabled Private Branch eXchanges (PBXs), IP PBXs, or SIP-enabled PBXs that share common user extension numbers. All users' extensions hosted on traditional or SIP-enabled PBXs or IP PBXs within a dial plan contain the same number of digits. Users can dial one another's telephone extensions without appending a special number to the extension or dialing a full telephone number.

UM dial plans are used in Unified Messaging to make sure that user telephone extensions are unique. In some telephony networks, multiple IP PBXs, traditional PBXs, or SIP-enabled PBXs exist. In these telephony networks, there could be two users who have the same telephone extension number. UM dial plans resolve this situation. Putting the two users into two separate UM dial plans makes their extensions unique. For more information, see UM dial plans.

If required, you can create a UM dial plan by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**, and then click **New +**.
2. On the **New UM Dial Plan** page, complete the following:
   - **Name** Type the name of the dial plan. A UM dial plan name is required and must be unique. However, the name you type is used only for display purposes in the EAC and the Shell. The maximum length of a UM dial plan name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

Although the box for the name of the dial plan can accept 64 characters, the name of the dial plan can't be longer than 49 characters. If you try to create a dial plan name that contains more than 49 characters, you'll receive an error message. The message will say that the UM mailbox policy couldn't be generated because the UM dial plan name is too long. This happens because, when you create a dial plan a default UM mailbox policy named <*DialPlanName*> **Default Policy** is also created. When the 15 characters in the default policy are added to the name of the dial plan, the total characters exceed the limit. The *name* parameter for both the UM dial plan and UM mailbox policy can be 64 characters. However, if the name of the dial plan is longer than 49 characters, the name of the default UM mailbox policy will be longer than 64 characters, and this isn't allowed by the system.

   - **Extension length (digits)** Enter the number of digits for extension numbers in the dial plan. The number of digits for extension numbers is based on the telephony dial plan created on a PBX. For example, if a user associated with a telephony dial plan dials a four-digit extension to call another user in the same telephony dial plan, you select 4 as the number of digits in the extension.

This is a required box that has a value range from 1 through 20. The typical extension length is from 3 through 7 numbers. If your existing telephony environment includes extension numbers, you must

specify a number of digits that matches the number of digits in those extensions.

When you create a Telephone extension dial plan, you're required to enter an extension number for the user if they're linked to a Telephone extension dial plan. An extension number is also required with Session Initiation Protocol (SIP) dial plans or E.164 dial plans when a UM-enabled user is linked to a SIP URI or E.164 dial plan. The extension number is used by Outlook Voice Access users when they access their Exchange mailbox.

- **Dial plan type** A Uniform Resource Identifier (URI) is a string of characters that identifies or names a resource. The main purpose of this identification is to enable VoIP devices to communicate with other devices over a network using specific protocols. URIs are defined in schemes that define a specific syntax and format and the protocols for the call. In simple terms, this format is passed from the IP PBX or PBX. After you create a UM dial plan, you won't be able to change the URI type without deleting the dial plan, and then re-creating the dial plan to include the correct URI type. You can select one of the following URI types for the dial plan:
  - **Telephone extension** This is the most common URI type. The calling and called party information from the VoIP gateway or IP Private Branch eXchange (PBX) is listed in one of the following formats: Tel:512345 or 512345@<*IP address*>. This is the default URI type for dial plans.
  - **SIP URI** Use this URI type if you must have a Session Initiation Protocol (SIP) URI dial plan such as an IP PBX that supports SIP routing or if you're integrating Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server and Unified Messaging. The calling and called party information from the VoIP gateway. IP PBX, or Communications Server 2007 R2 or Lync Server is listed as a SIP address in the following format: sip:<*username*>@<*domain* or *IP address*>:Port.
  - **E.164** E.164 is an international numbering plan for public telephone systems in which each assigned number contains a country code, a national destination code, and a subscriber number. The calling and called party information sent from the VoIP gateway or IP PBX is listed in the following format: Tel:+14255550123.

⚠Warning:

After you create a dial plan, you will be unable to change the URI type without deleting the dial plan, and then re-creating the dial plan to include the correct URI type.

- **VoIP security mode** Use this drop-down list to select the VoIP security setting for the UM dial plan. You can select one of the following security settings for the dial plan:
  - **Unsecured** By default, when you create a UM dial plan, it is set to not encrypt the SIP signaling or RTP traffic. In unsecured mode, the Client Access and Mailbox servers associated the UM dial plan send and receive data from VoIP gateways, IP PBXs, SBCs and other Client Access and Mailbox servers using no encryption. In unsecured mode, neither the Realtime Transport Protocol (RTP) media channel nor the SIP signaling information is encrypted.
  - **SIP secured** When you select **SIP secured**, only the SIP signaling traffic is encrypted, and the RTP media channels still use TCP, which isn't encrypted. With SIP secured, Mutual Transport Layer Security (TLS) is used to encrypt the SIP signaling traffic and VoIP data.

        - **Secured** When you select **Secured**, both the SIP signaling traffic and the RTP media channels are encrypted. Both the secure signaling media channel that uses Secure Realtime Transport Protocol (SRTP) and the SIP signaling traffic use mutual TLS to encrypt the VoIP data.
    - **Country/Region code** Use this box to type the country/region code number to be used for outgoing calls. This number will automatically be prepended to the telephone number that's dialed. This box accepts from 1 through 4 digits. For example, in the United States, the country/region code is 1. In the United Kingdom, it's 44.
3. Click **Save**.

If required, you can create a UM dial plan by running the following command in the Shell.

```
New-UMDialplan -Name MyUMDialPlan -URIType E164 -
NumberOfDigitsInExtension 5 -VoIPSecurity Secured
```

If required, you can configure an existing UM dial plan by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to view or modify, and then click **Edit** ✎.
3. On the **UM Dial Plan** page, click **Configure**. Use the configuration options to view specific dial plan settings and to enable or disable features.

If required, you can configure an existing UM dial plan by running the following command in the Shell.

```
Set-UMDialplan -Identity MyDialPlan -AccessTelephoneNumbers
4255551234 -AudioCodec Wma -CallAnsweringRulesEnabled
$false -OutsideLineAccessCode 9 -VoIPSecurity SIPSecured
```

When you deployed Exchange 2007 Unified Messaging, you were required to add a Unified Messaging server to a UM dial plan for it to answer incoming calls. This is no longer required. In Exchange 2013, Client Access and Mailbox servers can’t be linked with a Telephone extension or E.164 dial plan, but must be linked to SIP URI dial plans. Client Access and Mailbox servers will answer all incoming calls for all types of dial plans.

Step 7: Create or configure existing UM IP gateways

Depending on your existing Exchange 2007 deployment, you may be required to create new UM IP gateways or configure your existing ones. If you’re using SIP secured or Secured dial plans, you must create a UM IP gateway with an FQDN and use the Shell to configure it to listen on port 5061. For existing UM IP gateways, verify that they’re configured with an FQDN and are listening on port 5061. If the UM IP gateway doesn’t use an FQDN, use the EAC or the Shell to change the address. If the UM IP gateway doesn’t use port 5061, use the Shell to change the port. You can view the settings of a UM IP gateway by using the **Get-UMIPGateway** cmdlet.

A UM IP gateway represents a physical Voice over IP (VoIP) gateway, IP PBX, or SIP-enabled PBX.

Before a VoIP gateway, IP PBX, or SIP-enabled PBX can be used to answer incoming calls and send outgoing calls for voice mail users, a UM IP gateway must be created in the directory service.

The combination of the UM IP gateway and a UM hunt group establishes a link between a VoIP gateway, IP PBX, or SIP-enabled PBX and a UM dial plan. By creating multiple UM hunt groups, you can associate a single UM IP gateway with multiple UM dial plans. For more information, see UM IP gateways.

If required, you can create a UM IP gateway by using the EAC, as follows:

1. In the EAC, navigate to **Unified Messaging** > **UM IP gateways**, and then click **Add +**.
2. On the **New UM IP Gateway** page, enter the following information:
   - **Name** Use this box to specify a unique name for the UM IP gateway. This is a display name that appears in the EAC. If you have to change the display name of the UM IP gateway after it's been created, you must first delete the existing UM IP gateway, and then create another UM IP gateway that has the appropriate name. The UM IP gateway name is required, but it's used for display purposes only. Because your organization may use multiple UM IP gateways, we recommend that you use meaningful names for your UM IP gateways. The maximum length of a UM IP gateway name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
   - **Address** You can configure a UM IP gateway with either an IPv4 or IPv6 address or an FQDN. Use this box to specify the IP address or FQDN configured on the VoIP gateway, IP PBX, or SIP-enabled PBX. This box accepts only FQDNs that are valid and formatted correctly.

You can enter alphabetical and numeric characters. IPv4 addresses, IPv6 addresses, and FQDNs are supported. If you want to use mutual TLS between a UM IP gateway and a dial plan operating in either SIP secured or Secured mode, you must configure the UM IP gateway with an FQDN. You must also configure it to listen on port 5061 and verify that any VoIP gateways or IP PBXs have also been configured to listen for mutual TLS requests on port 5061. To configure a UM IP gateway, run the following command in the Shell: `Set-UMIPGateway -identity MyUMIPGateway -Port 5061`

If you use an FQDN, you must also make sure that you've correctly configured a DNS host record for the VoIP gateway, IP PBX, or SIP-enabled PBX so that the host name will be correctly resolved to an IP address. Also, if you use an FQDN instead of an IP address, and the DNS configuration for the UM IP gateway is changed, you must disable and then enable the UM IP gateway to make sure that configuration information for the UM IP gateway is updated correctly.

   - **UM Dial Plan** Click **Browse** to select the UM dial plan that you want to associate with the UM IP gateway. When you select a UM dial plan to associate with a UM IP gateway, a default UM hunt group is also created and associated with the UM dial plan that you selected. If you don't select a UM dial plan, you must manually create a UM hunt group and then associate that UM hunt group with a UM IP gateway that you have created.
3. Click **Save**.

If required, you can create a UM IP gateway by running the following command in the Shell.

```
New-UMIPGateway -Identity MyUMIPGateway -Address
```

```
"MyUMIPGateway.contoso.com"
```

If required, you can configure an existing UM IP gateway by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM IP gateways**, and then click **Edit** ✎.
2. On the **UM IP gateway** page, click **Configure**. Use the configure options to view specific UM IP gateway settings and to enable or disable features.

If required, you can configure an existing UM IP gateway by running the following command in the Shell.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
fe80::39bd:88f7:6969:d223%11 -IPAddressFamily Any -Status
Disabled -OutcallsAllowed $false
```

Step 8: Create a UM hunt group

Depending on your existing Exchange 2007 deployment, you may be required to create new UM hunt groups. A telephony hunt group provides a way to distribute telephone calls from a single number to multiple extensions or telephone numbers. In Unified Messaging, a UM hunt group is a logical representation of a telephony hunt group, and it links a UM IP gateway to a UM dial plan.

You need to have at least one UM hunt group for every IP PBX or PBX hunt group. When you complete the following procedure, one UM hunt group is created by default. If you have more than one IP PBX or PBX hunt group, you need to create additional UM hunt groups. To learn more about UM hunt groups, see UM hunt groups.

If required, you can create a UM hunt group by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan that you want to modify, and then click **Edit** ✎.
2. On the **UM Dial Plan** page, under **UM Hunt Groups**, click **New +**.
3. On the **New UM Hunt Group** page, enter the following information:
   - **Name** Use this box to create the display name for the UM hunt group. A UM hunt group name is required and must be unique, but it's used only for display purposes in the EAC and the Shell. If you have to change the display name of the hunt group after it's been created, you must first delete the existing hunt group and then create another hunt group that has the appropriate name. If your organization uses multiple hunt groups, we recommend that you use meaningful names for your hunt groups. The maximum length of a UM hunt group name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
   - **UM IP Gateway** Use this box to select a UM IP gateway. This box shows the name of the UM IP gateway that will be linked with the UM hunt group. To link a UM IP gateway to the UM hunt group, click **Browse**.
   - **Pilot Identifier** Use this box to specify a string that uniquely identifies the pilot identifier configured on the PBX or IP PBX. An extension number or a SIP Uniform Resource Identifier

(URI) can be used in this box. Alphanumeric characters are accepted in this box. For legacy PBXs, a numeric value is used as a pilot identifier. However, some IP PBXs and SIP-enabled PBXs can use SIP URIs.

4. Click **Save**.

If required, you can create a UM hunt group by running the following command in the Shell.

```
New-UMHuntGroup -Name MyUMHuntGroup -PilotIdentifier
5551234,55555 -UMDialPlan MyUMDialPlan -UMIPGateway
MyUMIPGateway
```

Tip:

You can't configure or change settings for a UM hunt group. If you want to change the configuration settings for a UM hunt group, you must delete it and add a new UM hunt group with the correct settings.

Step 9: Create or configure UM auto attendants

Depending on your existing Exchange 2007 deployment, you may be required to create new UM auto attendants. You can use UM auto attendants to create a voice menu system that lets external and internal callers use the UM auto attendant menu system to locate people and place or transfer calls to company users or departments in an organization. For more information, see Automatically answer and route incoming calls.

In smaller deployments, you may only want to deploy UM so that callers can leave voice mail for users. In these deployments, creating an auto attendant isn't required. However, in most cases, using auto attendants is very useful for external callers when they call in to your organization.

If required, you can create a UM auto attendant by using the EAC, as follows:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. Select the UM dial plan for which you want to add an auto attendant, and then click **Edit** ✎.
2. On the **UM dial plan** page, under **UM Auto Attendants**, click **Add +**.
3. On the **New UM Auto Attendant** page, complete the following boxes:
   - **Name** Use this box to create the display name for the UM auto attendant. A UM auto attendant name is required and must be unique. However, it's used only for display purposes in the EAC and the Shell. If you have to change the display name of the auto attendant after it's created, you must first delete the existing UM auto attendant and then create another auto attendant that has the appropriate name. If your organization uses multiple UM auto attendants, we recommend that you use meaningful names for your UM auto attendants. The maximum length of a UM auto attendant name is 64 characters, and it can include spaces.

Although you can name a new UM auto attendant to include spaces, if you integrate Unified Messaging with Lync Server, the name of the auto attendant can't include spaces. Therefore, if you created an auto attendant that has spaces in the display name and you're integrating with Lync Server, you must first delete that auto attendant and then create another auto attendant that doesn't include spaces in the display name.

- **Create this auto attendant as enabled** Select this check box to enable the auto attendant to answer incoming calls when you finish creating the UM auto attendant. By default, a new auto attendant is created as disabled. If you decide to create the UM auto attendant as disabled, you can use the EAC or the Shell to enable the auto attendant after you finish creating it.
- **Set the auto attendant to respond to voice commands** Select this check box to speech-enable the UM auto attendant. If the auto attendant is speech-enabled, callers can respond to the system or custom prompts used by the UM auto attendant using touchtone or voice inputs. By default, the auto attendant isn't speech-enabled when it's created. For callers to use a speech-enabled auto attendant in a language other than U.S. English (en-US), you must install the appropriate UM language pack and configure the properties of the auto attendant to use this language. The en-US UM language pack is installed by default when you install an Exchange 2013 Mailbox server.
- **Access numbers** Enter the extension or telephone numbers that callers will use to reach the auto attendant. Type an extension number or telephone number in the box, and then click **Add** to add the number to the list. The number of digits in the extension number or telephone number that you provide doesn't have to match the number of digits for an extension number configured on the associated UM dial plan. This is because direct calls are allowed to UM auto attendants.

The number of extension or access numbers you can enter is unlimited. However, you may create a new auto attendant without listing an extension number or telephone number. An extension number or telephone number isn't required. You can edit or remove an existing extension number or pilot identifier. To edit an existing extension number or telephone number, click **Edit**. To remove an existing extension number or telephone number from the list, click **Remove**.

4. Click **Save**.

If required, you can create a UM auto attendant by running the following command in the Shell.

```
New-UMAutoAttendant -Name MyUMAutoAttendant -UMDialPlan
MyUMDialPlan -PilotIdentifierList 56000,56100 -
SpeechEnabled $true -Status Enabled
```

If required, you can configure an existing auto attendant by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**, and then click **Edit** ✎.
2. On the **UM dial plan** page, under **UM Auto Attendants**, select the UM auto attendant that you want to view or configure, and then click **Edit** ✎. Use the configuration options to view specific auto attendant settings and to enable or disable features.

If required, you can configure an existing auto attendant by running the following command in the Shell.

```
Set-UMAutoAttendant -Identity MySpeechEnabledAA -
DTMFFallbackAutoAttendant MyDTMFAA -OperatorExtension 50100
-AfterHoursTransferToOperatorEnabled $true -
```

```
StaroutToDialPlanEnabled $true
```

Step 10: Create or configure UM mailbox policies

Depending on your existing Exchange 2007 deployment, you may be required to create new UM mailbox policies or configure existing UM mailbox policies. UM mailbox policies are required when you enable users for Unified Messaging. The mailbox of each UM-enabled user must be linked to a single UM mailbox policy. After you create a UM mailbox policy, you link one or more UM-enabled mailboxes to the UM mailbox policy. This lets you control PIN security settings such as the minimum number of digits in a PIN or the maximum number of logon attempts for the UM-enabled users who are linked to the UM mailbox policy. For more information, see UM mailbox policies.

If required, you can create a UM mailbox policy by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan that you want to modify, and then click **Edit** .
2. On the **UM dial plan** page, under **UM Mailbox Policies**, click **Add** .
3. On the **New UM Mailbox Policy** page, in the **Name** box, enter the name of the new UM mailbox policy.

Note:

Use this box to specify a unique name for the UM mailbox policy. This is a display name that appears in the EAC. If you must change the display name of the UM mailbox policy after it's been created, you must first delete the existing UM mailbox policy, and then create another UM mailbox policy that has the appropriate name. You can't delete a UM mailbox policy if any UM-enabled users are associated with it.The UM mailbox policy name is required, but it's used for display purposes only. Because your organization may use multiple UM mailbox policies, we recommend that you use meaningful names for your UM mailbox policies. The maximum length of a UM mailbox policy name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

4. Click **Save**.

Note:

When you save the UM mailbox policy, all the default settings, including PIN policies, voice mail features, and Protected Voice Mail settings, are enabled. If you want to customize or change any default settings for the UM mailbox policy you just created, use the **Set-UMMailbox** cmdlet or the EAC.

If required, you can create a UM mailbox policy in the Shell by running the following command.

```
New-UMMailboxPolicy -Name MyUMMailboxPolicy -UMDialPlan
MyUMDialPlan
```

If required, you can configure an existing UM mailbox policy by using the EAC:

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**, and then click **Edit** .
2. On the **UM dial plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to view or configure, and then click **Edit** . Use the configuration options to view specific UM

mailbox policy settings and to enable or disable features.

If required, you can configure an existing UM mailbox policy by running the following command in the Shell.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
LogonFailuresBeforePINReset 8 -MaxLogonAttempts 12 -
MinPINLength 8 -PINHistoryCount 10 -PINLifetime 60 -
ResetPINText "The PIN used to allow you access to your
mailbox using Outlook Voice Access has been reset."
```

Step 11: Move existing UM-enabled mailboxes to Exchange 2013

In Exchange 2007 Unified Messaging, after you've enabled users within the organization to use voice mail, a default set of UM properties is applied to the user so they can use UM features. For more information, see Voice mail for users.

During the process of upgrading, there will be a period of time during which you'll have mailboxes that are UM enabled both on Exchange 2007 Mailbox servers and on Exchange 2013 Mailbox servers. However, if you're moving all UM-enabled users over to Exchange 2013 Mailbox servers, you must use the EAC or the **New-MoveRequest** cmdlet in the Shell from an Exchange 2013 server to retain all of the properties and settings, including the user's PIN.

A move request is the process of moving a mailbox from one mailbox database to another. A local move request is a mailbox move that occurs within a single forest. For more information about mailbox moves, see:

- Mailbox moves in Exchange 2013
- New-MoveRequest
- New-MigrationBatch
- Moving Mailboxes

To move an Exchange 2007 mailbox to an Exchange 2013 Mailbox server by using the EAC:

1. In the EAC, click **Recipients** > **Migration**, and then click **Add** .
2. In the **New local mailbox move** wizard, select the user you want to move, click **OK**, and then click **Next**.
3. On the **Move configuration** page, specify a name for the new batch. Select which options you want for the archive mailbox and the mailbox database location, and then click **New**.

To move an Exchange 2007 mailbox to an Exchange 2013 Mailbox server by using the Shell, run the following command.

```
New-MoveRequest -Identity 'tony@alpineskihouse.com' -
TargetDatabase "DB01"
```

Step 12: Enable new users for UM or configure settings for an existing UM-enabled user

A user must have a mailbox before they can be enabled for Unified Messaging. But, by default, a user who has a mailbox isn't enabled for UM. After the user is UM-enabled, you can manage, modify, and configure the UM properties and voice mail features for them. You can enable a user for UM by using the EAC or the Shell. Learn more at Voice mail for users.

When you enable a user for UM, you must define at least one extension number that UM will use when voice mail is submitted to the user's mailbox and to allow the user to use Outlook Voice Access. After you enable the user for UM, you can add secondary extension numbers to the user's mailbox, modify or remove them by configuring the Exchange Unified Messaging (EUM) proxy address on the user's mailbox, or add or remove additional or secondary extensions for the user in the EAC. To add, modify, or remove extension numbers, E.164 numbers, or SIP addresses, see Voice mail-enabled user procedures.

To enable a user for Unified Messaging by using the EAC:

1. In the EAC, click **Recipients**.
2. In the list view, select the user whose mailbox you want to enable for Unified Messaging.
3. In the Details pane, under **Phone and Voice Features**, click **Enable**.
4. On the **Enable UM mailbox** page, click the **Browse** button next to **UM mailbox policy**, locate the UM mailbox policy to assign the user from the list, and then click **OK**.
5. On the **Enable UM mailbox** page, complete the following boxes:
   - **SIP address or E.164 number** Enter the SIP address or E.164 number for the user. These options are available if the user that you enable for Unified Messaging is assigned to a UM mailbox policy that's linked to either a SIP URI or an E.164 dial plan. Adding a SIP address or E.164 number for a user isn't available if the user is associated with a telephone extension dial plan. When you assign the user to a UM mailbox policy that's linked to a SIP URI or E.164 dial plan, you must also enter an extension number for the user. This extension number is used when users access their mailbox using Outlook Voice Access. The number of digits that you configure in this box must match the number of digits configured on the SIP URI or E.164 dial plan.
   - **Extension number** Enter the extension number for the user you're enabling for UM.

You must provide a valid extension number for the user, and it must match the number of digits specified on the dial plan. You can only enter numeric characters or digits from 1 through 20. The typical extension number is 3 to 7 digits long. The number of digits in the extension is set on the dial plan that's linked to the UM mailbox policy that's assigned to the user.

   - Under **PIN settings**, complete the following:
     - **Automatically generate PIN** Click this button to automatically generate a PIN for the UM-enabled user to use for voice mail access via Outlook Voice Access. This is the default setting. When you click this button, a PIN is automatically generated based on the PIN policies configured on the UM mailbox policy assigned to the user. We recommend that you

use this setting to help protect the user's PIN. The PIN is sent to the user in the welcome message they receive after they're enabled for UM. By default, they'll have to change this PIN when they first sign in to their mailbox to get their voice mail.

- **Type a PIN** Click this button to manually specify a PIN that the user will use to access the voice mail system.The PIN must comply with the PIN policy settings configured on the UM mailbox policy associated with this UM-enabled user. For example, if the UM mailbox policy is configured to accept only PINs that contain seven or more digits, the PIN you enter in this box must be at least seven digits long.
- **Require the user to reset their PIN the first time they sign in** Select this check box to force the user to reset their voice mail PIN when they access the voice mail system from a telephone using Outlook Voice Access for the first time. They will be prompted to enter a PIN that's more familiar to them.It's a security best practice to force UM-enabled users to change their PIN when they first sign in to help protect against unauthorized access to their data and Inbox. This check box is selected by default.

6. On the **Enable UM mailbox** page, review your settings. Click **Finish** to enable the user for Unified Messaging. Click **Back** to make configuration changes.

Enable a user for Unified Messaging in the Shell, run the following command.

```
Enable-UMMailbox -Identity tonysmith@contoso.com -
UMMailboxPolicy MyUMMailboxPolicy -Extensions 51234 -PIN
5643892 -NotifyEmail administrator@contoso.com -PINExpired
$true
```

If required, you can configure a user that's been enabled for UM by using the EAC:

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox for which you want to change the UM mailbox policy.
3. In the details pane, under **Phone and Voice Features** > **Unified Messaging**, click **View details**.
4. On the **UM Mailbox** page, click **UM mailbox settings** to view or change the following UM properties for an existing UM-enabled user:
   - **PIN Status** This display-only box shows the status of the user's mailbox. By default, when a user is UM-enabled, the PIN status is listed as **Not locked out**. However, if the user has input an incorrect Outlook Voice Access PIN multiple times, the status is listed as **Locked Out**.
   - **UM mailbox policy** This box shows the name of the UM mailbox policy associated with the UM-enabled user. You can click **Browse** to locate and specify the UM mailbox policy to be associated with this UM mailbox.
   - **Personal operator extension** Use this box to specify the operator extension number for the user. By default, an extension number isn't configured. The length of the extension number can be from 1 through 20 characters. This enables incoming calls for the UM-enabled user to be forwarded to the extension number that you specify in this box.

You can configure other types of operator extension numbers on dial plans and auto attendants. However, those extensions are generally meant for company-wide receptionists or operators. The personal operator extension setting could be used when an administrative assistant or personal

assistant answers incoming calls for a specific user.

5. On the **UM Mailbox** page, under **Other extensions**, you can add, change, and view extension numbers for the user.
   - To add an extension number, click **Add +**. On the **Add another extension** page, use **Browse** to select the UM dial plan, and then enter the extension number in the **Extension number** box.
   - To remove an extension number, select the extension number you want to remove, and then click **Remove —**.
6. If you make any changes, click **Save**.

If required, you can configure a user that's been enabled for UM in the Shell by running the following command.

```
Set-UMMailbox -Identity tony@contoso.com -
CallAnsweringAudioCodec Wma -CallAnsweringRulesEnabled
$false -FaxEnabled $false -UMSMSNotificationOption
VoiceMail
```

Step 13: Configure your VoIP gateways, IP PBXs, and SIP-enabled PBXs to send all incoming calls to the Exchange 2013 Client Access servers

When Exchange 2013 Client Access and Mailbox servers are installed, they're automatically enabled so they can answer incoming and outgoing voice calls and route voice mail messages to the intended recipients. When you're installing your Exchange 2013 Client Access and Mailbox servers and deploying Unified Messaging, you don't have to link or add Exchange 2013 Client Access or Mailbox servers to UM dial plans. Exchange 2013 Client Access and Mailbox servers answer all incoming calls and then use the UM dial plans to locate users.

The Exchange 2013 Client Access server is the entry point for any inbound calls or Session Initiation Protocol (SIP) requests for Unified Messaging. The service that handles the SIP requests on an Exchange 2013 Client Access server is the UM Call Router service and it runs on each Exchange 2013 Client Access server in your organization.

When you're upgrading to Exchange 2013 UM, you should have already installed and configured single or multiple VoIP gateways to connect to the PBXs in your telephony network, or installed and configured Session Initiation Protocol (SIP)-enabled PBXs or IP PBXs.

The last step in the process of upgrading to Exchange 2013 UM is to configure the VoIP gateways, IP PBXs, or SIP-enabled PBXs to send incoming calls—including callers who want to leave voice mail for a user, calls from UM-enabled users calling in to Outlook Voice Access, and calls from callers that dial in to a UM auto attendant—to your Exchange 2013 Client Access servers. All these calls are received first by a VoIP gateway, IP PBX, or SIP-enabled PBX and then forwarded on to the

Exchange 2013 Client Access servers in your Exchange 2013 organization. For more information, see the following resources:

UM services

Configuration notes for supported VoIP gateways, IP PBXs, and PBXs

Telephony advisor for Exchange 2013

Step 14: Disable call answering on an Exchange 2007 Unified Messaging server

You can disable call answering by disabling UM on an Exchange 2007 Unified Messaging server or by removing the UM server from a dial plan. When you disable UM, you prevent the Unified Messaging server from answering incoming calls. You can choose to disconnect all calls immediately or wait for existing calls to be processed before disabling the Unified Messaging server. You'll want to disable call answering before removing the server from a dial plan so that it will finish processing any incoming calls.

To disable Unified Messaging on an Exchange 2007 UM server by using the Exchange Management Console:

1. In the console tree of the EMC, navigate to **Server Configuration** > **Unified Messaging**.
2. In the result pane, select the Unified Messaging server to disable.
3. In the action pane, click one of the following:
   a. When you select the **Disable Immediately** option, the Unified Messaging server disconnects all calls connected to the Unified Messaging server.
   b. When you select the **Disable After Completing Calls** option, the Unified Messaging server won't accept new calls and won't be disabled until all calls have been processed.
4. In the confirmation dialog box, click **Yes** to continue.

To disable Unified Messaging on an Exchange 2007 UM server by using the Shell, run the following command.

```
Disable-UMServer -Identity MyUMServer -Immediate $true
```

Tip:
You can use the **Disable-UMServer** cmdlet from an Exchange 2007 UM server or the **Disable-UMService** cmdlet from an Exchange 2013 Mailbox server to disable call answering.

Step 15: Remove an Exchange 2007 Unified Messaging server from a dial plan

To process calls, an Exchange 2007 UM server must be added to at least one UM dial plan. A UM server can be added to multiple UM dial plans. You can remove an Exchange 2007 UM server from a UM dial plan. When you remove a UM server from a dial plan, the UM server will no longer answer calls or process UM calls for UM-enabled users.

To remove an Exchange 2007 UM server from a dial plan by using the Exchange Management Console:

1. In the console tree of the EMC, navigate to **Server Configuration** > **Unified Messaging**.
2. In the result pane, select the Unified Messaging server.
3. In the action pane, click **Properties**.
4. On the **UM Settings** tab, in the **Associated Dial Plans** section, click **Remove**.
5. In the confirmation dialog box, click **Yes** to confirm the deletion of the Exchange 2007 Unified Messaging server from the UM dial plan.
6. Click **OK** to close the properties window.

To remove an Exchange 2007 UM server from a dial plan by using the Shell, run the following command.

```
$dp= Get-UMDialPlan "MySIPDialPlan"
$s=Get-UMServer -id MyUMServer
$s.dialplans-=$dp.identity
Set-UMServer -id MyUMServer -dialplans:$s.dialplans
```

In this example, there are three SIP URI dial plans: SipDP1, SipDP2 and SipDP3. This example removes the UM server named `MyUMServer` from the SipDP3 dial plan.

```
Set-UMServer -id MyUMServer -DialPlans SipDP1,SipDP2
```

In this example, there are two SIP URI dial plans: SipDP1 and SipDP2. This example removes the UM server named `MyUMServer` from the SipDP2 dial plan.

```
Set-UMServer -id MyUMServer -DialPlans SipDP1
```

Tip:

You can use the **Set-UMServer** cmdlet in the Shell on an Exchange 2007 Unified Messaging server or the **Set-UMService** cmdlet on an Exchange 2013 Mailbox server to remove an Exchange 2007 UM server from a single or multiple dial plans. For example, to remove a UM server from all dial plans, run the following command: `Set-UMServer -identity MyUMServer -DialPlan $null`

How do you know this worked?

After you've set up Unified Messaging, verify the following to ensure it's working correctly:

- A user you've enabled for voice mail can sign in to Outlook Web App or Outlook and see a Welcome Message for Unified Messaging.

- UM users can receive voice messages.
- UM users can call in to an Outlook Voice Access number to listen to email, calendar items, and voice mail.
- UM is routing calls from outside of your organization, and you can place a call.

Checklist: Upgrade Exchange 2007 UM to Exchange 2013 UM

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-01*

Use this checklist to help you upgrade Exchange 2007 Unified Messaging (UM) to Exchange 2013 UM. Be sure to refer to this information when you're upgrading your Exchange 2007 organization and your UM deployment to Exchange 2013. For step-by-step instructions for upgrading to Exchange 2013 UM, see Upgrade Exchange 2007 UM to Exchange 2013 UM.

Before you start working with this checklist, make sure you're familiar with the concepts in:

- Planning for Unified Messaging
- Telephone system integration with UM
- Connect your voice mail system to your telephone network

For step-by-step guidance about how to upgrade from Exchange 2010 UM to Exchange 2013 UM, see Checklist: Upgrade Exchange 2010 UM to Exchange 2013 UM.

Checklist for upgrading Exchange 2007 UM to Exchange 2013 UM

Done?	Tasks	Topic
	Deploy and configure telephony components	Connect UM to your telephone system
	Review the system requirements before installing Exchange 2013	Exchange 2013 system requirements
	Verify that you meet the prerequisites for installation	Exchange 2013 prerequisites

	Install the required Client Access and Mailbox servers	Install Exchange 2013 using the Setup wizard
	Verify the installation and review the server setup logs	Verify an Exchange 2013 installation
	If required, install the required UM language packs	Install a UM language pack
	Import dial plan and auto attendant custom prompts	Import custom prompts from Exchange 2007 to Exchange 2013
	Export and import certificates	Deploying certificates for UM
	Configure the UM startup mode on all Exchange 2013 Client Access servers	Configure the startup mode on a Client Access server
	Configure the UM startup mode on all Exchange 2013 Mailbox servers	Configure the startup mode on a Mailbox server
	Create or configure existing UM dial plans	Create a UM dial plan Manage a UM dial plan
	Create or configure existing UM IP gateways	Create a UM IP gateway Manage a UM IP gateway
	Create a UM hunt group	Create a UM hunt group
	Create or configure UM auto attendants	Create a UM auto attendant Manage a UM auto attendant
	Create or configure UM mailbox policies	Create a UM mailbox policy Manage a UM mailbox policy

	Move existing UM-enabled mailboxes to Exchange 2013	Mailbox moves in Exchange 2013
	Enable new users for UM or configure settings for an existing UM-enabled user	Enable a user for voice mail Manage voice mail settings for a user
	Configure your VoIP gateways, IP PBXs, and SIP-enabled PBXs to send all incoming calls to the Exchange 2013 Client Access servers	Configuration notes for supported VoIP gateways, IP PBXs, and PBXs Connect a VoIP gateway, IP PBX, or session border controller to UM
	Disable call answering on the Exchange 2007 UM servers	How to Disable Unified Messaging on Exchange 2007
	Remove an Exchange 2007 UM server from a dial plan	How to Remove a Unified Messaging Server from a Dial Plan

Deploying Exchange 2013 UM and Lync Server overview

Exchange Server 2013 > Unified Messaging > Deploying voice mail and UM >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-06-11*

Unified Messaging (UM) and Microsoft Lync Server can be deployed together to provide voice messaging, instant messaging, enhanced user presence, audio/video conferencing, and an integrated email and messaging experience for users in your organization. Unified Messaging is used to provide call answering for voice mail, Outlook Voice Access, and auto attendant services. Microsoft Lync Server enables more advanced features found in Enterprise Voice, such as instant messaging (IM), conferencing, and inbound and outbound calling. This topic describes how to configure Unified Messaging and Microsoft Lync Server to support these features.

Tip:

Microsoft Office Communications Server 2007 R2 can also be deployed together with Unified Messaging. In this topic, "Microsoft Lync Server" refers to Microsoft Lync Server 2010 or Microsoft Lync Server 2013.

Looking for more information about Microsoft Lync Server? See Microsoft Lync Server.

Contents

Deploying Exchange UM and Lync Server overview

Unified Messaging combines voice and email messaging into a single messaging infrastructure. Microsoft Lync Server Enterprise Voice takes advantage of the UM infrastructure to provide voice mail, Outlook Voice Access, call notifications, and auto attendants.

The following list shows the simplified deployment steps for UM and Lync Server. Details about each step are included later in this topic.

1. Install Microsoft Lync Server in the same topology where the Client Access servers running the Microsoft Exchange Unified Messaging Call Router service and the Mailbox servers running the Microsoft Exchange Unified Messaging service will be installed. Confirm that at least one Lync pool is created.
2. Install a certificate that's valid and signed by a private or public certification authority (CA) and is trusted by Lync Server.
3. Install the Client Access servers and Mailbox servers. Verify installation.
4. Install a certificate that's valid and signed by the same CA as the certificate you installed on your Lync servers.
5. Create and configure a Session Initiation Protocol (SIP) URI dial plan.
6. Add all Client Access and Mailbox servers to the SIP URI dial plan. However, if you have multiple SIP URI dial plans, you must add all Client Access and Mailbox servers to all SIP URI dial plans.
7. Run the ExchUcUtil.ps1 script from the <Exchange Installation folder>\Exchange Server\Script folder on a Mailbox server.

Important:

The ExchUcUtil.ps1 script creates one or more UM IP gateways for Lync integration. You must disable outgoing calls on all UM IP gateways except one gateway that the script created. This includes disabling outgoing calls on UM IP gateways that were created before you ran the script. To disable outgoing calls on a UM IP gateway, see Disable outgoing calls on UM IP gateways.

8. Run **OcsUmUtil.exe** from the %CommonProgramFiles%\Microsoft Lync Server 2013\Support

folder on a Lync Server.

9. Deploy the Mediation Server and media gateways.
10. Install a certificate on your Mediation Server that's valid and signed by the same CA as the certificate you installed on your Lync servers.
11. Enable your users for UM and Enterprise Voice.

Certificate configuration recommendations

You must have a certificate that's trusted by both the computers running Exchange and the computers running Lync Server. In an environment that has Lync Server and Unified Messaging, use the following guidelines for deploying a trusted certificate:

- On your Lync servers, Client Access servers, Mailbox servers, Mediation Server, and media gateways, import a certificate that's valid and signed by a private or public CA. This should be a trusted third-party commercial certificate or a public key infrastructure (PKI) certificate.
- It's less complex if you import the same third-party commercial or PKI certificate to each Exchange server. Also, install this trusted certificate on each computer running Microsoft Lync Server and Mediation Server. This will make your certificate deployment less complicated and reduce the administrative overhead associated with deploying certificates. However, make sure you obtain a trusted certificate that supports subject alternative names (SANs).

When you're deploying Transport Layer Security (TLS) with UM, the certificates that are used on the Client Access server and the Mailbox server both must contain the local computer's fully qualified domain name (FQDN) in the certificate's Subject Name. To work around this issue, use a public certificate and import the certificate on all Client Access and Mailbox servers, any VoIP gateways, IP PBXs, and all the Lync servers.

If your deployment includes VoIP gateways or IP PBXs, and if you use a SIP secured or Secured dial plan, a trusted certificate is required between the Client Access and Mailbox servers and the VoIP gateways or IP PBXs. A trusted certificate is also required if a direct SIP connection is used. If you use a SIP secured or Secured dial plan, you can use the same trusted certificate on your Lync and Exchange servers that's used on your VoIP gateways or IP PBXs.

- When you connect Exchange Client Access and Mailbox servers to Microsoft Lync servers or to third-party SIP gateways or Private Branch eXchange (PBX) telephony equipment, you must use a certificate that's valid and signed by an internal or public, third-party certification authority (CA) to establish secured sessions. You can use a single certificate on all the Client Access and Mailbox servers as long as the certificate has the FQDNs of all the Client Access and Mailbox servers in its SAN list. Or, you can generate a different certificate for each Client Access and Mailbox server, with the FQDN of the local computer present in the subject common name (CN) or SAN list of the certificate for that server. Exchange UM doesn't support wildcard certificates with Microsoft Lync Server.

A non-wildcard Subject Name is required for Lync Server and Exchange to work together. UM and Lync Server use the Subject Name as a way to indicate that they're trusted SIP peers. Lync Server also needs a non-wildcard Subject Name in some call-routing scenarios. The FQDN must be used as the "Issued to" value.

For Exchange UM, it isn't supported to put a wildcard in the Certificate Name. However, you can put a wildcard in the SAN.

The following table shows the certificate requirements for installing and configuring certificates for Exchange UM.

Topology	Certificate configuration
Client Access and Mailbox on the same server (without Lync 2010 or 2013; non-SIP dial plans)	A certificate is required between Client Access and Mailbox servers. This is the same certificate that's used between the Client Access and Mailbox servers and the VoIP gateway, IP PBX, or SBC.
Client Access and Mailbox on different servers (without Lync 2010 or 2013; non-SIP dial plans)	A certificate is required. The certificate must match on the Client Access and Mailbox servers. A certificate is also required between Client Access and Mailbox servers and the VoIP gateway, IP PBX, or SBC. This can be the same or a different certificate than the certificate that is used between the Client Access and Mailbox servers.For Client Access and Mailbox servers, you can run the **Create-ExchangeCertificate** cmdlet from either server.
Client Access and Mailbox on the same server (with Lync 2010 or 2013 and SIP dial plans)	A certificate is required. The Client Access and Mailbox servers must have the same certificate as the Lync 2010 or 2013 servers.
Client Access and Mailbox on different servers (with Lync 2010 or 2013 and SIP dial plans)	A certificate is required. The Client Access and Mailbox servers must have the same certificate as the Lync 2010 or 2013 servers.

Return to top

Deployment steps

After you install the required servers in your organization, there's a recommended sequence of steps that you must perform in your Exchange Unified Messaging and Lync Server deployments to correctly deploy Enterprise Voice for your users.

For details about Microsoft Lync Server, see Microsoft Lync Server.

You must complete the following steps to configure Unified Messaging to work with the Enterprise Voice features in Lync Server:

1. Create one or more Unified Messaging SIP URI dial plans that each map to a corresponding Lync Server location profile. An Enterprise Voice location profile must be created for each Exchange UM dial plan. You can use the **Get-UMDialPlan** cmdlet to obtain the FQDN of a SIP URI dial plan. For more information about how to create a SIP URI dial plan, see Create a UM dial plan.

Important:
When you're integrating Exchange UM and Lync Server, you'll probably find it unnecessary to configure dialing rules or dialing rule groups in Exchange UM. Lync Server is designed to perform call routing and number translation for users in your organization, and will also do this when the calls are made by Unified Messaging on behalf of users.

2. Install a certificate on the Client Access and Mailbox servers that's valid and signed by a private or public CA. This is the same CA that was used on the Lync Servers.
3. Encrypt the Voice over IP (VoIP) traffic by configuring the SIP URI dial plan as SIP secured or Secured.

Caution:
If you set your security setting to SIP Secured to require encryption for SIP traffic only, this setting is insufficient on a dial plan if the Front End pool is configured to require encryption (which means that the pool requires encryption for both SIP and RTP traffic). When the dial plan and pool security settings aren't compatible, all calls to Exchange UM from the Front End pool will fail, resulting in an error indicating that you have an "Incompatible security setting".

Although a UM dial plan can be configured as SIP secured or Secured, we recommend that you configure the dial plan as Secured to enable Lync Phone Edition devices to work correctly. This is recommended because of the default encryption level settings configured in Lync Server. A Lync Phone Edition device will work only if the encryption settings are configured as shown in the following table. This table shows the relationship between the encryption settings for Lync Server and UM dial plans.

Encryption settings for Lync Phone Edition

Lync Server	UM dial plan
Encryption required (default)	Secured
Encryption optional	SIP secured/Secured
No encryption	SIP secured

4. Add all Client Access and Mailbox servers to the SIP dial plan. To enable the server to answer incoming calls, you must add all Exchange servers to a dial plan if you want them to answer calls from Lync Server.
5. Set the startup mode and the TLS listening port on the Client Access and Mailbox servers that are added to the SIP URI dial plan to Dual and then restart the Microsoft Exchange Unified Messaging service on each Mailbox server and the Microsoft Exchange Unified Messaging Call Router service on each Client Access server.

6. Create and configure a UM auto attendant. For details, see Set up a UM auto attendant.
7. When you enable users for voice mail, create a SIP address for the users who will use Enterprise Voice. In most cases, this SIP address will be the same SIP address that will be used when a user is enabled for Enterprise Voice. For details, see Enable a user for voice mail.

Important:

Users who are associated with a SIP URI dial plan can't receive incoming faxes. This is because incoming voice and fax calls are routed through a Mediation Server and faxing isn't supported when using a Mediation Server.

8. Open the Exchange Management Shell and run the exchucutil.ps1 script located in the %Program Files%\Microsoft\Exchange Server\V15\Scripts folder. The exchucutil.ps1 script does the following:
   - Grants Lync Server permission to read Exchange UM Active Directory components, specifically, the SIP URI dial plan that was created in the previous task. For details about how to configure permissions in Active Directory, see How to Use ADSI Edit to Apply Permissions.
   - Creates a UM IP gateway for each Lync Server pool or for each server running Lync Server Standard Edition that hosts users who will be enabled for Enterprise Voice. For details, see Create a UM IP gateway.
   - Create an Exchange UM hunt group for each UM IP gateway. The hunt group pilot identifier will be the name of the dial plan associated with the corresponding UM IP gateway. The hunt group must specify the UM SIP dial plan used with the UM IP gateway.
9. Enable users for voice mail. When you enable them, make sure you enter a valid SIP address for the user and link them to a SIP dial plan. For details, see Enable a user for voice mail.

You must also complete the following tasks to configure Lync Server to work with Exchange UM:

- Create location profiles or Lync dial plans. The location profile name doesn't have to match the FQDN of the corresponding UM dial plans.
- Assign location profiles to the Lync Server pools.
- Deploy and configure media gateways or Mediation Servers. You must also import a certificate from the same trusted CA as was used for the certificates on the Client Access and Mailbox servers and Lync Server.
- Define telephone usage, create and assign voice policies and outbound call routes.
- Configure the users for Enterprise Voice and add a TEL URI and SIP identifier.
- Run **ocsumutil.exe**, which creates the contact objects for Outlook Voice Access and for the auto attendants.

Note:

When you install Lync Server, the **msRTC-SIPLine** attribute is added to Active Directory. If you haven't installed Lync Server in your environment, this attribute isn't added to Active Directory, and caller ID name resolution across dial plans in a single forest and in cross-forest scenarios won't work correctly unless you configure Unified Messaging proxy addresses for users who aren't UM-enabled.

After you configure the Lync Server and the Unified Messaging servers, you must enable the user to use Lync Server and install Lync on the user's client computer.

Important:

When you're integrating Unified Messaging and Lync Server, missed call notifications aren't available to users who have a mailbox located on an Exchange 2007 or Exchange 2010 Mailbox server. A missed call notification is generated when a user disconnects before the call is sent to a Mailbox server.

Return to top

For more information

For more information about how to perform the tasks that must be completed for Microsoft Lync Server, see Microsoft Lync Server.

Configure UM to work with Lync Server

Topic Last Modified: *2013-06-11*

When you're integrating Microsoft Lync Server with Exchange Unified Messaging (UM), you have to run the ExchUcUtil.ps1 script in the Shell. The ExchUcUtil.ps1 script does the following:

- Creates a UM IP gateway for each Lync Server pool.

Important:

The ExchUcUtil.ps1 script creates one or more UM IP gateways. You must disable outgoing calls on all UM IP gateways except one gateway that the script created. This includes disabling outgoing calls on UM IP gateways that were created before you ran the script. To disable outgoing calls on a UM IP gateway, see Disable outgoing calls on UM IP gateways.

- Creates a UM hunt group for each UM IP gateway. The pilot identifier of each hunt group specifies the UM SIP URI dial plan used by the Lync Server Front End pool or Standard Edition server that's associated with the UM IP gateway.
- Grants Lync Server permission to read Active Directory UM container objects such as UM dial plans, auto attendants, UM IP gateways, and UM hunt groups.

Important:

Each UM forest must be configured to trust the forest in which Lync Server 2013 is deployed, and the forest in which Lync Server 2013 is deployed must be configured to trust each UM forest. If Exchange UM is installed in multiple forests, the Exchange Server integration steps must be performed for each UM forest or you'll have to specify the Lync Server domain. For example, *ExchUcUtil.ps1 –Forest:<lync-domain-controller-fqdn>*.

For additional management tasks related to integrating Lync Server and Unified Messaging, see Deploying Exchange 2013 UM and Lync Server overview.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Permissions Cmdlets" entry in the Exchange 2013 cmdlets topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to run the ExchUcUtil.ps1 script

Run the ExchUcUtil.ps1 script on any Exchange server in your organization that's in the same topology as Microsoft Lync Server. You can run the script from a Mailbox server using the Shell or you can run the script using Remote Windows PowerShell on a Client Access server. If you run the script on a Client Access server in your organization, the Client Access server will proxy the Remote Windows PowerShell session to a Mailbox server in the organization.

Important:

The ExchUcUtil.ps1 script creates one or more UM IP gateways. You must disable outgoing calls on all UM IP gateways except one gateway that the script created. This includes disabling outgoing calls on UM IP gateways that were created before you ran the script. To disable outgoing calls on a UM IP gateway, see Disable outgoing calls on UM IP gateways.

Important:

You must have the permissions of the Exchange Organization Management role or be a member of the Exchange Organization Administrators security group to run the script.

1. Open the Exchange Management Shell.
2. At the `c:\Windows\System32` prompt, type **cd "<drive letter>\Program Files\Microsoft \Exchange Server\V15\Scripts>.ExchUcUtil.ps1"**, and then press Enter.

How do you know this worked?

To verify that the ExchUcUtul.ps1 script completed successfully, do the following:

- Use the **Get-UMIPGateway** cmdlet or the EAC to view the new UM IP gateway or gateways that were created.
- Use the **Get-UMHuntGroup** cmdlet or the EAC to view the new UM hunt group or groups that were created.

Checklist: Integrate Exchange 2013 UM with Lync Server

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-20*

Use this checklist to install and deploy Unified Messaging (UM) and Microsoft Lync Server 2013. In this topic, "Lync Server" also refers to Lync Server 2010. However, Microsoft Office Communications Server 2007 R2 can also be deployed together with Unified Messaging.

Note:

Before you start working with this checklist, make sure you're familiar with the concepts in:

- Deploying Exchange 2013 UM and Lync Server overview
- Coexistence with Office Communications Server 2007 R2 and Lync Server

For more information about how to perform the tasks that must be completed for Lync Server, see Microsoft Lync Server 2013.

Checklist for deploying Microsoft Lync Server and Unified Messaging

Done?	Tasks	Topic
	Review the system requirements before installing Exchange Server 2013.	Exchange 2013 system requirements
	Verify that you meet the prerequisites for installation.	Exchange 2013 prerequisites
	Review the prerequisites for integrating Microsoft Lync Server 2013 and Microsoft Exchange Server 2013.	Prerequisites for Integrating Microsoft Lync Server 2013 and Microsoft Exchange Server 2013

		Tip: The Unified Communications Managed API (UCMA) 4.0 Runtime is required for Exchange 2013 and Lync Server 2010 and 2013 and is installed during installation. To download and review information about UCMA 4.0, see Unified Communications Managed API 4.0 Runtime
	Install the required Client Access and Mailbox servers.	Install Exchange 2013 using the Setup wizard
	Verify the installation and review the server setup logs.	Verify an Exchange 2013 installation
	If required, install the required UM language packs.	Install a UM language pack
	Create the number of SIP URI dial plans required for your organization.	Create a UM dial plan
	Configure the dial plan security setting.	Configure the VoIP security setting
	Configure the number of concurrent calls on your Mailbox servers.	Configure the number of incoming calls on a Mailbox server
	Configure Outlook Voice Access numbers and other settings.	Manage a UM dial plan
	Add all Client Access and Mailbox servers to each SIP URI dial plan.	Add Mailbox and Client Access servers to a SIP URI dial plan

	Configure outbound dialing for Unified Messaging. Allow all calls on the SIP URI dial plans and UM mailbox policies that are linked to those dial plans.	Authorize calls for users in a dial plan Authorize calls for a group of users
	Create the required number of auto attendants.	Create a UM auto attendant
	Set up and configure each of the UM auto attendants.	Set up a UM auto attendant
	Create, import, and enable a new Exchange certificate for UM or enable a mutually-trusted third-party certificate.	Add Mailbox and Client Access servers to a SIP URI dial plan
	Configure the UM startup mode to Dual or TLS for each Client Access and Mailbox server.	Configure the startup mode on a Mailbox server Configure the startup mode on a Client Access server
	Restart the Microsoft Exchange Unified Messaging service and the Unified Messaging Call Router service on all Exchange servers to load the required certificates.	Stop the Microsoft Exchange Unified Messaging service Start the Microsoft Exchange Unified Messaging service Stop the Microsoft Exchange Unified Messaging Call Router service Start the Microsoft Exchange Unified Messaging Call Router service
	Create a UM mailbox policy or configure the default UM	Create a UM mailbox policy

	mailbox policy.	Manage a UM mailbox policy
	Enable users for Unified Messaging with a SIP address and link them to a SIP URI dial plan.	Enable a user for voice mail
	Review the Lync Server 2013 Planning documentation.	Planning
	Install and deploy Lync Server 2013.	Deploying Lync Server 2013
	Import the mutually-trusted internal PKI or third-party certificate that is imported on the Exchange UM servers.	Configure Certificates for Servers Configure Certificates on the Server Running Microsoft Exchange Server Unified Messaging
	If required, start Lync services on servers to load the certificates.	Start Services on Servers
	Open the Exchange Management Shell and run the exchucutil.ps1 script located in the %Program Files% \Microsoft\Exchange Server \V15\Scripts folder.	Configure UM to work with Lync Server
	Review the requirements for Enterprise Voice.	Software Prerequisites for Enterprise Voice Security and Configuration Prerequisites for Enterprise Voice

	Deploy and configure media gateways or Mediation Servers and define peers.	Deploying Mediation Servers and Defining Peers
	Configure a trunk between a Mediation Server and one or more of the peers to provide public switched telephone network (PSTN) connectivity.	Configuring Trunks
	Create and configure a Lync dial plan and create, define, and associate normalization rules.	Configuring Dial Plans
	Configure voice policies and define telephone usage and outbound call routes.	Configuring Voice Policies, PSTN Usage Records, and Voice Routes
	Run the Exchange Integration utility (ocsumutil.exe), which creates the contact objects for Outlook Voice Access and for the auto attendants.	Configure Lync Server 2013 to Work with Unified Messaging on Microsoft Exchange Server
	Define, deploy, and configure any required advanced Enterprise Voice features.	Deploying Advanced Enterprise Voice Features
	Enable the users for Enterprise Voice. Enter a line URI and assign a voice policy and a Lync dial plan.	Enable Users for Enterprise Voice

Coexistence with Office Communications Server 2007 R2 and Lync Server

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-29*

Communications Server 2007 R2 and Lync Server provide many end-user features, including instant messaging (IM), presence, multiparty IM, and their voice mail functionality can be integrated with Exchange Unified Messaging (UM). For deployments that integrate Lync Server 2010 or 2013, users can be enabled for Enterprise Voice, which lets users who are enabled for voice mail access their voice mail by using Lync Server components.

When you integrate UM with the earlier versions of Office Communications Server or Lync Server, not all features are available. For users to take full advantage of the new enhanced end-user features such as high-resolution photos, the unified contact store, and Lync Archiving Integration, they must have user accounts on Lync Server 2013 and Exchange Server 2013, and must be using the latest version of the Lync 2013 client software. For example, the unified contact store isn't available to users who've been enabled for Enterprise Voice on Lync Server 2010. Also, high-resolution photos can't be displayed in Lync 2010.

When you're integrating Exchange UM with Office Communications Server 2007 R2 or Lync Server 2010, you may need to install additional hotfixes, rollups, cumulative updates, and service packs on the Office Communications Server 2007 R2 or Lync 2010 servers that have been deployed in your organization. What you're required to install will depend on your deployment topology and the product versions you're integrating with Exchange UM.

Required hotfixes, rollups, cumulative updates, and service packs

The following table shows the fixes that are required for each version of the products for integration with UM.

Office Communications Server 2007 R2	Office Communications Server 2007 R2 cumulative update 10 or later.

Lync Server 2010	Lync Server 2010 cumulative update 3 or later.
Lync Server 2013	No updates are required.

Deploying certificates for UM

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-29*

You can use mutual Transport Layer Security (mutual TLS) to enable Unified Messaging (UM) to encrypt data sent between your Microsoft Exchange 2013 servers and VoIP gateways, IP PBXs, session border controllers (SBCs), and Microsoft Lync Server. Certificates bind the identity of the certificate owner to a pair of electronic keys (public and private) that are used to encrypt and sign information digitally.

If you're using SIP secured or Secured dial plans in your Exchange 2010 organization, you'll need to import the certificates that were used to your Exchange 2013 Client Access servers running the Microsoft Exchange UM Call Router service and Mailbox servers running the Microsoft Exchange UM service. You can use one of the following certificates for the UM and UM Call Router services:

- A self-signed (Exchange) certificate
- An internal public key infrastructure (PKI) certificate
- A third-party commercial certificate

By default, when you install Unified Messaging, a self-signed certificate is created and used. This self-signed certificate can be used with VoIP gateways, IP PBXs, and SBCs, but not when you're integrating UM with Lync Server. If you're deploying certificates to be used with Lync Server, you need to use the Exchange certificate wizard or the **New-ExchangeCertificate** cmdlet to obtain a certificate that was issued by an internal or PKI certification authority (CA), or purchase a commercial or third-party certificate that is mutually trusted by Unified Messaging and Lync Server.

Deploying certificates for VoIP gateways, IP PBXs, and SBCs

To enable UM to encrypt data that's sent between VoIP gateways, IP PBXs, and SBCs you need to do the following:

- Use the self-signed UM certificate, create a new Exchange certificate request and obtain an internal PKI certificate, or purchase a third-party commercial certificate that you can use for mutual TLS between UM and VoIP gateways, IP PBXs, and SBCs.

- Import the certificate that will be used on all Client Access and Mailbox servers in your organization.
- Enable the certificate to be used by UM services.
- Import the certificate on your VoIP gateways, IP PBXs, and SBCs.
- Configure the UM dial plan as SIP secured or Secured.
- Configure the UM startup mode on Client Access and Mailbox servers in your organization.
- Create the required UM IP gateways with a fully qualified domain name (FQDN).
- Configure the listening port on the UM IP gateways to use TLS port 5061.
- Restart the Unified Messaging Call Router service on all Client Access servers and restart the Microsoft Exchange Unified Messaging service on all Mailbox servers.

Deploying certificates for Lync Server

To encrypt data that's sent between UM and Lync Server, you need to do the following:

- Create a new Exchange certificate request and obtain an internal PKI certificate, or purchase a third-party commercial certificate. The certificate must be mutually trusted by UM and Lync Server.
- Import the certificate that will be used on all Client Access and Mailbox servers in your organization.
- Enable the certificate to be used by UM services.
- Import the certificate on the computers running Lync Server.
- Configure the SIP URI UM dial plan as SIP secured or Secured.
- Configure the UM startup mode on Client Access and Mailbox servers in your organization.
- Run OcsUmUtil.exe from a Lync Server computer.
- Restart the Unified Messaging Call Router service on all Client Access servers and restart the Microsoft Exchange Unified Messaging service on all Mailbox servers in your organization. For details, see UM services procedures.
- Restart the Lync Server Front-End service on all Lync Servers that are part of an Enterprise Edition Front End pool or restart the Standard Edition Front End Servers. You can use the **Stop-CsWindowsService** cmdlet to stop Lync Server services and the **Start-CsWindowsService** cmdlet to start Lync Server services.

The **Start-CsWindowsService** and **Stop-CsWindowsService** cmdlets are similar to the generic Windows PowerShell cmdlets **Start-Service** and **Stop-Service**. If you want, you could use the **Start-Service** or **Stop-Service** cmdlets to start and stop a Lync Server service. However, the **Start-CsWindowsService Stop-CsWindowsService** cmdlets include a *ComputerName* parameter that makes it easy to stop and start a Lync Server service on a remote computer. To do this, you include the *ComputerName* parameter followed by the fully qualified domain name (FQDN) of the remote computer. The **Start-Service** and **Stop-Service** cmdlets don't have a comparable parameter.

Note:
To fully integrate UM and Lync Server, you must also run the ExchUcUtil.ps1 script on any Client Access or Mailbox server in your organization

Deploying certificates for UM procedures

Unified Messaging > Deploying voice mail and UM > Deploying certificates for UM >

Topic Last Modified: *2013-04-19*

Create certificates for UM

Import or export certificates for UM

Assign a certificate to the UM and UM Call Router services

Create certificates for UM

Deploying voice mail and UM > Deploying certificates for UM > Deploying certificates for UM procedures >

Topic Last Modified: *2013-04-29*

You can use the New Exchange Certificate wizard in the EAC or the Shell to create self-signed certificates or certificate requests for an internal public key infrastructure (PKI) certificate. For Unified Messaging (UM), you can use one of these certificates for both the Microsoft Exchange Unified Messaging service and the Microsoft Exchange Unified Messaging Call Router services. You can use the same certificate for both services, or a different certificate for each service. You can also purchase and import a third-party commercial certificate for UM services. If you're using a self-signed certificate for UM, you may need to include the name of your Client Access and Mailbox servers in the subject alternative name (SAN).

By default, when you install Exchange Server 2013, two self-signed certificates are created: **Microsoft Exchange Server Auth Certificate** and **Microsoft Exchange**. The **Microsoft Exchange** self-signed certificate can be used by UM to encrypt data, but you must assign the certificate to the UM and UM Call Router services. After you assign the certificate to the Unified Messaging services, it can be copied to and imported to the VoIP gateways, IP PBXs, and SIP-enabled PBXs. However, instead of using the default self-signed certificates, you may need to create another one specifically for Unified Messaging.

Caution:
Self-signed certificates can't be used when you're integrating UM with Microsoft Lync Server.

For additional management tasks related to managing certificates for Unified Messaging, see Deploying certificates for UM procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Certificate management" entry in the Exchange and Shell infrastructure permissions topic and the "UM service" entry in the Unified Messaging permissions topic. You must also log on by using an account that's a member of the local Administrators group on that computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to create a certificate request for UM

1. In the EAC, navigate to **Servers** > **Certificates**, and then click **Add +**.
2. On the **New Exchange certificate** page, select **Create a request for a certificate from a certification authority**, and then click **Next**.
3. Enter a friendly name for the certificate, and then click **Next**.
4. If you don't need a wildcard certificate, click **Next**. If you need a wildcard certificate, select **Request a wildcard certificate. A wildcard certificate can be used to secure all sub-domains under your root domain with a single certificate**, enter the name of the root domain, and then click **Next**.
5. Under **Store certificate request on this server**, click **Browse** to go to the location where you want to store the file. You can store the certificate request on any Client Access or Mailbox server in your Exchange organization. Select the location, click **OK**, and then click **Next**.
6. If you requested a wildcard certificate, skip to step 9.
7. If you didn't request a wildcard certificate, you'll need to specify the domains you want to be included in your certificate. If you want to edit a domain, click **Edit**, and then click **Next**.
8. Under **Based on your selections, the following domains will be included in your certificate. You can add additional domains here, or make changes**, you can add, edit, remove, or check the name of domains that are listed under **Domain**. Then click **Next**.
9. Under **Specify information about your organization. This is required by the certification authority**, enter the following:
   - **Organization name**
   - **Department name**
   - **City/Locality**
   - **State/Province**

- **County/Region name** For this option, use the drop-down list to select the country or region.

10. Under **Save the certificate request to the following file**, enter the name of the certificate file, and then click **Finish**.

Use the Shell to create a certificate request for UM

This example creates a new Exchange certificate request for a Mailbox server named `MyMailboxServer` with a friendly name of `CertUM`.

```
New-ExchangeCertificate -FriendlyName 'CertUM' -
GenerateRequest -PrivateKeyExportable $true -KeySize '2048'
-DomainName '*.northwindtraders.com' -SubjectName
'C=US,S=wa,L=redmond,O=northwindtraders,OU=servers,CN=
northwindtraders.com' -Server 'MyMailboxServer'
```

Use the EAC to create a self-signed certificate for UM

1. In the EAC, navigate to **Servers** > **Certificates**, and then click **Add +**.
2. On the **New Exchange certificate** page, choose **Create a self-signed certificate**, and then select **Next**.
3. Enter a friendly name for the certificate, and then select **Next**.
4. Click **Add +** to select the Exchange servers that you want to apply this certificate to, and then select **Next**.
5. Specify the domains that you want to be included in your certificate, and then select **Next**. If you want to add a domain for a service, click **Edit** ✎.
6. Verify that the domains you included are correct, and then select **Finish**.

Important:
When you use the EAC to create a self-signed certificate, you won't be prompted to enable services for the certificate. After the certificate has been created, you can use the EAC or the **Enable-ExchangeCertificate** cmdlet in the Shell to enable the Exchange services. For more information about how to assign a certificate to UM services, see Assign a certificate to the UM and UM Call Router services.

Use the Shell to create a self-signed certificate for UM

This example creates a new Exchange self-signed certificate for a Mailbox server named `MyMailboxServer` with a friendly name of `UMCert`.

```
New-ExchangeCertificate -Services 'UM, UMCallRouter' -
DomainName '*.northwindtraders.com' -FriendlyName
'UMSelfSigned' -SubjectName
'C=US,S=WA,L=Redmond,O=Northwindtraders,OU=Servers,CN=
```

```
Northwindtraders.com' -PrivateKeyExportable $true
```

Tip:

When you specify the services you want to enable by using the *Services* parameter, you'll be prompted to assign those services. In this example, you'll be prompted to enable the certificate for the UM and UM Call Router services. For more information about how to enable a certificate for services, see Assign a certificate to the UM and UM Call Router services.

Import or export certificates for UM

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-12-18*

You can use the EAC or the Shell to import or export self-signed, internal public key infrastructure (PKI), or third-party commercial certificates. For Unified Messaging (UM), you can use one of these certificates for the Microsoft Exchange Unified Messaging service and the Microsoft Exchange Unified Messaging Call Router service. You can use the same certificate for both services, or a different certificate for each service.

Importing certificates for Exchange can be useful when you want to:

- Import a certificate that was exported to a file.
- Import a PKI certificate file that was generated by an internal certification authority.
- Import a third-party commercial certificate.

Exporting an existing certificate from the certificate store on the local Exchange server can be useful when you want to:

- Export it so that it can be imported on another Exchange server.
- Export it so that it can be imported on a VoIP gateway, IP PBX, or SIP-enabled PBX.
- Export the certificate so that you can back up the certificate and its private key.

For additional management tasks related to managing certificates for Unified Messaging, see Deploying certificates for UM procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Certificate management" entry in the Exchange and Shell infrastructure permissions topic and the "UM service" entry in the Unified Messaging permissions topic. You must also log on by using an account that's a member of the local Administrators group on that computer.

- Before you export a certificate, use the **Get-ExchangeCertificate** cmdlet to verify that the *PrivateKeyExportable* attribute on the certificate is set to `$true`.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to export a certificate

1. In the EAC, click **Servers** > **Certificates** > **More options** ···, and then click **Export Exchange certificate**.
2. On the **Export Exchange certificate** page, in the **File to export to** box, enter the name of the certificate file.
3. In the **Password** box, enter the password that you want to use to protect the private key, and then click **OK**.

Use the Shell to export a certificate

This example exports the certificate with the Thumbprint A36DE2B9B62980A717EBD0C3052F5F0B08FBFFCC to a file after it prompts you for a user name and password.

```
$file = Export-ExchangeCertificate -Thumbprint
A36DE2B9B62980A717EBD0C3052F5F0B08FBFFCC -
BinaryEncoded:$true -Password (Get-Credential).password
```

This example does the following:

1. Uses the **Get-ExchangeCertificate** cmdlet to find the certificate that you want to export.
2. Uses the **Export-ExchangeCertificate** cmdlet to set the password for the certificate.
3. Outputs the certificate to a file after you input the user name and password.

```
$file = Get-ExchangeCertificate -DomainName
umcorp.northwindtraders.com | Export-ExchangeCertificate -
BinaryEncoded:$true -Password (Get-Credential).password

Set-Content -Path "d:\umcerts\selfsigned.pfx" -Value
$file.FileData =Encoding Byte
```

Use the EAC to import a certificate

1. In the EAC, click **Servers** > **Certificates** > **More options** ···, and then click **Import Exchange certificate**.
2. On the **Import Exchange certificate** page, in the **File to import from** box, enter the shared folder path and the name of the certificate file. If the certificate is protected with a password, enter the password in the **Password** box, and then click **Next**.
3. Click **Add +** to select the servers that you want to apply the certificate to, and then click **OK**. If you want to remove a server from the list view, click **Remove —**, and then click **Finish**.

Use the Shell to import a certificate

This example imports a certificate from the d:\certificates\exchange\SelfSignedUMCert.pfx certificate file after you enter a user name and password.

```
Import-ExchangeCertificate -FileData ([Byte[]]$(Get-Content -Path d:\certificates\exchange\SelfSignedUMCert.pfx -Encoding Byte -ReadCount 0)) -Password:(Get-Credential).password
```

Assign a certificate to the UM and UM Call Router services

Topic Last Modified: *2013-04-29*

You can use the EAC or the Shell to assign a self-signed, internal public key infrastructure (PKI), or third-party commercial certificate for specific Exchange services. When you use the **New-ExchangeCertificate** cmdlet to assign the certificate to Exchange services with the *Services* parameter, you're prompted to assign the certificate to Exchange services. If you use the EAC to create a certificate, the New Exchange Certificate wizard won't prompt you to assign the certificate to Exchange services. You need to edit the properties of the certificate and assign the certificate by selecting which services you want to assign it to.

Different services have different certificate requirements. For example, some services may only require a server name in the **Subject Name** or **Subject Alternative Name** boxes of a certificate and other services may require a fully qualified domain name (FQDN). Make sure that the certificate name can support the uses required by the services you enable it for.

Caution:

Self-signed certificates can't be used when you're integrating Unified Messaging (UM) with Microsoft Lync Server.

For additional management tasks related to managing certificates for Unified Messaging, see Deploying certificates for UM procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Certificate management" entry in the Exchange and Shell infrastructure permissions topic and the "UM service" entry in the Unified Messaging permissions topic. You must also log on by using an account that's a member of the local Administrators group on that computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to assign a certificate to the Unified Messaging and UM Call Router services

1. In the EAC, navigate to **Servers** > **Certificates**.
2. In the list view, select the certificate that you want to assign to the Unified Messaging and UM Call Router services, and then click **Edit** .
3. On the *<Certificate name>* page, select **Services**, and then select **UM** and **UM call router**.
4. Click **Save**.

Use the Shell to assign a certificate to the Unified Messaging and UM Call Router services

This example assigns a certificate to the Unified Messaging and UM Call Router services.

```
Enable-ExchangeCertificate -Thumbprint
5113ae0233a72fccb75b1d0198628675333d010e -Services 'UM,
UMCallRouter'
```

UM languages, prompts, and greetings

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-22*

You can install and configure language packs to support multiple languages in Unified Messaging (UM) environments.

UM language packs enable callers and Outlook Voice Access users to interact with the voice mail system in multiple languages. After you install an additional language on a Mailbox server, callers and Outlook Voice Access users can hear email messages and interact with the voice mail system in that language.

There are several key components that rely on UM language packs to enable users and callers to interact effectively with Unified Messaging in multiple languages. Each UM language pack includes a Text-to-Speech (TTS) engine, the pre-recorded prompts and support for Automatic Speech Recognition (ASR), and Voice Mail Preview for a specific language. This topic discusses UM language packs, the UM components that use the UM language packs, and how UM language packs—after they're installed—can be used to configure UM dial plans and UM auto attendants to use other languages.

Exchange Unified Messaging language packs are version-specific and platform-specific. Since Exchange Server 2007, there have been multiple releases for UM language packs, including the RTM version of Exchange 2007, Exchange 2007 SP1, SP2, and SP3, the RTM version of Exchange Server 2010, Exchange 2010 SP1 and SP2, and Exchange 2013. For some of these versions, both 32-bit and 64-bit downloads are available, but for other releases only 64-bit downloads are available.

It's very important that you install the correct version and platform of the UM language packs on a Mailbox server. Don't install UM language packs on a Mailbox server that's running an earlier version of Exchange or that's designed for a 32-bit platform.

Contents

Overview of UM language packs

Unified Messaging language packs allow a Mailbox server to speak additional languages to callers and recognize other languages when callers use ASR or when voice messages are transcribed. UM language packs contain:

- Pre-recorded prompts in the language of the UM language pack. For example, "After the tone, please record your message. When you've finished recording, hang up, or press the # key for more options."
- Grammar files in the language of the UM language pack that are used by a Mailbox server to look up the names of given users in the directory.
- Text-to-Speech (TTS) translation so that content (email, calendar, contact information, etc.) can be read to callers in the language of the UM language pack.
- Support for Automatic Speech Recognition, which allows callers to interact with UM using the voice user interface (VUI) in the language of the UM language pack.
- Support for Voice Mail Preview, which allows users to read the transcript of voice mail messages in a specific language from within a supported email client such as Outlook or Outlook Web App.

UM language packs include pre-recorded prompts, TTS conversion support for a specific language, and in some cases, support for ASR. In multiple-language environments, you may have to install additional UM language packs because some callers prefer to be prompted in a different language, or because they receive email in more than one language. You must install multiple UM language packs to support the ability of the Mailbox server to read an email message that contains more than one language, because the TTS conversion system must be instructed which language to select based on the text of the message to be read. If the Unified Messaging language pack hasn't been installed, the email message will be illogical and incoherent when it's read back to the user. Installing the appropriate language pack enables the TTS engine to read email and calendar items to the Outlook Voice Access user by using the correct language and also provides language-specific pre-recorded prompts for Unified Messaging. In some cases, they may also provide support for ASR.

Note:

The TTS engine converts text to speech, but it doesn't convert speech to text. UM-enabled users can send an email message that has a voice file attached to another user. However, they can't create and send a text-based email message to another user.

When you install a language pack, the installation program does the following:

1. Copies the language prompts that will be used to configure UM dial plans and auto attendants.
2. Allows the TTS engine to read messages when Outlook Voice Access users access their Inbox.
3. Enables ASR for speech-enabled UM dial plans and auto attendants for the language installed.
4. Enables Voice Mail Preview for clients in other languages.

You can add UM language packs by using the **Setup.exe** command or by running the

<UMLanguagePack>.exe installation program after you've downloaded the UM language pack from Exchange Server 2013 UM Language Packs. However, you have to use the Setup.exe command to remove a UM language pack. There's no Exchange Management Shell cmdlet that you can use to add or remove languages from a Mailbox server. For more information about how to install a UM language pack, see Install a UM language pack. For more information about how to remove a UM language pack, see Remove a UM language pack.

Note:

By default, when you install a Mailbox server, the U.S. English (en-US) language pack is installed. It can't be removed unless you remove the Mailbox server from the computer.

Return to top

The following table lists the Unified Messaging language packs that are currently available. It also lists the installation file name for each UM language pack and the culture ID for the UM language.

UM language pack installation file names and culture IDs

Language	Country/Region	Culture ID	Installation file name	Availability
Catalan	Spain	ca-ES	UMLanguagePack.ca-ES	Download available
Chinese (Hong Kong)	China	zh-HK	UMLanguagePack.zh-HK	Download available
Chinese (Simplified)	China	zh-CHS	UMLanguagePack.zh-CN	Download available
Chinese (Traditional)	Taiwan	zh-TW	UMLanguagePack.zh-TW	Download available
Danish	Denmark	da-DK	UMLanguagePack.da-DK	Download available
Dutch	Netherlands	nl-NL	UMLanguagePack.nl-NL	Download available
English	Australia	en-AU	UMLanguagePack.en-AU	Download available
English	Canada	en-CA	UMLanguagePack.en-CA	Download available

English	India	en-IN	UMLanguagePack.en-IN	Download available
English	United Kingdom	en-GB	UMLanguagePack.en-GB	Download available
English	United States	en-US	Included with installation of a Mailbox server	Download available
Finnish	Finland	fi-FI	UMLanguagePack.fi-FI	Download available
French	Canada	fr-CA	UMLanguagePack.fr-CA	Download available
French	France	fr-FR	UMLanguagePack.fr-FR	Download available
German	Germany	de-DE	UMLanguagePack.de-DE	Download available
Italian	Italy	it-IT	UMLanguagePack.it-IT	Download available
Japanese	Japan	ja-JP	UMLanguagePack.ja-JP	Download available
Korean	Korean	ko-KR	UMLanguagePack.ko-KR	Download available
Norwegian (Bokmal)	Norway	nb-NO	UMLanguagePack.nb-NO	Download available
Polish	Poland	pl-PL	UMLanguagePack.pl-PL	Download available
Portuguese	Brazil	pt-BR	UMLanguagePack.pt-BR	Download available

Portuguese	Portugal	pt-PT	UMLanguagePack.pt-PT	Download available
Russian	Russia	ru-RU	UMLanguagePack.ru-RU	Download available
Spanish	Spain	es-ES	UMLanguagePack.es-ES	Download available
Spanish	Mexico	es-MX	UMLanguagePack.es-MX	Download available
Swedish	Sweden	sv-SE	UMLanguagePack.sv-SE	Download available

Return to top

UM language components and features

There are several key components and features in Unified Messaging that enable users and callers to interact with a multiple-language voice mail system. For these components and features to work correctly and enable callers to interact with the system in multiple languages, the UM language packs must be installed correctly on a Mailbox server.

Pre-recorded prompts

The Mailbox server role is installed with a set of default audio prompt files. These audio files contain the recordings for Outlook Voice Access menus, voice mail greetings, and numbers that are used by Exchange Unified Messaging. The audio files are played by a Mailbox server to incoming callers, both internal and external. Many of the audio files are default prompts that provide the users of the Telephone User Interface (TUI) and Outlook Voice Access the information they need to move through the TUI and the Voice User Interface (VUI). The prompts are located in <*Program Files*>\Microsoft\Exchange Server\V15\UnifiedMessaging\Prompts\<language>. The prompts used by the Mailbox server to help callers move through the menus shouldn't be replaced or changed.

When an additional UM language pack is installed, the pre-recorded prompts for that language will also be installed. After a UM language pack is installed, the pre-recorded prompts for that language can be used by UM dial plans and auto attendants.

TTS languages

Unified Messaging relies on a Text-to-Speech (TTS) engine. TTS functionality is provided by the Microsoft Speech Server service. The TTS engine reads and converts written text into audible output that can be heard by a caller. The TTS engine reads and converts the following items in a user's mailbox:

- Email and voice mail message bodies, subjects, and names
- Calendar item bodies, subjects, locations, and names
- Personal contact names
- Users' default voice mail greetings

Note:

After a user has recorded personalized voice mail greetings, the TTS version of the voice greetings are no longer used.

Automatic Speech Recognition

In addition to TTS, ASR support is included in Unified Messaging. ASR functionality is provided by the Microsoft Speech Server service. ASR enables callers to use voice commands to move through menus and interact with items from their individual mailboxes, including messages, personal contacts, and calendar. ASR support is included with each language pack.

Return to top

Voice Mail Preview

UM language packs also provide support for Voice Mail Preview, which allows users to quickly triage their voice messages by reading their transcripts from within a supported email client such as Outlook or Outlook Web App.

When a caller leaves a voice message for a UM-enabled user, the voice message file and a transcript of the voice message are placed in the body of the voice message that's sent to the user's mailbox.

All UM language packs are single files that can be downloaded. These language packs include the pre-recorded prompts, grammar files, Text-to-Speech (TTS) translation, and ASR. However, not all the UM language packs contain support for Voice Mail Preview.

The following UM language packs contain support for all the components and features, including Voice Mail Preview:

- English (US) - (en-US)
- English (Canada) (en-CA)
- French (France) - (fr-FR)
- Italian - (it-IT)
- Polish (pl-PL)
- Portuguese (Portugal) (pt-PT)
- Spanish (Spain) (es-ES)

By default, after you install the Mailbox server, the server will send voice mail previews to UM-enabled users if a supported UM language pack is installed.

There are Unified Messaging Voice Mail Preview partners that offer enhanced transcription support for the Voice Mail Preview feature. These partners employ people to correct voice mail transcriptions that were created using ASR. Each Voice Mail Preview partner must meet a set of requirements to be certified to interoperate with Unified Messaging.

If you find that the voice mail previews sent to your users aren't accurate enough, you can contact one of the certified Voice Mail Preview partners listed on the Microsoft Pinpoint for Unified Messaging page and sign up with them at an additional cost.

You can download the UM language packs from the Microsoft Download Center. For details, see Install a UM language pack.

Unified Messaging languages

To enable callers to use the multiple language features found in Unified Messaging, you must first install a UM language pack. Then you have the option to configure other UM components.

- Install the UM language pack on the Mailbox servers in your organization.
- If required, configure the default language for a UM dial plan. This lets Outlook Voice Access users associated with the UM dial plan use the new language when they access their mailbox. However, users can still configure their language setting in the Outlook Web App Options.
- If you need to enable multiple languages on auto attendants, configure the language setting on a UM auto attendant. By default, a UM auto attendant uses the UM dial plan language. However, you can change this setting and enable unauthenticated callers to connect to your organization and move through the auto attendant menus in the language that you've specified on the UM auto attendant.

Unified Messaging language packs

You install a UM language pack on a Mailbox server using Setup.exe. After you install the new language pack, the language associated with the language pack will be added to the list of available languages that you can use. You can view the languages that have been installed using the Get-UMService cmdlet in the Exchange Management Shell.

When you install the UM language pack, the files that are used by the TTS engine and the pre-recorded prompts for the chosen language are copied and made available for users who connect to the voice mail system. You can download the UM language packs from the Microsoft Download Center. For details, see Install a UM language pack.

UM dial plan languages

Each UM dial plan that's created contains a default language setting. The UM dial plan language

setting is needed because Unified Messaging may have to use TTS conversion or play a standard audio prompt for Outlook Voice Access users when they access their mailbox. You don't have to select a default dial plan language.

When you first install Exchange, U.S. English will be the default language, and the only available language option for your dial plan. After you install a UM language pack on a Mailbox server, the language associated with the language pack will be listed as an available option when you configure the default language for the dial plan.

The default language is important to callers. When an Outlook Voice Access user calls in to the voice mail system, the language that is used is based on the language setting in Outlook Web App that was set when the user first signed in to their mailbox. Unified Messaging compares the language set in Outlook Web App to the list of available languages on the dial plan with which the user is associated. If there is no suitable match, the default UM dial plan language will be used. For example, if you have a dial plan that contains only users from France, you may want to change the default language setting on the dial plan to French.

UM auto attendant languages

By default, because UM auto attendants are associated with a UM dial plan when they're created, they use the default language setting of the associated UM dial plan. However, this setting can be changed after the UM auto attendant is created.

The UM auto attendant language setting is needed because Unified Messaging may have to use TTS conversion or play a standard audio prompt to a caller. Unified Messaging doesn't check whether the language of custom prompts for the auto attendant matches the language setting on the auto attendant. However, as a best practice, make sure that the language setting of the auto attendant matches the language of the custom prompts. Otherwise, the caller may hear the system shift from one language to another.

Being able to change the UM auto attendant language setting is also useful if you need several different language-specific auto attendants for callers.

Client language selection process

UM language packs enable callers and Outlook Voice Access users to interact with the Unified Messaging system in multiple languages. After you install additional language packs on a Mailbox server, callers and Outlook Voice Access users can hear email messages and interact with the voice mail system, and Outlook Web App users and users who are using Outlook 2010 or a later version can view the transcript of a voice message using Voice Mail Preview in a specific language.

To support a specific language, a UM client language pack for that language must be installed on each Mailbox server.

In some cases, if a UM language pack for a specific language hasn't been installed and isn't

available, a client fallback language may be used. Fallback UM client languages are available to be used in place of some languages but, for other languages, no fallback language is available. If there isn't a UM language pack installed for a specific language, and no fallback language is available for that language, en-US (US English) will be used.

The following table includes a list of client languages and the fallback languages that are used when a specific UM language pack hasn't been installed on a Mailbox server.

Client fallback languages for UM

Language	Country/ Region	Culture ID	First language chosen, if installed	Second language chosen, if installed	Third language chosen, if installed
Catalan	Spain	ca-ES	ca-ES	en-US	
Chinese (Hong Kong)	China	zh-HK	zh-HK	zh-CN	zh-TW
Chinese (Simplified)	China	zh-CN	zh-CN	zh-HK	zh-TW
Chinese (Traditional)	Taiwan	zh-TW	zh-TW	zh-HK	zh-CN
Danish	Denmark	da-DK	da-DK	en-US	
Dutch	Netherlands	nl-NL	nl-NL	en-US	
English	Australia	en-AU	en-AU	en-US	
English	Canada	en-CA	en-CA	en-US	
English	India	en-IN	en-IN	en-US	
English	United Kingdom	en-GB	en-GB	en-US	
English	United States	en-US	en-US		
Finnish	Finland	fi-FL	fi-FL	en-US	
French	Canada	fr-CA	fr-CA	fr-FR	en-US

French	France	fr-FR	fr-FR	fr-CA	en-US
German	Germany	de-DE	de-DE	en-US	
Italian	Italy	it-IT	it-IT	en-US	
Japanese	Japan	ja-JP	ja-JP	en-US	
Korean	Korea	ko-KR	ko-KR	en-US	
Norwegian (Bokmal)	Norway	nb-NO	nb-NO	en-US	
Polish	Poland	pl-PL	pl-PL	en-US	
Portuguese	Brazil	pt-BR	pt-BR	pt-PT	en-US
Portuguese	Portugal	pt-PT	pt-PT	pt-BR	en-US
Russian	Russia	ru-RU	ru-RU	en-US	
Spanish	Spain	es-ES	es-ES	es-MX	en-US
Spanish	Mexico	es-MX	es-MX	es-ES	en-US
Swedish	Sweden	sv-SE	sv-SE	en-US	

Return to top

Voice mail greetings, announcements, menus, and prompts

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-30*

When you install Unified Messaging (UM), a common set of default audio files used for the voice mail system and for menu prompts, greetings, and informational announcements is installed. Although you can create a fully functional UM auto attendant or dial plan that uses only the default

audio prompts, these prompts are too generic to serve as an acceptable public interface for many companies. This topic discusses the system and menu prompts, greetings, and informational announcements that are used by UM dial plans and auto attendants and how they're used when callers access the voice mail system.

Contents

Overview of audio prompts and greetings

After Unified Messaging is installed, audio files for UM dial plans and auto attendants are copied to the Mailbox server. By default, the installation program copies the audio files to the Program Files \Microsoft\Exchange Server\V15\Unified Messaging\Prompts\<*language*> folder. If you've installed the U.S. English version, a folder named \en is created during installation to hold the U.S. English versions of the system prompts. The Mailbox server plays these system prompts to callers so they can hear greetings, menu prompts, and informational announcements and so they can navigate the UM menus.

These system audio files or prompts should never be replaced. However, UM enables you to customize UM dial plan and auto attendant welcome greetings, main menu prompts, and informational announcements.

The following table summarizes the prompts and greetings used with UM dial plans.

Audio prompts for UM dial plans

Prompts and greetings	Description
System prompts	Must not be modified.
Welcome greeting	The default welcome greeting is a system prompt that is played by default. However, you can use a customized greeting file that you create.
Informational announcement	By default, informational announcements are disabled. If you enable an informational announcement, you must specify a customized

	greeting file.

The following table summarizes the prompts and greetings used with UM auto attendants.

Audio prompts for UM auto attendants

Prompts and greetings	Description
System prompts	Must not be modified.
Business hours menu prompts	By default, business hours menu prompts are enabled and a system prompt is played. However, you can use a customized greeting file that you create.
Non-business hours menu prompts	By default, non-business hours menu prompts are enabled and a system prompt is played. However, you can use a customized greeting file that you create.
Business hours greeting	By default, a business hours greeting is enabled and a system prompt is played. However, you can use a customized greeting file that you create. This is also known as a welcome greeting.
Non-business hours greeting	By default, a non-business hours greeting is enabled and a system prompt is played. However, you can use a customized greeting file that you create. This is also known as a welcome greeting.
Informational announcement	By default, informational announcements are disabled. If you enable an informational announcement, you must specify a customized greeting file.

Caution:
Modifying the installed system prompts isn't supported.

Return to top

System prompts

Unified Messaging is installed with a set of default audio prompts for use with Outlook Voice Access, dial plans, and auto attendants. Hundreds of system prompts for each language are installed on a Mailbox server. The Mailbox server plays the audio files for these system prompts to callers when they access the voice mail system. The following are some examples of these system prompts:

- "Please enter your PIN."
- "To access your mailbox, enter your extension."
- "To contact someone, press the # key."
- "Spell the name of the person you are calling, last name first."
- "To reach a specific person, just tell me the name."

Caution:

Modifying the installed system prompts isn't supported.

Note:

When the Unified Messaging service starts on the Mailbox server, it will verify that all the system prompts are available. If a system prompt can't be found, Unified Messaging will return an error. To fix the error that is returned, locate the event using Event Viewer and copy the file listed in the **Event Properties** window from the installation DVD into the appropriate folder on the Mailbox server.

UM dial plan greetings and announcements

After you install the Mailbox server and create a UM dial plan, you have the option to use the audio files for the default system prompts that are created during installation or to create customized audio files that can be used with UM dial plans.

UM dial plans have a welcome greeting and an optional informational announcement you can modify. The welcome greeting is used when an Outlook Voice Access user or another caller calls the subscriber access number. The callers hear a default welcome greeting that says, "Welcome, you are connected to Microsoft Exchange." You might want to change this default greeting and provide an alternative welcome greeting specific to your company, for example, "Welcome to Outlook Voice Access for Woodgrove Bank." If you customize this greeting, you can record the customized greeting and save it as a .wav file, and then you can configure the dial plan to use this customized greeting.

Unified Messaging allows for an informational announcement to follow the welcome greeting. By default, there is no informational announcement configured. However, you may want to provide one for callers. You can use the informational announcement for general announcements that change more often than the welcome greeting or for announcements required by corporate

compliance policies. When it's important that the whole informational announcement is heard, you can configure it to be uninterruptible. This prevents a caller from pressing a key or speaking a command to interrupt and stop the informational announcement.

The following table describes the UM dial plan greetings and informational announcements.

UM dial plan greetings and informational announcements

Greeting	Default example	Customized example
Welcome greeting	"Welcome, you are connected to Microsoft Exchange."	"Welcome to Outlook Voice Access for Woodgrove Bank."
Informational announcement	By default, an informational announcement isn't configured.	"By using this system you agree to adhere to all corporate policies when you are accessing this system."

When you are customizing and configuring greetings and announcements, make sure the language setting configured on the UM dial plan is the same as the language of the custom prompts you create. If not, a caller may hear a message or greeting in one language and another message or greeting in a different language.

Return to top

UM auto attendant greetings, announcements, and menu prompts

As with UM dial plans, UM auto attendants have a welcome greeting, an optional informational announcement, and an optional custom menu prompt. You can configure different versions of the welcome greeting and menu prompt for business hours and non-business hours. You can modify all of them.

The welcome greeting is the first thing a caller hears when a UM auto attendant answers the call. By default, this says, "Welcome to the Microsoft Exchange auto attendant." The audio file that is played for the call is the default system prompt for the UM auto attendant. However, you may want to provide an alternative greeting specific to your company, for example, "Thank you for calling Woodgrove Bank." To customize this welcome greeting, record the customized greeting and save it as a .wav file, and then configure the auto attendant to use this customized greeting. As with the welcome greetings, you can also customize the menu prompts.

Unified Messaging also allows for an informational announcement to follow a business hours greeting or a non-business hour greeting. By default, no informational announcement is configured, but you may want to provide one to callers. The informational announcement can announce your

company's business hours, for example, "Our business hours are 8:00 A.M. to 5:00 P.M., Monday through Friday, and 8:30 A.M. to 1:00 P.M. on Saturday." The informational announcement can also provide information required for compliance with corporate policies, for example, "Calls may be monitored for training purposes." When it's important that the whole informational announcement is heard, you can configure it to be uninterruptible. This prevents the caller from pressing a key or speaking a command to interrupt and stop the informational announcement.

The following table describes the UM auto attendant greetings and informational announcements.

UM auto attendant greetings, informational announcement, and menu prompts

Greeting	Default example	Customized example
Business hours greeting	"Welcome to the Microsoft Exchange auto attendant."	"Thank you for calling Woodgrove Bank."
Non-business hours greeting	No default non-business hours greeting is played until you configure the business hours for the auto attendant. However, the business hours greeting is played for callers during all times of the day.	"You have reached Woodgrove Bank after business hours. Our business hours are from 8:00 A.M. until 5:00 P.M., Monday through Friday."
Informational announcement	By default, informational announcements aren't configured.	"Calls may be monitored for training purposes."
Business hours main menu prompt	No default business hours main menu prompt will be played until you configure key mappings on the auto attendant.	"For technical support, press or say 1. For corporate offices and administration, press or say 2. For sales, press or say 3."
Non-business hours main menu prompt	No default non-business hours main menu prompt will be played until you configure key mappings and the business hours schedule on the auto attendant.	"Your call is very important to us. However, you have reached Woodgrove Bank after business hours. If you want to leave a message, please press or say 1, and we will return

		your call as soon as possible."

As with UM dial plans, make sure the language setting configured on the UM auto attendant is the same as the language of the custom greetings you create and is set to the same language as the UM dial plan. If not, a caller may hear a message or greeting in one language and another message or greeting in a different language.

Return to top

Customizing greetings, announcements, and menu prompts, and navigation menus

Although the system prompts mustn't be replaced or changed, you'll probably want to customize the greetings, informational announcements, menu prompts and navigation menus used with UM dial plans and auto attendants. After installation, you can configure the UM dial plans and auto attendants to use custom audio files (.wav or .wma). You must follow these steps before you can enable custom voice prompts for callers:

1. Record the custom greetings, announcements, and prompts and save then as .wav files. The Linear PCM (16 bit/sample), 8 kilohertz (kHz) audio codec must be used to encode the .wav files. If you don't use this specific format for the .wav files, an error will be generated stating that the source file is in an unsupported format. Although an error is generated, the error won't appear in Event Viewer.
2. Configure the UM dial plan or auto attendant to use the customized greetings, announcements, and prompts.

By default, when you create a UM auto attendant, the business and non-business hours greetings or prompts aren't configured and no key mappings are defined for business or non-business hours main menu prompts. To correctly configure customized greetings and prompts for an auto attendant, you must:

- Configure business and non-business hours on the **Business hours** page.
- Create the greeting audio (.wav or .wma) files that will be used for the business and non-business hours welcome greetings.
- Configure the business and non-business hours welcome greetings on the **Greetings** page.
- Create the greeting files that will be used for the business and non-business hours main menu prompt greetings.
- Configure the business and non-business hours main menu prompt greetings on the **Greetings** page.
- Enable and configure the business and non-business hours menu navigation on the **Menu navigation** page.

Return to top

UM languages, prompts, and greetings procedures

Exchange Server 2013 > Unified Messaging > UM languages, prompts, and greetings >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-17*

Install a UM language pack

Set the default language on a dial plan

Select the language for an auto attendant

Remove a UM language pack

Import and export custom greetings, announcements, menus, and prompts

Import custom prompts from Exchange 2007 to Exchange 2013

Enable custom prompt recording using the telephone user interface

Install a UM language pack

Unified Messaging > UM languages, prompts, and greetings > UM languages, prompts, and greetings procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-16*

To make a language available in the list of available Unified Messaging languages on a UM dial plan or UM auto attendant, you must first install the appropriate UM language pack. You install the language pack on a Mailbox server running the Microsoft Exchange Unified Messaging service by using the language-specific self-extracting executable file or the **setup.exe /AddUmLanguagePack** command. Before you can install a UM language pack, you must first download it to a local folder on the Mailbox server. You can download UM language packs from Exchange Server 2013 UM Language Packs. There's a separate executable file for each language.

After you install the appropriate UM language pack, you can view the list of installed UM language packs by viewing the drop-down list on the **Settings** page of a UM dial plan or the **Language for automated voice interface** drop-down list on the **General** page of a UM auto attendant. You can also configure the default language to be a language other than English (en-US) on UM dial plans and auto attendants.

Caution:

The UM language packs for Microsoft Exchange Server 2007 or Exchange 2007 Service Pack 1 (SP1), SP2, or SP3 or Exchange 2010 Service Pack 1 SP1, SP2, or SP3 can't be used on an Exchange 2013 Mailbox server.

For additional tasks related to UM languages, see UM languages, prompts, and greetings procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox server (UM service)" entry in Unified Messaging permissions.
- Verify that the Mailbox server is installed on a different computer than the Client Access server or that the Client Access and Mailbox servers are on the same hardware.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the UM Language Pack Installation (.exe) file to install a UM language pack

1. From the Microsoft Download Center, download the language-specific UM language pack (.exe) file into a local folder on the Mailbox server.
2. Double-click the UMLanguagePack.<*CultureCode*>.*exe* file. For example, for the German UM language pack, you would download the file named UMLanguagePack.de-DE.exe.
3. In the Exchange 2013 Setup wizard, on the **License Agreement** page, read the terms of the agreement, select **I accept the terms in the license agreement**, and then click **Next**.
4. On the **Unified Messaging Language Pack** page, verify that the correct language is listed in the **The following Unified Messaging Language Pack(s) will be installed** window, and then click **Install**.
5. Click **Finish** to complete the installation of the UM language pack.

Use setup.exe to install a UM language pack

This example installs the Japanese (ja-JP) UM language pack that's been downloaded to the D:\Exchange\UMLanguagePacks folder on a Mailbox server.

```
setup.exe /AddUmLanguagePack:ja-JP /s:d:\Exchange
\UMLanguagePacks /IAcceptExchangeServerLicenseTerms
```

This example installs the Mexican Spanish (es-MX) and German (de-DE) UM language packs that have been downloaded to the D:\Exchange\UMLanguagePacks folder on a Mailbox server.

```
setup.exe /AddUmLanguagePack:es-MX,de-DE /s:d:\Exchange
\UMLanguagePacks /IAcceptExchangeServerLicenseTerms
```

⚠Warning:

If you don't use the /IAcceptExchangeServerLicenseTerms parameter, you'll see the following error: Welcome to Microsoft Exchange Server 2013 Unattended Setup. You need to accept the license terms to install Microsoft Exchange Server 2013. To read the license agreement, visit http://go.microsoft.com/fwlink/p/?LinkId=150127. To accept the license agreement, add the /IAcceptExchangeServerLicenseTerms parameter to the command you're running. For more information, run setup /?.

For more information about available UM languages and the culture codes, see UM languages, prompts, and greetings.

Set the default language on a dial plan

Unified Messaging > UM languages, prompts, and greetings > UM languages, prompts, and greetings procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-22*

You can set the default language for a Unified Messaging (UM) dial plan. Each dial plan you create will initially use English (en-US) as the default language. The English (en-US) language pack is installed on all versions of Microsoft Exchange Server 2013 and can't be removed.

If you want to select another language as the default language, for example, German (de-DE), you must first download the German UM language pack .exe file from Exchange Server 2013 UM Language Packs and install that language pack on the Mailbox server by using the UMLanguagePack.de-de.exe installation file. After you've installed the language pack, you can set the default language to a language other than English (en-US) on your UM dial plans.

For additional tasks related to UM languages, see UM languages, prompts, and greetings procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to set the default language on a UM dial plan

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan that you want to modify, and then, on the toolbar, click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. On the **Settings** page, under **Audio language**, select the language you want to set from the drop-down list.
5. Click **Save** to accept your changes.

Use the Shell to set the default language on a UM dial plan

This example sets the default language on a UM dial plan named `MyUMDialPlan` to German.

```
Set-UMDialPlan -Identity MyUMDialPlan -DefaultLanguage de-DE
```

This example sets the default language on a UM dial plan named `MyUMDialPlan` to Japanese.

```
Set-UMDialPlan -Identity MyUMDialPlan -DefaultLanguage ja-JP
```

This example sets the default language on a UM dial plan named `MyUMDialPlan` to Australian English.

```
Set-UMDialPlan -Identity MyUMDialPlan -DefaultLanguage en-AU
```

Select the language for an auto attendant

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can configure the default prompt language setting on a Unified Messaging (UM) auto attendant. The language setting available on a UM auto attendant enables you to configure the default prompt language on the auto attendant. When you're using the default system prompts for the auto attendant, this is the language that the caller hears when the auto attendant answers the incoming call. This language setting affects only the default system prompts that are provided after you have installed the Mailbox server that is running the Microsoft Exchange Unified Messaging service. This setting doesn't affect custom prompts that are configured on an auto attendant. The languages that are available are based on the Unified Messaging language packs that are installed on the Mailbox server.

Each auto attendant you create will initially use English (en-US) as the default language. The English (en-US) language pack is installed by default on all versions of Microsoft Exchange 2013 and can't be removed. If you want to select another language, for example, German (de-DE), you must first download the German UM language pack .exe file from Exchange Server 2013 UM Language Packs and install the UM language pack on the Mailbox server by using the UMLanguagePack.de-de.exe installation file. After you've installed the UM language pack, you can set the default language to a language other than English (en-US) on UM auto attendants.

For additional tasks related to UM languages, see UM languages, prompts, and greetings procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the default language setting

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to modify, and then on the toolbar, click **Edit** .
3. On the **UM dial plan** page, under **UM Auto Attendants**, select the UM auto attendant you want to change, and then click **Edit** .
4. On the **General** page, under **Language for automated voice interface**, select the required language from the drop-down list.
5. Click **Save** to accept your changes.

Use the Shell to configure the default language setting

This example sets the default language on the UM auto attendant `MyUMAutoAttendant` to English (Great Britain).

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -Language
en-GB
```

This example sets the default language on the UM auto attendant `MyUMAutoAttendant` to German.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -Language
de-DE
```

Remove a UM language pack

Unified Messaging > UM languages, prompts, and greetings > UM languages, prompts, and greetings procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-14*

You can use the EAC or the Shell to manage Unified Messaging (UM) languages on Mailbox servers running the Microsoft Exchange Unified Messaging service. However, to remove a language from the list on a UM dial plan, you must remove the appropriate UM language pack from the Mailbox server by using the **Setup.exe /RemoveUmLanguagePack** command. After you remove the UM

language pack from the Mailbox server, the language won't be available when you configure a UM dial plan or a UM auto attendant. You can view the UM language packs that are installed by viewing the properties of the Mailbox server or by using the **Get-UMService** cmdlet.

For additional tasks related to UM languages, see UM languages, prompts, and greetings procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox server (UM service)" entry in the Unified Messaging permissions topic.
- Verify that a UM language pack other than en-US is installed.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use Setup.exe to remove a UM language pack

At a command prompt, run the following command.

```
Setup.exe /RemoveUmLanguagePack:<UmLanguagePackName>
```

In the previous command, *<UmLanguagePackName>* is the name of the UM language pack, for example, fr-FR.

Caution:

You can't use the Setup.exe file that's located in the \Bin folder to remove a UM language pack after you've installed any updates. You must use the Setup.exe file from the Exchange 2013 DVD or the downloaded source files. If you don't, you'll see the following error: There is a version mismatch between the running application and the installed application.

Import and export custom greetings, announcements, menus, and prompts

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-04-16*

You can import and export the audio files that you've recorded to use on Unified Messaging (UM) dial plans and auto attendants. For example, you might want to export and save a copy of an audio file if you're upgrading from a previous version of Exchange. Or, you might need to import a copy of a recorded audio prompt before configuring a dial plan or auto attendant.

The audio files are used for the following purposes:

- On UM dial plans, audio files are used for customized welcome greetings and informational announcements. They're played when Outlook Voice Access users call in to an Outlook Voice Access number.
- On UM auto attendants, audio files are used for customized non-business and business hours greetings, informational announcements, menu prompts, and navigation menus. They're played when callers call in to a UM auto attendant.

The following audio file formats are supported for custom greetings, announcements, menus, and prompts:

- .wma files encoded with Windows Media Audio 9.2 - 96 kbps/44 kHz/stereo 1-pass CBR (Windows Sound Recorder)
- Windows Media Audio Voice 9 - 8 kbps/8 kHz/Mono, and .wav files encoded with the Linear PCM (16 bit/sample), 8 kilohertz (kHz) audio codec

You import the audio files that are used by UM dial plans and auto attendants into the system mailbox named {e0dc1c29-89c3-4034-b678-e6c29d823ed9} and export the audio files from this system mailbox. The audio files can be imported and exported by using the **Import-UMPrompt** and **Export-UMPrompt** cmdlets.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" and "UM auto attendants" entries in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- You can only use the Shell to perform this procedure.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to import custom greetings, announcements, menus, and prompts for UM dial plans and auto attendants

This example imports the welcome greeting file named welcomegreeting.wav from d:\UMPrompts into the UM dial plan `MyUMDialPlan`.

```
[byte[]]$c = Get-content -Path "d:\UMPrompts
\welcomegreeting.wav" -Encoding Byte -ReadCount 0
Import-UMPrompt -UMDialPlan MyUMDialPlan -PromptFileName
"welcomegreeting.wav" -PromptFileData $c
```

This example imports the welcome greeting file named welcomegreeting.wav from d:\UMPrompts into the UM auto attendant `MyUMAutoAttendant`.

```
[byte[]]$c = Get-content -Path "d:\UMPrompts
\welcomegreeting.wav" -Encoding Byte -ReadCount 0
Import-UMPrompt -UMAutoAttendant MyUMAutoAttendant -
PromptFileName "welcomegreeting.wav" -PromptFileData $c
```

Use the Shell to export custom greetings, announcements, menus, and prompts from UM dial plans and auto attendants

This example exports the welcome greeting for the UM dial plan `MyUMDialPlan` and saves it as the file named welcomegreeting.wav.

```
$prompt = Export-UMPrompt -PromptFileName
"customgreeting.wav" -UMDialPlan MyUMDialPlan
set-content -Path "d:\DialPlanPrompts\welcomegreeting.wav"
-Value $prompt.AudioData -Encoding Byte
```

This example exports the business hours welcome greeting for the UM auto attendant

MYUMAutoAttendant and saves it as the file named BusinessHoursWelcomeGreeting.wav.

```
$prompt = Export-UMPrompt -BusinessHoursWelcomeGreeting -
UMAutoAttendant MyUMAutoAttendant
set-content -Path "d:\UMPrompts
\BusinessHoursWelcomeGreeting.wav" -Value $prompt.AudioData
-Encoding Byte
```

Import custom prompts from Exchange 2007 to Exchange 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-16*

You can import the audio files that contain custom greetings, announcements, menus, and prompts from Exchange 2007 Unified Messaging (UM) to Exchange 2013 Unified Messaging. Using a Shell script, the prompts are imported into an Exchange system mailbox named {e0dc1c29-89c3-4034-b678-e6c29d823ed9}, which is created when you install Microsoft Exchange 2013. This system mailbox is used in Unified Messaging to store dial plan and auto attendant custom greetings, announcements, menus, prompts, and UM reports.

The audio files, in .wav or .wma format, are used as follows:

- On UM dial plans, audio files are used for customized welcome greetings and informational announcements. They're played when Outlook Voice Access users call in to an Outlook Voice Access number.
- On UM auto attendants, audio files are used for customized non-business and business hours greetings, informational announcements, menu prompts, and navigation menus. They're played when callers call in to a UM auto attendant.

You use the MigrateUMCustomPrompts.ps1 script to migrate a copy of all Exchange Server 2007 UM custom greetings, announcements, menus, and prompts to Exchange 2013 UM for all Exchange 2007 UM dial plans and UM auto attendants.

By default, the MigrateUMCustomPrompts.ps1 script is located in the <Program Files>\Microsoft \Exchange Server\V15\Scripts folder on an Exchange 2013 server.

Note:

The MigrateUMCustomPrompts.ps1 script is included with Exchange 2013. It must be run on an Exchange 2013 Mailbox server in the same organization with your Exchange 2007 UM

servers.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" and “UM auto attendants” entries in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- You can only use the Shell to perform this procedure. To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the MigrateUMCustomPrompts.ps1 script to migrate a copy of all custom prompts for UM dial plans and auto attendants

1. Click **Start** > **All Programs** > **Microsoft Exchange Server 2013** > **Exchange Management Shell**.
2. In the Shell, at the prompt, type the path to the script. For example, type **cd "D:\Program Files \Microsoft\Exchange Server\V15\Scripts"**, and then press Enter.
3. At the Shell prompt, type **".\MigrateUMCustomPrompt",** and then press Enter.

Enable custom prompt recording using the telephone user interface

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-16*

You can use the Shell to enable the recording of custom prompts and greetings for Unified Messaging (UM) dial plans and auto attendants using the telephone user interface (TUI). This can be useful when you want to change a custom greeting or announcement by using the EAC or the Shell, or when there's an emergency such as an organization closure because of severe weather. When you're changing a custom greeting or announcement on a UM auto attendant, you must enable TUI prompt recording on the dial plan that the UM auto attendant is linked to.

For additional management tasks related to UM dial plans, see **UM Dial Plan Procedures**.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" and "UM auto attendants" entries in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to enable a custom prompt or greeting recording using the TUI

To record custom prompts and greetings by using the telephone user interface (TUI), follow these steps:

1. Create a domain user account that cannot log on interactively.
2. Delegate the Exchange Organization Administrator role to the domain user account.

3. Create a mailbox for the domain user.
4. Enable the domain user's mailbox for Unified Messaging.

Important:
Allow only those administrators who are managing prompts and greetings access to the extension number and PIN for the user account. Use this user account only for managing prompts over the telephone.

5. Create and save a .wav or .wma file to use for a custom greeting for the UM dial plan or auto attendant.

Note:
MP3 files can't be used for custom prompts.

6. Use the EAC or the Shell to configure the dial plan to use the custom welcome greeting or configure the auto attendant to use the business or non-business hours greeting. For details about configuring a dial plan, see Enable a customized greeting for Outlook Voice Access users. For details about configuring an auto attendant, see Enable a customized business hours greeting or Enable a customized non-business hours greeting.
7. Run the following cmdlet:

```
Set-UMDialPlan -identity MyUMDialPlan -
TUIPromptEditingEnabled $true
```

Note:
Before you can enable the recording of a custom prompt or greeting, you must sign in to the mailbox that's set up for recording prompts. After you record the new prompt or greeting, you must sign out and then sign back in before you can hear the new prompt or greeting when you use the TUI.

Perform TUI prompt recording on a UM auto attendant

1. Verify that the auto attendant is linked to the dial plan that you've enabled for TUI prompt recording.
2. Call a phone number that's been configured on the UM auto attendant.
3. While the non-business or business hours greeting for the auto attendant is being played, press the pound key (#), and then press the star key (*).
4. You'll be prompted to enter the extension number for the user. Enter the extension number of the UM-enabled user who has permission to perform TUI prompt recording.
5. You'll be prompted for a PIN. Enter the user's PIN.
6. Follow the system prompts to edit or update the greeting or informational announcement for the auto attendant.

Perform TUI prompt recording on a UM dial plan

1. Call an Outlook Voice Access number you use to sign in to Outlook Voice Access.
2. While the welcome greeting for the dial plan is being played, press the pound key (#), and then

press the star key (*).
3. If you're calling from a phone that's used by a UM-enabled user, you'll be prompted for a PIN. Instead of entering the PIN, press the star key (*). You'll be prompted for an extension number. Enter the extension number of the UM-enabled user who has permission to perform TUI prompt recording.
4. If you're calling from a phone that's not used by a UM-enabled user, you'll automatically be prompted for an extension number. Enter the extension number of the UM-enabled user who has permission to perform TUI prompt recording.
5. You'll be prompted for a PIN. Enter the user's PIN.
6. Follow the system prompts to edit or update the welcome greeting for the dial plan or the informational announcement.

Telephone system integration with UM

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-20*

To successfully deploy Unified Messaging (UM), you must have a good understanding of basic telephony concepts and telephony components. After you understand telephony basics, you can integrate UM into an Exchange organization. Basic concepts and components include the following:

- Circuit-switched and packet-switched networks
- Private Branch eXchange (PBX)
- IP PBX
- Voice over Internet Protocol (VoIP)
- VoIP gateways

In an on-premises, hybrid, or Office 365 environment, connecting and configuring the required telephony components is the most complex and important step in successfully deploying UM, with or without Lync Server Enterprise Voice. You'll need to connect and configure VoIP gateways, advanced VoIP gateways, PBXs, IP PBXs, and session border controllers (SBCs) for a traditional telephony network and connect to a telephony network if you'll be using Microsoft Lync Server and UM.

Planning and deploying a new deployment of UM or upgrading a legacy voice mail system can pose challenges for organizations. It requires significant knowledge about VoIP gateways, PBXs, IP PBXs, Microsoft Lync Server, and Unified Messaging. Depending on your technical experience with Exchange and voice mail systems, you might want to obtain the assistance of a Unified Messaging specialist. An Exchange Unified Messaging specialist will help make sure that there's a smooth transition from a legacy or third-party voice mail system to Exchange Unified Messaging. For more information about how to contact a Unified Messaging specialist, see Microsoft Exchange Server

2013 Unified Messaging (UM) Specialists.

Integrating your telephony network

Unified Messaging requires that you integrate your Exchange Server deployment with your existing telephony network or integrate UM with Microsoft Lync Server for your organization. To successfully deploy and manage UM voice mail you need to make a careful analysis of your existing telephony infrastructure or your Microsoft Lync Server Enterprise Voice deployment and complete the necessary planning steps.

VoIP gateways

When you're deploying UM in an Exchange organization, you must either install, deploy, and configure a single or multiple VoIP gateways to connect to the PBXs in your telephony network, or install, deploy, and configure Session Initiation Protocol (SIP)-enabled PBXs or IP PBXs.

A VoIP gateway is a third-party hardware device that connects a legacy PBX to your LAN. The VoIP gateway lets the PBX system communicate with the Exchange servers in your organization.

UM relies on the VoIP gateway's ability to translate or convert Time Division Multiplexing (TDM) or circuit-switched based protocols like ISDN and QSIG from a PBX to IP-based or VoIP-based protocols like SIP, Realtime Transport Protocol (RTP), or T.38 for Realtime Fax Transport. The VoIP gateway is integral to the functionality and operation of UM. The VoIP gateway can also connect to PBX systems that use VoIP instead of public switched telephone network (PSTN) circuit-switched protocols.

Choosing the correct VoIP gateway, IP PBX, SIP-enabled PBX, or SBC is only the first part of integrating your telephony network with UM. You must configure those devices to work with UM. In both on-premises and hybrid deployments, you would need to deploy the required Client Access and Mailbox servers, and create and configure all necessary UM components. For Office 365 with hosted voice mail, you're not required to install and configure any server. The components allow you to make the connection from your telephony, circuit-switched network to your IP data network and to enable voice mail for the users in your organization. For details and supported telephony devices, see the following resources:

- Telephony advisor for Exchange 2013
- Configuration notes for supported VoIP gateways, IP PBXs, and PBXs
- Configuration notes for supported session border controllers

Microsoft Lync Server

Unified Messaging can use Microsoft Lync Server to combine voice messaging, instant messaging, enhanced presence, audio/video conferencing, and email into a familiar, integrated communications experience. Providing Enterprise Voice features to the users in your organization by integrating UM and Microsoft Lync Server has the following benefits:

- Enhanced presence notifications across a variety of applications that keep users informed of the availability of contacts.
- Integration of instant messaging, voice messaging, conferencing, email, and other communication modes, which enables users to select the most appropriate mode for the task. Users can also switch from one mode to another as needed.
- Availability of communications alternatives from any location where an Internet connection is available.
- A smart client (Microsoft Lync) for telephony, instant messaging, and conferencing.
- Continuity of the user experience across multiple devices.

The Exchange UM routing component handles voice mail routing between Lync Server and Exchange servers to integrate Lync Server with Unified Messaging features. The Exchange UM routing component found in Lync Server also handles rerouting of voice mail over the PSTN if Exchange servers aren't available. If you have Enterprise Voice deployed at branch office sites, and those sites don't have a resilient WAN link to a central site, a Survivable Branch Appliance that you deploy at the branch site provides voice mail for branch users if a WAN link goes down. When the WAN link is unavailable, the Survivable Branch Appliance does the following:

- Reroutes unanswered calls over the PSTN to an Exchange server in the central site.
- Provides the ability for a user to retrieve voice messages over the PSTN.
- Queues missed call notifications, and then uploads them to the Exchange server when the WAN link is restored.

For more information about Microsoft Lync Server, see Microsoft Lync Server.

⚠**Warning:**
When you're integrating Unified Messaging and Lync Server in an on-premises or hybrid deployment, missed call notifications aren't available to users who have a mailbox located on Exchange 2007 or Exchange 2010 Mailbox servers. A missed call notification is generated when a user disconnects before the call is sent to a Mailbox server.

Telephony concepts and components

Exchange Server 2013 > Unified Messaging > Telephone system integration with UM >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-25*

If you're planning and deploying Microsoft Exchange 2013 Unified Messaging (UM) on your network, you must understand Unified Messaging and telephony networks. This topic provides an overview of telephony infrastructure concepts and components that will help you plan and deploy servers running Exchange 2013 Unified Messaging.

Contents

Overview

In versions of Microsoft Exchange before Exchange Server 2007, the Exchange administrator's main responsibility was managing email messages and, sometimes, managing a network infrastructure. Earlier versions of Exchange didn't have Unified Messaging capabilities. Exchange Server version 5.5, Exchange 2000 Server, and Exchange Server 2003 administrators focused on the Exchange environment and the network infrastructure, and relied heavily on telephony consultants to manage their telephony environment and infrastructure.

Concepts and components

To successfully deploy Unified Messaging in Exchange 2013, you must have a good understanding of basic telephony concepts and telephony components. After you gain a good understanding of telephony basics, you can successfully integrate Exchange 2013 Unified Messaging into an Exchange 2013 organization. Basic concepts and components include the following:

- Circuit-switched and packet-switched networks
- Private Branch eXchange (PBX)
- IP PBX
- Voice over Internet Protocol (VoIP)
- VoIP gateways

Circuit-switched networks

In circuit-switched networks, such as the Public Switched Telephone Network (PSTN), multiple calls are transmitted across the same transmission medium. Frequently, the medium used in the PSTN is copper. However, fiber optic cable might also be used.

A circuit-switched network is a network in which there exists a dedicated connection. A dedicated connection is a circuit or channel set up between two nodes so that they can communicate. After a call is established between two nodes, the connection may be used only by these two nodes. When the call is ended by one of the nodes, the connection is canceled.

Note:

PSTN is a grouping of the world's public circuit-switched telephone networks. This grouping resembles the way that the Internet is a grouping of the world's public IP-based packet-switched networks.

There are two basic types of circuit-switched networks: analog and digital. Analog was designed for voice transmission. For many years, the PSTN was only analog, but today, circuit-based networks such as the PSTN have transitioned from analog to digital. To support an analog voice transmission signal over a digital network, the analog transmission signal must be encoded or converted into a digital format before it enters the telephony WAN. On the receiving end of the connection, the digital signal must be decoded or converted back into an analog signal format.

There are advantages and disadvantages to circuit-switched networks. Circuit-switched networks have several disadvantages. Circuit-switched networks can be relatively inefficient, because bandwidth can be wasted. This isn't the case when VoIP is used on a packet-switched network. VoIP shares the available bandwidth with all other network applications and makes more efficient use of the available bandwidth. Another disadvantage to circuit-switched networks is that you have to provision for the maximum number of telephone calls that will be required for peak usage times and then pay for the use of the circuit or circuits to support the maximum number of calls.

Circuit switching has one big advantage over packet-switched networks. In a circuit-switched network, when you use a circuit, you have the full circuit for the time that you're using the circuit without competition from other users. This isn't the case with packet-switched networks.

Note:

Synchronous Digital Hierarchy (SDH) has become the primary transmission protocol for most PSTN networks. SDH is carried over fiber optic networks.

Return to top

Packet-switched networks

Packet switching is a technique that divides a data message into smaller units called packets. Packets are sent to their destination by the best route available, and then they are reassembled at the receiving end.

In packet-switched networks such as the Internet, packets are routed to their destination through the most expedient route, but not all packets traveling between two hosts travel the same route, even those from a single message. This almost guarantees that the packets will arrive at different times and out of order. In a packet-switched network, packets (messages or fragments of messages) are individually routed between nodes over data links that may be shared by other nodes. With packet switching, unlike circuit switching, multiple connections to nodes on the network share the available bandwidth.

Note:

With circuit switching, all packets go to the receiver in order and along a single path.

Packet-switched networks exist to enable data communication on the Internet throughout the world. A public data network or packet-switched network is the data counterpart to the PSTN.

Packet-switched networks are also found in such network environments as LAN and WAN networks. A WAN packet-switched environment relies on telephone circuits, but the circuits are arranged so that they retain a permanent connection with their endpoint. In a LAN packet-switched environment, such as with an Ethernet network, the transmission of the data packets relies on packet switches, routers, and LAN cables. In a LAN, the switch establishes a connection between two segments only long enough to send the current packet. Incoming packets are saved to a temporary memory area or buffer in memory. In an Ethernet-based LAN, an Ethernet frame contains the payload or data portion of the packet and a special header that includes the media access control (MAC) address information for the source and destination of the packet. When the packets arrive at their destination, they are put back in order by a packet assembler. A packet assembler is needed because of the different routes that the packets may take.

Packet-switched networking has made it possible for the Internet to exist and, at the same time, has made data networks—especially LAN-based IP networks—more available and widespread.

Return to top

PBX

A legacy PBX is a telephony device that acts as a switch for switching calls in a telephony or circuit-switched network.

Note:

A legacy PBX is a PBX that cannot pass IP packets. In many businesses, legacy PBXs have been replaced by IP PBXs.

A PBX is a telephony device used by most medium-size and larger-size companies. A PBX enables users or subscribers of the PBX to share a certain number of outside lines for making telephone calls considered external to the PBX. A PBX is a much less expensive solution than giving each user in a business a dedicated external telephone line. Telephone sets, in addition to fax machines, modems, and many other communication devices, can be connected to a PBX.

The PBX equipment is typically installed at a business's premises and connects calls between the telephones located and installed in the business site. A limited number of outside lines, also known as trunk lines, are typically available for making and receiving calls external to the business from an external source such as the PSTN.

Internal business calls made to external telephone numbers using a PBX are made by dialing 9 or 0 in some systems followed by the external number. An outgoing trunk line is automatically selected to complete the call. Conversely, the calls placed between users within the business don't ordinarily require special dialing digits or use of an external trunk line. This is because the internal calls are routed or switched by the PBX between telephones physically connected to the PBX.

In medium-size and larger-size businesses, the following PBX configurations are possible:

- A single PBX that supports the whole business.
- A grouping of two or more PBXs not networked or connected to each other.
- A grouping of two or more PBXs connected together or networked.

Note:
An Exchange 2013 UM dial plan can span more than one PBX or IP PBX.

Return to top

IP PBX

An IP PBX is a PBX that supports the IP protocol to connect phones using an Ethernet or packet-switched LAN and sends its voice conversations in IP packets. A hybrid IP PBX supports the IP protocol for sending voice conversations in packets, but also connects traditional analog and digital circuit-switched Time Division Multiplex (TDM) telephones. An IP PBX is telephone switching equipment that resides in a private business instead of the telephone company.

IP PBXs are frequently easier to administer than legacy PBXs, because administrators can easily configure their IP PBX services using an Internet browser or another IP-based utility. Plus, no additional wiring, cabling, or patch panels must be installed. With an IP PBX, moving an IP-based telephone is as simple as unplugging a telephone and plugging it in at a new location, instead of the costly service calls to move a telephone from legacy PBX vendors. Additionally, businesses that own an IP PBX don't have the additional infrastructure costs required to maintain and manage two separate circuit-switched and packet-switched networks.

SIP-enabled PBXs

A SIP-enabled PBX is a telephony device that acts as a networking switch for switching calls in a telephony or circuit-switched network. However, the difference between a SIP-enabled PBX and a traditional PBX is that the SIP-enabled PBX can connect to the Internet and use the SIP protocol to make calls over the Internet.

SIP-enabled PBXs use a format for calls that includes a SIP URI containing a global E.164 number and the "user=phone" parameter. For example: sip:+14255551234@contoso.com;user=phone

The E164 number begins with a leading "+", and doesn’t contain a phone-context parameter or any separators. SIP-enabled PBXs support both TCP and UDP. UDP is still widely used with legacy systems. SIP-enabled PBXs also support Mutual Transport Layer Security (mutual TLS) and DNS lookups.

VoIP

Voice over Internet Protocol (VoIP) is a technology that contains hardware and software that enables people to use an IP-based network as the transmission medium for telephone calls. In VoIP,

voice data is sent in packets using IP instead of traditional circuit transmissions or the circuit-switched telephone lines of the PSTN. A VoIP gateway that you connect to your IP network uses VoIP protocols to send voice data packets between Exchange 2013 Client Access and Mailbox servers and a PBX system.

VoIP gateways

A VoIP gateway is a third-party hardware device or product that connects a legacy PBX to your LAN. The VoIP gateway lets the PBX system communicate with your Exchange 2013 Mailbox and Client Access servers running the Microsoft Exchange Unified Messaging and Unified Messaging Call Router services.

Note:
The VoIP gateway can also connect to PBX systems that use VoIP instead of PSTN circuit-switched protocols.

Exchange 2013 Unified Messaging relies on the VoIP gateway's abilities to translate or convert TDM or telephony circuit-switched based protocols like ISDN and QSIG from a PBX to IP-based or VoIP-based protocols like Session Initiated Protocol (SIP), Realtime Transport Protocol (RTP), or T.38 for Realtime Facsimile Transport. The VoIP gateway is integral to the functionality and operation of Unified Messaging.

Important:
After you install the VoIP gateway, IP PBX, or SIP-enabled PBX, you must create a UM IP gateway to represent the physical device. After you create a UM IP gateway, the Client Access and Mailbox servers linked with the UM IP gateway will send a SIP OPTIONS request to the VoIP gateway, IP PBX, or SIP-enabled PBX to ensure that the device is responsive. If the VoIP gateway doesn't respond to the SIP OPTIONS request from the Mailbox server, the Mailbox server will log an event with ID 1088 stating that the request failed. To resolve this issue, ensure that the VoIP gateway, IP PBX, or is available and online and that the Unified Messaging configuration on both the Client Access and Mailbox server is correct.

For more information about IP PBX and PBX configurations, see PBX and IP PBX configurations.

Return to top

For more information

UM protocols, ports, and services

Telephony advisor for Exchange 2013

PBX and IP PBX configurations

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-04-29*

Increasingly, organizations are purchasing, installing, and maintaining the hardware components, for example, Private Branch eXchanges (PBXs) or IP PBXs, that are required to support their own telephony systems. Many organizations are buying their own telephony equipment and training their staff to reduce expenses associated with maintaining their telephony systems and because they want more control over the telephony features they offer.

For an organization to own and maintain their telephony network, they must buy the required telephony hardware components. They must also consider the day-to-day maintenance of the telephony equipment and the training required for their staff to support their telephony system. This topic discusses the different types of telephony business or organizational systems and the telephony hardware components they require. The topic also gives examples of the different types of telephony configurations.

Important:

We recommend that all customers who plan to deploy Microsoft Exchange 2013 Unified Messaging obtain the help of a UM specialist. This will help ensure a smooth upgrade from a legacy voice mail system. Rolling out a new UM deployment or performing an upgrade of an existing voice mail system requires significant knowledge about PBXs, IP PBXs, and Unified Messaging. For more information about who to contact, see the Microsoft PinPointwebsite.

Contents

Overview of telephony systems

In circuit-switched networks, such as the Public Switched Telephone Network (PSTN), multiple calls are transmitted across the same transmission medium. Frequently, the medium that's used in the PSTN is copper. However, fiber optic cable might also be used.

A circuit-switched network is a network in which there exists a dedicated connection. A dedicated connection is a circuit or channel that's set up between two nodes so they can communicate. After a call is established between two nodes, the connection may be used only by these two nodes. When the call is ended by one of the nodes, the connection is canceled.

Different types or categories of telephone systems found in businesses and organizations include a circuit-based network, an IP-based network, or both. Each type of telephone system has distinct

advantages and disadvantages you need to consider when planning and implementing a telephony system.

- **Centrex:** Centrex is a type of telephone service that telephone companies lease to businesses and organizations. A traditional Centrex telephone system eliminates the need for a business or organization to purchase the telephony hardware used onsite to support the organization's telephone system. Typically, Centrex systems are used by small offices that rent Centrex services from a telephone company on a line-by-line and month-by-month basis. Centrex telephony systems are sometimes used by larger organizations, but are most frequently found in government, public, and private organizations. Centrex frequently uses analog telephone lines for the connections to a business or organization. But it can also use T1-circuits with a demultiplexer onsite to support analog and digital telephones or ISDN lines.

In a Centrex-based telephony system, the telephone company's central office acts as the telephone exchange. It's designed specifically to support the needs of a given organization. The central telephone office routes the calls that originate from inside the company to the appropriate internal or external telephone number. Centrex uses the telephone company’s central office exchange to route internal calls back to an extension. For example, with Centrex, the telephone exchange or telephone company's central office knows which extensions are internal. So an employee who's located within the organization's telephony network can dial another employee in the same telephony network or dial plan by using a four-digit extension number. When a call is dialed to the internal telephone extension number, it's forwarded to the telephone company’s central office and then routed back to the extension number that initiated the call.

A variation of a traditional Centrex telephony system is called *IP Centrex*. In an IP Centrex telephone system, the call is sent through a Voice over IP (VoIP) gateway located at a telephone company’s central office or located onsite at a service provider. In this kind of telephone system, the VoIP gateway translates the call into IP-based data packets that can be sent over the Internet or over a VoIP–based network. However, if the call is sent over the Internet, there's typically another VoIP gateway that receives the call and then translates the call back to a traditional circuit-switched call.

Organizations that currently have a traditional Centrex telephone system in place have to install, deploy, and maintain one or more VoIP gateways for Unified Messaging to work correctly. Unified Messaging may require that you install, deploy, and maintain VoIP gateways to work with IP Centrex. Several variables will determine whether you need a VoIP gateway. These variables include the type of telephones used in your organization (analog, digital, or IP) and the protocols supported by the IP Centrex system.

- **Key telephone:** In a Key telephone system, the telephone company’s central office is connected to the organization using standard analog or digital telephone lines. A single telephone extension number is connected to multiple telephones so when a call is placed into the organization using this telephone number, all the telephones associated with that line or extension number will ring at the same time.

With Key telephone systems, individual users share lines across telephones. Therefore, callers won't experience frequent busy signals when they try to call into an organization. Key telephone systems are typically used by small offices where internal call volume is high but external call volume is low.

Key telephone systems have become more sophisticated over time and can work with Unified Messaging if a VoIP gateway is added. However, some less sophisticated systems may not work even if a supported VoIP gateway is used.

- **PBX:** A legacy PBX is a telephony device that switches calls in a telephony or circuit-switched network. A legacy PBX is a PBX that doesn't have a network adapter and can't pass IP packets. Because they can't pass IP packets, some businesses and organizations have replaced legacy PBXs with IP PBXs. For a list of PBXs supported by Unified Messaging, see Telephony advisor for Exchange 2013.

PBXs are used by most medium- and larger-sized companies. A PBX enables users or subscribers of the PBX to share a certain number of outside lines for making telephone calls considered external to the PBX. A PBX is a much less expensive solution than giving each user in a business a dedicated external telephone line. Telephones, in addition to fax machines, modems, and many other communication devices, can be connected to a PBX.

The PBX equipment is typically installed on an organization's premises and connects calls between the telephones located onsite and the telephone company. A limited number of outside lines, also known as trunk lines, are typically available for making and receiving calls external to the business from an external source such as the PSTN.

To enable a legacy PBX to be used with Unified Messaging, you need to deploy a supported VoIP gateway. For a list of supported VoIP gateways, see Telephony advisor for Exchange 2013.

- **IP PBX:** An IP PBX is a PBX that has a network adapter that supports the IP protocol. It's a piece of telephone switching equipment that generally resides in an organization or business instead of being located at a telephone company office. There are two types of IP PBXs: traditional IP PBXs and hybrid IP PBXs. Both traditional IP PBXs and hybrid IP PBXs support the IP protocol for sending voice conversations in packets to VoIP-based telephones. However, hybrid IP PBXs also connect traditional analog and digital telephones.

IP PBXs are frequently easier to administer than legacy PBXs, because administrators can more easily configure IP PBX services using an Internet browser or another IP-based tool. Also, no additional wiring, cabling, or patch panels have to be installed. With an IP PBX, you can move an IP-based telephone by merely unplugging the telephone and plugging it in at a new location. This lets you avoid the costly service calls required to move a telephone from legacy PBX vendors. Additionally, organizations that own an IP PBX don't have to incur the additional infrastructure costs required to maintain and manage separate circuit-switched and packet-switched networks. For a list of IP PBXs supported for Unified Messaging, see Telephony advisor for Exchange 2013.

Return to top

Legacy and traditional PBX configurations

On telephony networks that have legacy or traditional PBXs, a PBX does the following:

- Creates connections or circuits between the telephone sets of two users.
- Maintains the connection as long as the users need the connection.
- Provides information for accounting purposes (for example, meters calls).

In addition to the three functions included in the previous list, PBXs may offer other calling features such as:

- Auto attendants
- Call accounting
- Call pick-up
- Call transfer
- Call waiting
- Conference calling
- Direct Inward Dialing (DID)
- Do Not Disturb (DND)

Although there are several manufacturers of PBXs, they all fit into two basic categories: analog and digital. These types of PBXs are frequently known as *legacy* or *traditional* PBXs.

Typically, PBX systems are connected to the telephone company's central office by using special telephone lines, known as T1- and E1-lines. T1- and E1-lines have multiple channels. These telephone lines are also known as *trunk lines*. They let the central office or the PBX send multiple calls over the same line for better efficiency using a simplified wiring layout. A PBX can also work with analog or ISDN lines.

By correctly configuring your PBX, you can control how many channels or lines you want to configure to receive calls that come from external callers and how many channels or lines to devote to calls that come from callers inside your organization. Configuring the number of channels or lines helps prevent busy signals and lets you configure the number of channels or lines devoted to applications such as call centers. Correctly configuring your PBX is a cost-effective method for managing the channels or lines in your organization because it reduces the number of leased lines required.

A PBX can route a specific dialed telephone number to a specific telephone so users can have their own individual telephone number or extension number. This is known as a Direct Inward Dialing number. When the telephone number is dialed for a user, the telephone company sends the DID number to the PBX by using Dialed Number Identification Service (DNIS). Because the telephone company uses DNIS to send the number, there's no need for operator intervention to route the call. The PBX has the information about the call to correctly route it to the number that was dialed by the caller. For a list of PBXs supported by Unified Messaging, see Telephony advisor for Exchange 2013.

Return to top

Analog and digital PBXs

Analog PBXs send voice and call signaling information, such as the touch tones of a dialed telephone number, as an analog sound. Therefore, the sound is never digitized. To correctly direct the call, the PBX and the telephone company's central office have to listen for the signaling information.

Note:

Touchtone is more technically known as dual tone multi-frequency. When a caller presses a key on a telephone keypad, the telephone produces two separate tones: a high-frequency tone and a low-frequency tone. When a person speaks into the telephone, only a single tone or frequency is emitted. Sending two tones with different frequencies at the same time reduces the possibility that the signaling tones will be interpreted as a human voice or that a human voice will be interpreted as the signaling tones.

Digital PBXs encode or digitize the analog sound into a digital format. Digital PBXs typically encode the voice sounds using a standard industry audio codec like G.711 or G.729. After the digitized voice is encoded, it's sent over a channel by using circuit switching. Circuit switching sets up an end-to-end open connection. It leaves the channel open for the length of the call and for the caller's exclusive use. However, the signaling method that's used by the PBX depends on the manufacturer. PBX manufacturers may have their own proprietary signaling method for call setup.

Note:

Digital PBXs can support both digital and analog trunk lines.

In larger organizations, PBXs make it possible for employees in separate physical locations to contact one another by dialing an extension number for a user. This can be done by using a single PBX or may involve multiple PBXs networked together. PBXs at different office locations can be connected to a single transparent circuit-switched network by using T1- or E1-lines. When these lines connect PBXs together, they are frequently known as *tie lines*. The PBXs communicate with one another across the tie lines using a PBX-to-PBX protocol, such as QSIG. QSIG lets a set of PBXs act as if they are a single PBX.

This kind of PBX environment can also include advanced features, such as call transferring and telephone conferencing. In addition to allowing for advanced features, having two connected PBXs can also save the organization money because long distance charges between employees in the different locations will be reduced. This is because a call made between two employees remains on a tie line between the PBXs and requires that the user dial only an extension number for the other user instead of placing a long distance call.

In a telephony environment that includes a single or multiple analog or digital PBXs, a VoIP gateway is required between the PBX and the Exchange 2013 Client Access and Mailbox servers to convert the circuit-based protocols found on a telephony network into the IP-based protocols found on a data network. For more information about VoIP gateways, see the following topics:

- UM IP gateways
- Connect a VoIP gateway to communicate with a PBX

For a list of VoIP gateways supported for Unified Messaging, see Telephony advisor for Exchange 2013.

Return to top

IP PBX configurations

An IP PBX is a PBX that supports the IP protocol to connect telephones by using an Ethernet or packet-switched LAN. It sends voice conversations in IP or data packets. An IP PBX may have multiple interfaces. These include interfaces for a data network and other interfaces that allow for a connection to a telephony or circuit-switched network.

The development of real-time Internet protocols has made it possible to successfully send voice and fax messages over a data network. Such real-time Internet protocols include the VoIP protocols used with Unified Messaging: Session Initiation Protocol (SIP) over Transmission Control Protocol (TCP) for voice messaging. These protocols have made it possible to successfully send voice and fax messages over a data network. Real-time VoIP protocols are required to send voice messages over a packet-switched or data network so the delivery order and timing of data packets can be maintained and controlled. If these protocols weren't used to maintain and control the delivery and timing of the data packets, a person's voice would be broken up and sound incoherent or the images might appear garbled. For a list of IP PBXs supported for Unified Messaging, see Telephony advisor for Exchange 2013.

Note:
Unified Messaging supports only SIP over TCP.

Traditional IP PBX configurations

A standard or traditional IP PBX contains at least a single network interface that connects to a data network using VoIP protocols. It may also contain additional network interfaces or other telephony interfaces that enable it to connect to an existing telephony network such as the PSTN. The connection to the data network allows for communication with other VoIP hosts located on the data network by using IP data packets. These VoIP hosts include other IP PBXs, VoIP-based telephones, VoIP gateways, and Client Access and Mailbox servers that are running UM services. A traditional IP PBX doesn't support analog or digital telephones. It supports only VoIP telephones.

Because the IP PBX can already connect to a data network and can convert the circuit-based protocols from the PSTN to packet-switched VoIP protocols, a VoIP gateway may not be required to enable communication with Client Access and Mailbox servers on the data network.

IP PBX hybrid configurations

Hybrid IP PBXs can provide analog, digital, and VoIP-based capabilities. If the correct interfaces are installed on an IP PBX and the software that supports multiple types of interfaces is installed correctly, the IP PBX is considered a hybrid IP PBX. An IP PBX hybrid makes it possible to use a mixture of analog, digital, and IP-based telephones.

Most modern IP PBXs can support and provide all three types of voice communication or a traditional IP PBX can be upgraded to a hybrid IP PBX by installing the necessary interfaces and software or firmware updates.

The mixture of analog, digital, and IP-based telephones makes it possible for users in your

organization to use many new features and also provides great flexibility in your telephony environment. Using an IP PBX hybrid also allows for a more gradual migration to a completely VoIP-based telephony environment and voice messaging system for your organization.

Several factors determine whether a VoIP gateway will be required when you connect with Client Access and Mailbox servers. One of these factors is the compatibility of the VoIP protocols used by the IP PBX or hybrid IP PBX and Unified Messaging. If a VoIP gateway isn't required, it will reduce the complexity of the telephony infrastructure, and the support that you must have for Unified Messaging will be simpler.

Return to top

Calling or called party identification

Calling or called party identification is a telephone company service that can tell the person who is receiving the call the telephone number and sometimes the name of the person who is calling and other information about the call. This information is sent over a serial cable by using call signaling. When a call is received by a PBX or IP PBX from a telephone company, the call includes calling identification information such as the following:

- The calling party's number
- The called party's number
- Status codes such as a ring-no-answer, the state or condition of the line, line busy, and call forward always
- The line or port number that's being used for the call
- In telephony, the signaling information is used to exchange information between endpoints on a network to set up, control, and end calls. Several signaling methods used by VoIP gateways and IP PBXs are supported by Unified Messaging. The signaling method that's used depends on the type of device that's being used and the type of signaling method that's used by the telephone company. The most important factor is that the device that's connecting to the telephone company and to the VoIP gateway or IP PBX must support at least one of the signaling methods that enable calling or called party information to be sent and received by callers. For more information about signaling configuration information for a supported VoIP gateway, see Telephony advisor for Exchange 2013.

Although other signaling methods can be used, the two most popular signaling methods are as follows:

- **Simplified Message Desk Interface (SMDI):** SMDI is a protocol that's used to provide signaling, call control, and calling identification information from an interface between a telephone system and a voice mail system. It's used to provide the voice mail system with the information it needs to process an incoming call. Every time an incoming call is sent by using SMDI over a serial interface or RS-232 interface, the information that's sent will identify the line or port, the type of call, and the calling or called party numbers. The SMDI cable connects from a device such as a PBX to a serial connection on the VoIP gateway. However, SMDI is also used with IP PBXs. The SMDI protocol allows for a maximum of only 10 digits for each calling and called number. This is

a limitation of the protocol and can't be changed.

- **In-band:** In-band signaling allows for the exchange of signaling, call control, and calling identification information from a telephone company. This information is sent over the same channel and in the same band (300 Hz to 3.4 kHz) as the voice and other sounds that are being made during the call. For example, when a user places a call by using DTMF or touchtone dialing and talks to the called party, both the touchtone and the voice conversation use the same channel and band. In-band signaling is less secure because the control signals are exposed to the user and is a less popular signaling method than SMDI. In-band signaling applies only to Channel Associated Signaling (CAS).

Return to top

Connect UM to your telephone system

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-30*

Unified Messaging (UM) combines voice messaging and email messaging into one mailbox that can be accessed from many different devices. Users can listen to their voice mail messages from their email Inbox or by using Outlook Voice Access from any telephone.

When you're deploying UM in a Microsoft Exchange organization, you must install, deploy, and configure a single or multiple Voice over IP (VoIP) gateways to connect to the Private Branch eXchanges (PBXs) in your telephony network or install, deploy, and configure Session Initiation Protocol (SIP)-enabled PBXs or IP PBXs. If you're upgrading your current voice mail system, you'll need to deploy the devices that connect to your telephony network, install your Exchange Client Access and Mailbox servers, and then create the required UM components that allow your telephony network to connect to your data network. This enables incoming calls from the telephony network to connect to your VoIP gateways, IP PBXs, or SIP-enabled PBXs, and those devices to connect to your Exchange organization.

If you're installing, deploying, and configuring either Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server, you won't use VoIP gateways, IP PBXs, or SIP-enabled PBXs to directly connect to Exchange Unified Messaging. Instead, a Lync Mediation Server and VoIP gateway or an advanced VoIP gateway that has the functionality of a Mediation Server and a VoIP gateway will allow you to connect to Exchange UM. UM users that are enabled for Enterprise Voice can retrieve, listen to, and respond to voice messages and make outbound calls. They also have access to other Lync-related features including presence and Instant Messaging (IM) by using Office Communicator or a Lync client.

The following information will help you set up and deploy UM and enable voice mail features for

users in your organization:

- Connect a VoIP gateway, IP PBX, or session border controller to UM Learn how to connect VoIP gateways or IP PBXs to UM.
- Telephony advisor for Exchange 2013 Learn about supported VoIP gateways, IP PBXs, and PBXs.
- Configuration notes for supported VoIP gateways, IP PBXs, and PBXs Learn how to set up your VoIP gateways, IP PBXs, and PBXs.

Telephony advisor for Exchange 2013

Unified Messaging > Telephone system integration with UM > Connect UM to your telephone system >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

Unified Messaging (UM) requires that you integrate Microsoft Exchange with the existing telephony system for your organization. A successful deployment requires you to make a careful analysis of your existing telephony infrastructure and to perform the correct planning steps to deploy Unified Messaging.

The planning phase can be a significant challenge to Exchange administrators who have little or no experience with a telephony network. To help address this challenge, see Resources to Help with Your UM Deployment later in this topic.

To see the supported VoIP gateways for Unified Messaging, determine whether your PBX is supported using a specific VoIP gateway model or manufacturer, whether your IP PBX is supported using a direct SIP connection, or to see supported session border controllers (SBCs) for Exchange Online UM, click one of the following links:

- Supported VoIP gateways
- Supported PBXs when using an AudioCodes VoIP gateway
- Supported PBXs when using a dialogic VoIP gateway
  - PBXs supported when using a DMG1000 series Media Gateway
  - PBXs supported when using a DMG 2000 series Media Gateway
  - PBXs supported when using a DMG3000 series Media Gateway
- Supported IP PBXs
- Supported IP PBXs when using SIP media gateways
- Exchange Unified Messaging, Office Communications Server 2007 R2, and Microsoft Lync Server

Resources to help with your UM deployment

It's challenging to create guidelines for deploying telephony networks. They can be very different from one another because they can include VoIP gateways, IP PBXs, and PBXs with different

configuration settings, firmware, and requirements. However, several resources are available to help you successfully deploy Unified Messaging:

- **Unified Messaging specialists** UM specialists are systems integrators who have received technical training about Exchange Unified Messaging conducted by the Exchange engineering team. To help ensure a smooth transition to Unified Messaging from legacy voice mail systems, Microsoft recommends that all customers engage a UM specialist. For contact information, visit Microsoft Exchange Server 2013 Unified Messaging (UM) Specialists or Microsoft Pinpoint for Unified Messaging.
- **Configuration Notes for Supported VoIP Gateways, IP PBXs and PBXs** These configuration notes contain settings and other information that's very useful when you're configuring VoIP gateways, IP PBXs, and PBXs to communicate with the Unified Messaging servers that are on your network. For more information, see Configuration notes for supported VoIP gateways, IP PBXs, and PBXs.
- **Configuration Notes for Supported Session Border Controllers** These configuration notes contain settings and other information that's very useful when you're configuring session border controllers (SBCs) to communicate with the Unified Messaging servers in hybrid and Exchange Online UM deployments. For more information, see Configuration notes for supported session border controllers.

Before you engage a Unified Messaging specialist, you should be able to answer key questions that they'll ask. Having the answers to the following questions will help make the conversation between you and the UM specialist productive:

- How many existing telephone or voice mail users, or both, are in your organization?
- How many users do you intend to provide with Unified Messaging?
- Which PBX or PBXs do you intend to use for integration with Unified Messaging?
- How many PBXs does your organization have? Specify the vendors, types (circuit- or IP-based), models, and firmware versions.
- Are the PBXs networked, and are they centralized or located in multiple locations?
- What voice mail system or systems does your organization currently use? Specify the vendors, types, models, and firmware versions.
- How are the voice mail systems integrated into your PBXs (Analog, T1/E1, PRI, Digital set emulation, VoIP, other)?
- Are you currently using voice networking?
- What type of fax system or systems does your organization use, and does the fax system or systems support inbound fax routing to Exchange?
- Does your organization use automated attendants?
- Do you need support for phone-only users, that is, users who won't have email access?

Supported VoIP gateways

Integrating Unified Messaging with PBXs requires you to use one or more VoIP gateways to translate the circuit-switched protocols that are used by TDM-based PBXs to IP-based, packet-switched protocols that are used by Unified Messaging. VoIP gateway vendors with several models

of VoIP and media gateways have been tested and are supported for Unified Messaging.

Interoperability testing of Unified Messaging with VoIP gateways, IP PBXs, and SBCs is now integrated with the Microsoft Unified Communications Open Interoperability Program. For more information, see Microsoft Unified Communications Open Interoperability Program.

The Microsoft Unified Communications Open Interoperability Program qualification program for VoIP gateways, IP PBXs, and advanced VoIP gateways ensures that customers have a seamless setup and support experience when they're using qualified telephony VoIP gateways and IP PBXs with Microsoft Unified Communications software. Only products that meet rigorous and extensive testing requirements and conform to the specifications and test plans receive qualification.

For details about configuring supported VoIP gateways, IP PBXs, PBXs, and SBCs, see one of the following resources:

- Configuration notes for supported VoIP gateways, IP PBXs, and PBXs
- Configuration notes for supported session border controllers

Interoperability was verified for the following VoIP gateway vendors:

- AudioCodes
- Dialogic
- The following table shows the VoIP gateway vendor, the VoIP gateway model, and the protocols that are supported by each model.

Supported VoIP gateways for Unified Messaging

Vendor	Model	Supported protocols
AudioCodes	MediaPack 114/8 FXO	• Analog with In-Band DTMF • Analog with SMDI
AudioCodes	Mediant 1000	• Analog with In-Band DTMF • Analog with SMDI • BRI Q.SIG • T1/E1 Q.SIG • IP-to-IP
AudioCodes	Mediant 2000	• T1/E1 CAS • T1/E1 Q.SIG • IP-to-IP
Dialogic	DMG1000PBXDNIW	Digital Set Emulation
Dialogic	DMG1000LSW	• Analog with In-Band DTMF • Analog with SMDI
Dialogic	DMG2000	• T1 CAS • T1/E1 Q.SIG

Dialogic	DMG3000	• BRI Q.SIG
NET	VX1200	• T1 Q.SIG
Quintum	Tenor DX Series	• T1 Q.SIG

Supported PBXs when using an AudioCodes VoIP gateway

The following table shows the PBXs that are supported using AudioCodes VoIP gateways, including MediaPack-114 FXO, MediaPack-118 FXO, and Mediant 2000.

PBXs supported with an AudioCodes VoIP gateway

PBX manufacturer	PBX model/type	AudioCodes model "x" - replace with 4 or 8 per need "y" - replace with 1, 2, 4, 8 or 16 per need
Alcatel	OmniPCX 4400	• MediaPack 11x/FXO/AC/SIP-0 • Mediant2000/ySpans/SIP
Aastra	M1000, M2000	• Mediant2000/ySpans/SIP
Avaya	Definity G3	• MediaPack 11x/FXO/AC/SIP-0 • Mediant1000/ySpans/SIP • Mediant2000/ySpans/SIP
Avaya	Magix/Merlin	• MediaPack 11x/FXO/AC/SIP-0
Avaya	S8300	• MediaPack 11x/FXO/AC/SIP-0 • Mediant1000/ySpans/SIP • Mediant2000/ySpans/SIP
Avaya	S8700	• MediaPack 11x/FXO/AC/SIP-0 • Mediant1000/ySpans/SIP • Mediant2000/ySpans/SIP
Avaya	IP Office	• MediaPack 11x/FXO/AC/SIP-0 • Mediant2000/ySpans/SIP
Cisco	CallManager 4.x	• Mediant1000/IP-to-IP • Mediant2000/IP-to-IP
NEC	Electra Elite	• MediaPack 11x/FXO/AC/SIP-0

NEC	NEAX2400	• MediaPack 11x/FXO/AC/SIP-0 • Mediant2000/ySpans/SIP/ RS232
NeXspan	S	• MediaPack 11x/FXO/AC/SIP-0
Nortel	Communication Server-1000M, 1000S, 1000E	• Mediant1000/ySpans/SIP • Mediant2000/ySpans/SIP
Nortel	Meridian 11c, 51c, 61c, 81c	• Mediant1000/ySpans/SIP • Mediant2000/ySpans/SIP
Panasonic	KX-TES824, KX-TEA308	• MediaPack 11x/FXO/AC/SIP-0
Panasonic	KX-TDA30, KX-TDA100, KX-TDA200, KX-TDA600	• MediaPack 11x/FXO/AC/SIP-0
Shortel	IP Telephony System	• MediaPack 11x/FXO/AC/SIP-0
Siemens	HiCom 150E	• MediaPack 11x/FXO/AC/SIP-0
Siemens	HiPath 3550	• MediaPack 11x/FXO/AC/SIP-0
Siemens	HiPath 4000	• MediaPack 11x/FXO/AC/SIP-0 • Mediant1000/ySpans/SIP • Mediant2000/ySpans/SIP
Tadiran Telecom	Coral Flexicom, Coral IPX	• MediaPack 11x/FXO/AC/SIP-0 • Mediant1000/ySpans/SIP • Mediant2000/ySpans/SIP

Supported PBXs when using a Dialogic VoIP gateway

Each Dialogic VoIP gateway model supports different PBXs. The following tables show the PBX manufacturer and model and which Dialogic VoIP gateway can be used. Each VoIP gateway uses different signaling methods, densities, and protocols.

PBXs supported when using a DMG1000 series Media Gateway

The following table shows the PBXs that are supported with the low-density Dialogic Media

Gateway (DMG1000). However, when an analog DMG1000 is used, supplemental signaling (RS232 SMDI, MD110, MCI protocols, or Inband DTMF signaling) is required.

PBXs supported when using a low-density Dialogic DMG1000 series VoIP gateway

PBX manufacturer	PBX model/type	DMG model and additional signaling
Aastra	Aastra MD110 (formerly Ericsson MD110)	DMG1008LSW Analog connectivity using the MD110 RS232 protocol
Alcatel	Omni PCX 4400	DMG1008LSW
Avaya	Definity G3 S8100, S8300, S8700, and S8710 (Communications Mgr SW V2.0 or later versions)	DMG1008DNIW
Intercom		DMG1008LSW Analog connectivity using SMDI serial protocol
Mitel	SX-200D, SX-200 Light, SX-2000 Light, SX-2000 S, SX-2000 VS, SX-200 ICP	DMG1008MTLDNIW
NEC	2000, 2400, 2400 IPX	DMG1008DNIW
Nortel	Meridian 1 - Option 11, 21, 21A, 51, 61, 71, and 81 Meridian SL1 - Generic X11, Release 15 or later versions Nortel Communication Server - 1000M, 1000S, 1000E with V3.0 or later versions	DMG1008DNIW
Nortel	SL 100	DMG1008LSW

		Analog connectivity using SMDI serial protocol
Siemens	HiCom 300E CS	DMG1008DNIW
Siemens	HiCom 300E (European)	DMG1008LSW Analog connectivity using Inband DTMF signaling
Siemens/ROLM	8000 (SW release 80003 or later versions) 9000 (All versions) 9751 (All versions of SW release 9005) 9751 (SW release 9006.4 or later versions)	DMG1008RLMDNIW
Siemens	HiPath 4000	DMG1008LSW
Toshiba	CTX (SW version AR1ME021.00)	DMG1008LSW
Others	Various	DMG1008LSW Analog connectivity using either Inband DTMF or SMDI

PBXs supported when using a DMG 2000 series Media Gateway

The following table shows the PBXs that are supported with the T1/E1 Dialogic Media Gateway (DMG2000). The DMG2000 gateway, which comes in single span (DMG2030DTIQ), dual span (DMG2060DTIQ), or quad span (DMG2120DTIQ) densities, supports the following protocols:

- T1 CAS
- T1 Q.SIG
- E1 Q.SIG
- T1 NI-2

- T1 5ESS
- T1 DMS100

If Channel Associated Signaling (CAS) signaling is used, supplemental signaling (RS232 SMDI, MD110, MCI protocols, or Inband DTMF signaling) is required. If Q.SIG signaling is used, the PBX must support the supplemental services that are associated with calling and called party information and the call transfer capabilities required by Unified Messaging.

PBXs supported with the DMG2000 Media Gateway

PBX manufacturer	PBX model/type	Required software version	Protocol and additional signaling
Alcatel	Omni PCX 4400	Version 3.2.712.5	T1 Q.SIG E1 Q.SIG
Avaya	Definity G3	Version 3 or later	T1 CAS
Avaya	S8500	Manager SW V2.0 or later versions	T1 CAS T1 Q.SIG E1 Q.SIG
Ericsson	MD110	Release MX1 TSW R2A (BC13)	E1 Q.SIG
Intercom			CAS (w/ SMDI serial protocol)
NEC	2400 IMX	Release 5200 Dec. 92 1b or later versions	CAS (w/ MCI serial protocol)
NEC	2400 IPX	R17 Release 03.46.001	T1 Q.SIG
Nortel	Meridian 1 - Option 11	Release 15 or later versions, and options 19 and 46 are required	T1 Q.SIG E1 Q.SIG
Nortel	Communication Server 1000	Version 2121, Release 4	T1 Q.SIG E1 Q.SIG
Siemens	HiCom 300E CS	Release 9006.4 or later	T1 CAS

		(Note: North American software load only)	
Siemens	HiPath 4000	V2 SMR 9 SMPO	T1 Q.SIG E1 Q.SIG
Mitel	SX-2000 S, SX-2000 VS	LW 34	T1 Q.SIG E1 Q.SIG
Mitel	3300	Version 5.1.4.8	T1 Q.SIG E1 Q.SIG

PBXs supported when using a DMG4008BRI series Media Gateway

The DMG4000 series Media Gateway comes with several TDM interface options. The DMG4008BRI supports 4-port/8-channel densities and supports the following protocols:

- ISDN BRI Q.SIG
- ETSI-DSS1 (Euro ISDN)
- NET 3 (Belgium)
- VN3 (France)
- 1TR6 (Germany)
- INS-64 (Japan)
- 5ESS Custom (North America - AT&T)
- National ISDN (NI1 - North America)

The following table shows the PBXs that are supported using a Dialogic 4000 Media Gateway Series (DMG4008).

PBXs supported using a DMG4008BRI Media Gateway

PBX manufacturer	PBX model/type	Required software version	Protocol and additional signaling
Siemens	HiCom 300	SA300-V3.05	BRI-Q.SIG (ECMAV2)
Siemens	HiPath 4000	S.0 B4400	BRI-Q.SIG (ECMAV2)

Supported IP PBXs

IP PBXs are also supported by Unified Messaging. The following table shows the IP PBXs that are

supported using a direct SIP connection to Unified Messaging.

IP PBXs supported when using a direct SIP connection

PBX manufacturer	PBX model/type	Required software version
Aastra	MX-ONE	4.0
Avaya	Aura	5.2.1 with Service Pack 5 (SP5)
Avaya	Communication Server 2100	CS2100 SE13
Cisco	Call Manager, Unified Communications Manager	5.1, 6.x, 7.0 and8.0

IP PBXs supported when using SIP media gateways

IP PBXs using SIP media gateways are also supported by Unified Messaging. The following table shows the IP PBXs that are supported using IP to IP capabilities of SIP media gateways to connect to Unified Messaging.

IP PBXs supported when using a SIP media gateway

PBX manufacturer	PBX model/type	SIP gateway model
Cisco	Call Manager 4.x	AudioCodes Mediant 1000/2000 (IP-to-IP enabled)

Exchange Unified Messaging, Office Communications Server 2007 R2, and Microsoft Lync Server

For on-premises and hybrid deployments, Exchange Unified Messaging can be deployed together with Microsoft Office Communications Server 2007 R2, Microsoft Lync Server 2010 or Lync Server 2013 to provide voice messaging, Instant Messaging (IM), enhanced user presence, audio-video conferencing, and an integrated email and messaging experience for users in your organization. For more information, see:

- Deploying Exchange 2013 UM and Lync Server overview
- Microsoft Lync Server 2013

To find out more about the Microsoft Unified Communications Open Interoperability Program for enterprise telephony infrastructure, including finding qualified SIP PSTN gateways and IP PBXs and the process for telephony infrastructure vendors to join and participate in the program, see Microsoft Unified Communications Open Interoperability Program.

Configuration notes for supported VoIP gateways, IP PBXs, and PBXs

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-23*

This page provides links to configuration notes that have been created and tested by Microsoft or a VoIP gateway partner. When Microsoft or a partner deploys Unified Messaging with a new VoIP gateway and PBX or IP PBX configuration, the prerequisites and configuration settings are documented. This information is used to create a configuration note.

Each PBX configuration note contains information about how to deploy Unified Messaging with a specific telephony configuration, and includes the manufacturer, model, and firmware version for the VoIP gateways, IP PBXs, or PBXs. In addition, each PBX configuration note includes other information, such as:

- Contributors in authoring the configuration note.
- Detailed prerequisites, including the following:
  - Features that have to be enabled or disabled on the PBX.
  - Specialized hardware that has to be installed.
  - Whether a VoIP gateway is required.
  - Features that must be present on the VoIP gateway, if one is needed.
  - Specific cabling requirements between an IP gateway and a PBX.
  - A list of Unified Messaging features that may not be available with a given telephony configuration.

To find out more about the Microsoft Unified Communications Open Interoperability Program for enterprise telephony infrastructure, including finding qualified SIP PSTN gateways and IP PBXs and the process telephony infrastructure vendors can use to join and participate in the program, see Microsoft Unified Communications Open Interoperability Program.

VoIP gateway, IP PBX, and PBX configuration notes

Microsoft is working with VoIP gateway partners, AudioCodes and Dialogic, to add to the list of PBXs that are tested. Because we are currently testing many combinations of telephony components, this topic is updated frequently. Please check back if you can't locate the appropriate configuration note for your deployment.

The following configuration notes are available:

• Aastra • Alcatel • Avaya • Cisco • Inter-Tel • Intecom • Mitel • NEC	• NeXspan • Nortel • Panasonic • Rolm • ShoreTel • Siemens • Tadiran • Toshiba

Aastra

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
Aastra MD110 (formerly Ericsson MD110)	MX1 TSW R2A (aka BC13)	Analog – Serial MD110	Dialogic	DMG1008LSW	Dialogic	Download
Aastra MD110 (formerly Ericsson MD110)	MX1 TSW R2A (aka BC13)	E1 Q.SIG	Dialogic	DMG2030DTIQ	Dialogic	Download
Aastra MX-ONE	4.0	Direct SIP Connection	N.A.	N.A.	Aastra	Download

Alcatel

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
OmniPCX 4400	R4.2-d2.304-4-h-il-c6s2	Analog – In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download

Avaya

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
Aura	Communication Manager 5.2.1 with SP 5 Session Manager 5.2.	Direct SIP Connection	N.A.	N.A.	Avaya	Download
CS 2100	CS 2100 SE13	Direct SIP Connection	N.A.	N.A.	Avaya	Download
Definity G3	R009i.05.122.4	Digital Set Emulation (DNI7434)	Dialogic	DMG1008D NIW	Dialogic	Download
Definity G3	R013i.01.1.628.7	Analog – In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download
Definity G3	R013i.01.1.628.7	T1 CAS – In-Band DTMF	AudioCodes	Mediant 2000	AudioCodes	Download
Definity G3	R013i.01.1.628.7	T1 Q.SIG	AudioCodes	Mediant 1000/2000	AudioCodes	Download
Definity G3	R013i.01.1.628.7	E1 Q.SIG	AudioCodes	Mediant 1000/2000	AudioCodes	Download
Merlin Magix	Release 1.5 v.6.0	Analog – In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download
S8300	G3xV11 Communication Manager	Analog – In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download

	1.3					
S8300	R013x.01.2.632.1	T1 CAS – In-Band DTMF	AudioCodes	Mediant 2000	AudioCodes	Download
S8300	R013x.01.2.632.1	E1 Q.SIG	AudioCodes	Mediant 1000/2000	AudioCodes	Download
S8500	Communication Manager 3.0 (R013x00.1.346.0)	E1 Q.SIG	Dialogic	DMG2030DT IQ	Dialogic	Download
S8500	Communication Manager 3.0 (R013x00.1.346.0)	T1 CAS – In-Band DTMF	Dialogic	DMG2030DT IQ	Dialogic	Download
S8500	Communication Manager 3.0 (R013x00.1.346.0)	T1 Q.SIG	Dialogic	DMG2030DT IQ	Dialogic	Download
S8700	R011x.02.0.110.4	E1 Q.SIG	AudioCodes	Mediant 1000/2000	AudioCodes	Download

Cisco

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
Cisco Call Manager 4.x	4.x	IP-to-IP	AudioCodes	AudioCodes	AudioCodes	Download

Cisco Call Manager 5.1	5.1.0.9921-12	Direct SIP Connection	N.A.	N.A.	Microsoft	Download
Cisco Unified Communications Manager 6.0 and 6.1	6.x	Direct SIP Connection	N.A.	N.A.	Microsoft	Download
Cisco Unified Communications Manager 7.0	7.0.2.20000-5	Direct SIP Connection	N.A.	N.A.	Microsoft	Download
Cisco Unified Communications Manager 8.0	8.0.3.20000-5	Direct SIP Connection	N.A.	N.A.	Microsoft	Download

Inter-Tel

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
5000	Inter-Tel 5000 v2.1	T1 CAS – In-Band DTMF	AudioCodes	Mediant 2000	AudioCodes	Download
Axxess	Axxess V9.0	T1 CAS – In-Band DTMF	AudioCodes	Mediant 2000	AudioCodes	Download

Intecom

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
PointSpan M6880	40PS3.5.K.2	T1 CAS - SMDI	AudioCodes	Mediant 2000	AudioCodes	Download

Mitel

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
3300	5.1.4.8	E1 Q.SIG	Dialogic	DMG2030DTIQ	Dialogic	Download
3300	5.1.4.8	T1 Q.SIG	Dialogic	DMG2030DTIQ	Dialogic	Download
SX2000	5.0.24	Digital Set Emulation (DNISS430)	Dialogic	DMG1008MTLDNIW	Dialogic	Download
3300	7	T1 Q.SIG	AudioCodes	Mediant 1000/2000	AudioCodes	Download

NEC

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
Electra Elite 192	SP034V4.5	Analog – In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download
NEAX2400IMX	version 7400	T1 CAS - serial MCI	Dialogic	DMG2030DTIQ	Dialogic	Download
NEAX2400IMX & IPX	version 7400	Digital Set Emulation (DNIDtermIII)	Dialogic	DMG1008DNIW	Dialogic	Download
NEAX2400IPX	Ver. R18.06.24.000	T1 CAS – serial MCI	AudioCodes	Mediant 2000	AudioCodes	Download

NEAX2400IPX	Ver. R18.06.24.000	Analog – serial MCI	AudioCodes	MP-11x FXO	AudioCodes	Download

NEAX 2400IPX	Ver.17 Rel.03.46.001	T1 Q.SIG – serial MCI	Dialogic	DMG2030DTIQ	Dialogic	Download			

NeXspan

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
S	RMS1 version R1.3 E1TA	Analog – In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download

Nortel

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
CS1000	3.0 & 4.5	E1 Q.SIG	AudioCodes	Mediant 1000/2000	AudioCodes	Download
Meridian 81C	4.5	E1 Q.SIG	AudioCodes	Mediant 2000	AudioCodes	Download
Meridian 81C	4.5	T1 Q.SIG	AudioCodes	Mediant 1000/2000	AudioCodes	Download
Option11c	Release 25	Digital Set Emulation (DNI2616)	Dialogic	DMG1008DNIW	Dialogic	Download
Option11c	Release 25	T1 Q.SIG	Dialogic	DMG2030DTIQ	Dialogic	Download

Option11c	Release 25	E1 Q.SIG	Dialogic	DMG2030DT IQ	Dialogic	Download
CS-1000M (Succession)	Release 25.40	E1 Q.SIG	Dialogic	DMG2030DT IQ	Dialogic	Download

Panasonic

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
KX-TDA200	001-001	Analog - In-Band DTMF	AudioCodes	Mediant 1000	AudioCodes	Download
KX-TDA200	3	Analog - In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download
KX-TES824	2.0.2	Analog - In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download

Rolm

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
9751	9005	Digital Set Emulation (DNIRP400)	Dialogic	DMG1008RL MDNIW	Dialogic	Download

ShoreTel

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
IP Telephony System	6.1	Analog – SMDI	AudioCodes	MP-11x FXO	AudioCodes	Download
IP Telephony	7.5	Analog –	AudioCodes	Mediant	AudioCodes	Download

System		SMDI		1000		

Siemens

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
HiCom 150E	Rel. 2.2	Analog – In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download
HiCom 300	SA300-V3.05	BRI QSIG	Dialogic	DMG3000	Dialogic	Download
HiCom 300	9006.4S MR3	Digital Set Emulation (DNIOptiset)	Dialogic	DMG1008DNIW	Dialogic	Download
HiCom 300	9006.4S MR3	T1 CAS - In-Band DTMF	Dialogic	DMG2030DTIQ	Dialogic	Download
HiPath 3550	Rel. 3	Analog – In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download
HiPath 4000	Ver 3.0 SMR5 SMP4	Analog – In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download
HiPath 4000	SA300-V3.05	BRI QSIG	Dialogic	DMG3000	Dialogic	Download
HiPath	Ver 3.0	T1 Q.SIG	AudioCo	Mediant	AudioCodes	Download

4000	SMR5 SMP4		des	1000/2000		
HiPath 4000	Version 2.0 SMR9 SMP0	Analog - In-Band DTMF	Dialogic	DMG1008LSW	Dialogic	Download
HiPath 4000	Version 2.0 SMR9 SMP0	T1 Q.SIG	Dialogic	DMG2030DTIQ	Dialogic	Download

Tadiran

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
Coral Flexicom	14.67.49	Analog – In-Band DTMF	AudioCodes	MP 11x FXO	AudioCodes	Download
Coral Flexicom	14.67.49	BRI QSIG	AudioCodes	Mediant 1000	AudioCodes	Download
Coral Flexicom	14.67.49	E1 CAS - In-Band DTMF	AudioCodes	Mediant 2000	AudioCodes	Download
Coral Flexicom	14.67.49	E1 Q.SIG	AudioCodes	Mediant 1000/2000	AudioCodes	Download
Coral IPX	14.67.49	Analog – In-Band DTMF	AudioCodes	MP-11x FXO	AudioCodes	Download
Coral IPX	14.67.49	BRI QSIG	AudioCodes	Mediant 1000	AudioCodes	Download
Coral IPX	14.67.49	E1 CAS – In-Band DTMF	AudioCodes	Mediant 2000	AudioCodes	Download

Coral IPX	14.67.49	E1 QSIG	AudioCodes	Mediant 1000/2000	AudioCodes	Download

Toshiba

PBX model	PBX software release	Protocol	Gateway vendor	Gateway model	Configuration author	Configuration file download
CTX	AR1ME021.00	Analog – SMDI	Dialogic	DMG1008LSW	Dialogic	Download
CTX	AR1ME021.00	Analog – In-Band DTMF	Dialogic	DMG1008LSW	Dialogic	Download

Connect a VoIP gateway, IP PBX, or session border controller to UM

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-16*

You must configure the Voice over IP (VoIP) gateways and IP Private Branch eXchanges (PBXs) correctly when you deploy Unified Messaging (UM) for your organization. If you're deploying UM in a hybrid environment, you'll also need to correctly configure your session border controllers (SBCs). To do this, you need to configure the interface or interfaces of the VoIP gateways, IP PBXs, and SBCs to communicate with Client Access servers running the Microsoft Exchange Unified Messaging Call Router service and Mailbox servers running the Microsoft Exchange Unified Messaging service.

Important:

When you perform administrative tasks for a VoIP gateway, IP PBX, or SBC using a web browser, the HTTP requests sent over the network aren't encrypted. To increase the level of security for the VoIP gateways, IP PBXs, or SBCs on your network, use Internet Protocol security (IPsec) or Secure Sockets Layer (SSL) to help protect the administrative credentials and data transmitted over the network. We also recommend that you use a strong authentication mechanism and complex administrative passwords to protect the administrative credentials for the device.

Telephony IP device interfaces

There are several types of ports or interfaces that you must configure to enable communication between a PBX, VoIP gateway, IP PBX, or SBC and the Client Access and Mailbox servers on your network. When you configure a VoIP gateway, IP PBX, or SBC, you must consider whether the device is analog, digital, or analog and digital.

If the VoIP gateway interface that connects to a PBX is analog, you must correctly configure the appropriate settings to enable the VoIP gateway to communicate with your Client Access and Mailbox servers. For both IP PBXs and SBCs, you must also correctly configure the IP interfaces to enable these devices to communicate with Client Access and Mailbox servers. All of these devices have different types of connections or ports that are available depending on the device model and vendor.

To enable communication with the Client Access and Mailbox servers on your network:

- For VoIP gateways, you must configure the telephony interfaces to communicate with your PBXs, and you must configure the IP interface for the device.
- For IP PBXs, you must configure the circuit-based and network or IP-based connections.
- For SBCs, you must configure both IP interfaces, one for your network and the other interface that connects over the Internet or a dedicated WAN connection.

The following is a list of resources found on the Exchange TechCenter that provide information that can help you correctly configure your VoIP gateways, IP PBXs, and SBCs:

- **Supported IP gateway, IP PBX, and PBX documentation** Telephony advisor for Exchange 2013 includes configuration files and setup information that you can use when you configure VoIP gateways, IP PBXs, PBXs, and SBCs.
- **Configuration and technical notes** Configuration notes for supported VoIP gateways, IP PBXs, and PBXs includes configuration files and setup information that you can use when you configure VoIP gateways, IP PBXs, and PBXs.
- **Configuration notes for Exchange UM online** Configuration notes for supported session border controllers includes configuration files and setup information that you can use when you configure SBCs.

Unified Messaging specialists are available to assist you in configuring your telephony and IP-based network devices. A Unified Messaging specialist can help make sure that there's a smooth transition to Unified Messaging from a legacy or third-party voice mail system or help you plan and deploy a new voice mail system with Exchange Unified Messaging. Deploying a new voice mail system or upgrading a legacy one requires significant knowledge about VoIP gateways, IP PBXs, PBXs, and Unified Messaging. For more information about how to contact a Unified Messaging specialist, see Microsoft Exchange Server 2013 Unified Messaging (UM) Specialists or certified UM partners at Microsoft Pinpoint.

After you configure a VoIP gateway, IP PBX, or SBC IP interface, you must create and configure a UM IP gateway to represent each device that you've deployed. For more information about how to create a UM IP gateway, see Create a UM IP gateway.

After you create a UM IP gateway, the Client Access and Mailbox servers associated with the UM IP gateway send a SIP OPTIONS request to the VoIP gateway, IP PBX, or SBC to ensure that it's responsive. If the VoIP gateway, IP PBX, or SBC doesn't respond, the Mailbox server will log an event with ID 1088 stating that the request failed. To resolve this issue, make sure that the VoIP gateway, IP PBX, or SBC is available and online and the Unified Messaging configuration is correct.

A Client Access server and a Mailbox server will communicate only with a VoIP gateway, IP PBX, or SBC that's listed as a trusted Session Initiation Protocol (SIP) peer. An event with ID 1175 will be logged when multiple DNS hosts share the same IP address. This event may occur if you've configured your DNS zones with FQDNs for the VoIP gateways on your network. Unified Messaging protects against unauthorized requests by retrieving the internal URL of the Unified Messaging Web Services virtual directory that's located on the Mailbox server and then using the URL to build the list of FQDNs for the trusted SIP peers. After two FQDNs are resolved to the same IP address, this event will be logged.

Note:

You must restart the Microsoft Exchange Unified Messaging service if a VoIP gateway, IP PBX, or SBC is configured to have an FQDN and the DNS record of the VoIP gateway, IP PBX, or SBC is changed after the service has been started. If you don't restart the service, the Mailbox server won't be able to locate the VoIP gateway, IP PBX, or SBC. This occurs because a Mailbox server maintains a cache for all VoIP gateways, IP PBXs, or SBCs in memory, and DNS resolution is performed only when the service is restarted or when the configuration of a VoIP gateway, IP PBX, or SBC has changed.

Connect UM to a supported VoIP gateway

Unified Messaging > Telephone system integration with UM > Connect UM to your telephone system >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-19*

When you're setting up Unified Messaging (UM), you must configure the Voice over IP (VoIP) gateways, IP PBXs, SIP-enabled PBXs, or session border controllers (SBCs) on your network to communicate with the Client Access servers running the Microsoft Exchange Unified Messaging Call Router service and the Mailbox servers running the Microsoft Exchange Unified Messaging service in your Exchange organization. You must also configure the Client Access and Mailbox servers to communicate with the VoIP gateways, IP PBXs, SIP-enabled PBXs, or SBCs.

Note:

After you've connected your Client Access and Mailbox servers to the VoIP gateways, IP PBXs,

SIP-enabled PBXs, or SBCs on your data network you must create the required UM components and also enable users for Unified Messaging so they can use the voice mail system.

Steps

Here are the basic steps for connecting VoIP gateways, IP PBXs, SIP-enabled PBXs, or SBCs to Client Access and Mailbox servers:

Step 1: Install the Client Access and Mailbox servers in your organization.

Step 2: Create and configure a Telephone Extension, SIP URI, or E.164 UM dial plan.

Step 3: Create and configure a UM IP gateway. You must create and configure a UM IP gateway for each VoIP gateway, IP PBX, SIP-enabled PBX, or SBC that will be accepting incoming calls and sending outgoing calls.

Step 4: Create a new UM hunt group if needed. If you create a UM IP gateway and don't specify a UM dial plan, a UM hunt group will be automatically created.

See the following sections for information about each step.

Step 1: Install your Client Access and Mailbox servers

When you're deploying Exchange servers in your organization, you can install the Client Access and Mailbox servers on the same computer or install the Client Access server on a separate computer from the Mailbox server. To install the Client Access server, run Setup.exe from the installation media. You use the same command when you're installing the Client Access server on a separate computer from the Mailbox server or installing it on the same computer. For details, see Install Exchange 2013 using the Setup wizard. If you want to add features and functionality to your existing Exchange server, you can use **Programs and Features** or Setup.exe.

Step 2: Create and configure a UM dial plan

After you've installed the required servers, you must first create a UM dial plan. A UM dial plan contains the configuration settings that enable you to connect to your telephony network by linking to a single or multiple UM IP gateways. A UM IP gateway and UM hunt group are directly linked to a UM dial plan and are also required. For details, see Create a UM dial plan.

A UM dial plan establishes a link from the telephone extension number of a user to their UM-enabled mailbox. When you create a UM dial plan, you can configure the number of digits in the extension numbers, the Uniform Resource Identifier (URI) type, and the VoIP security setting for the dial plan.

There are three types of dial plans: Telephone Extension, Session Initiation Protocol (SIP) URI, and E.164. When you create and configure a UM dial plan, you must determine the type of information

that's sent from your IP PBX, SIP-enabled PBX, or PBX and whether the calls are being sent to a Client Access or Mailbox server in a telephone extension or E.164 format. If you're deploying Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server, you must create and configure a SIP URI dial plan.

Step 3: Create and configure a UM IP gateway

After you create a UM dial plan, you must create a UM IP gateway to represent each VoIP gateway, IP PBX, or SBC on your network. When you create a UM IP gateway, you can configure it to use an IP address or a fully qualified domain name (FQDN). If you use an FQDN, you must make sure you've correctly configured a DNS host record for the IP gateway so the host name will be correctly resolved to an IP address.

After you install a VoIP gateway, IP PBX, or SBC, you must create a UM IP gateway to represent the physical hardware device. After you've created a UM IP gateway, the Client Access server that use the UM IP gateway will send a SIP OPTIONS request to the VoIP gateway, IP PBX, or SBC to ensure that it's responsive. If the VoIP gateway, IP PBX, SIP-enabled PBX, or SBC doesn't respond, the Client Access server will log an event with ID 1088 stating that the request failed. To resolve this issue, make sure that the VoIP gateway, IP PBX, or SBC is available and online and that the Unified Messaging configuration is correct.

Client Access and Mailbox servers will communicate only with a VoIP gateway, IP PBX, or SBC that's listed as a trusted SIP peer. In some cases, if two VoIP gateways, IP PBXs, or SBCs are configured to use the same IP address, an event with ID 1175 will be logged. This event may occur if you've configured your DNS zones with fully qualified domain names (FQDNs) for the VoIP gateways on your network. Unified Messaging protects against unauthorized requests by retrieving the internal URL of the Unified Messaging Web Services virtual directory that's located on the Mailbox server and then using the URL to build the list of FQDNs for the trusted SIP peers. After two FQDNs are resolved to the same IP address, this event will be logged.

You must restart the Microsoft Exchange Unified Messaging service if a UM IP gateway is configured to use an FQDN and the DNS record of the UM IP gateway is changed after the service has been started. If you don't restart the service, the Mailbox server won't be able to locate the UM IP gateway. This occurs because a Mailbox server maintains a cache for all UM IP gateways in memory and DNS resolution is performed only when the service is restarted or when the configuration of a VoIP gateway, IP PBX, or SBC has changed.

Exchange Unified Messaging supports various VoIP gateway vendors and other vendors of IP PBXs, SIP-enabled PBXs, and SBCs. Each of the supported VoIP gateways is designed to connect to a variety of third-party PBX systems.

For detailed information about VoIP gateways, see the following topics:

- Create a UM IP gateway
- Configuration notes for supported VoIP gateways, IP PBXs, and PBXs
- Connect a VoIP gateway, IP PBX, or session border controller to UM

For details, see Connect your voice mail system to your telephone network.

Step 4: Create a new UM hunt group (if needed)

Hunt group is a term used to describe a group of Private Branch eXchange (PBX) or IP PBX extension numbers that are shared by users. Hunt groups are used to efficiently distribute calls into or out of a specific business unit. Creating and defining a hunt group minimizes the chance that a caller who places an incoming call will receive a busy signal when the call is received.

UM hunt groups mirror the hunt groups that are used on PBXs and IP PBXs. When you configure your PBXs or IP PBXs, you must create and configure one or more UM hunt groups. UM hunt groups act as a link between the UM IP gateway and the UM dial plan.

Depending on how you create your UM IP gateway, you may have to create one or multiple new UM hunt groups. If you don't link a UM IP gateway with a dial plan when you create the UM IP gateway, a single UM hunt group is created by default. If you link a UM IP gateway to a UM dial plan when you create the UM IP gateway, all incoming calls will be sent through the UM IP gateway and those calls will be accepted by Client Access and Mailbox servers. If you don't link a UM IP gateway to a UM dial plan when you create the UM IP gateway, you'll need to create a UM hunt group with the correct pilot identifier for incoming calls to be forwarded from an UM IP gateway to a dial plan.

If you have multiple Outlook Voice Access and auto attendant numbers and have linked a UM IP gateway to a dial plan, you'll need to delete the UM hunt group that was created by default and create multiple UM hunt groups. For details about how to create a UM hunt group, see Create a UM hunt group.

Connect a VoIP gateway to communicate with a PBX

Unified Messaging > Telephone system integration with UM > Connect UM to your telephone system >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-30*

When you configure your telephony and data networks for Unified Messaging (UM) in Microsoft Exchange Server 2013, you must configure the VoIP gateways so they communicate with Client Access servers running the Microsoft Exchange Unified Messaging Call Router service and Mailbox servers running the Microsoft Exchange Unified Messaging service. You must also configure the VoIP gateways to communicate with the Private Branch eXchanges (PBXs) in your organization. You

can use the information and links in this topic to configure a VoIP gateway to communicate with a PBX.

Configuring a VoIP gateway

When you configure a VoIP gateway, you must consider whether the VoIP gateway device is analog, digital, or analog and digital. If the VoIP gateway interface that connects to a PBX is analog, you must correctly configure the appropriate settings to enable the VoIP gateway to communicate with a PBX. If the VoIP gateway interface that connects to a PBX is digital, it may require no additional configuration to enable the digital interface to communicate with a PBX.

The following suggested resources in the Exchange TechCenter provide information that can help you correctly configure your VoIP gateways and PBXs:

- **Supported VoIP gateway, IP PBX, and PBX documentation** Telephony advisor for Exchange 2013 includes configuration files and setup information that you can use when you configure VoIP gateways and PBXs.
- **Configuration and technical notes** Configuration notes for supported VoIP gateways, IP PBXs, and PBXs includes configuration files and setup information that you can use when you configure VoIP gateways and PBXs.
- **Configuring an AudioCodes-based VoIP gateway** The Microsoft Exchange Server Resource page provides the latest support and configuration information to help you configure AudioCodes-based VoIP gateways for use with Unified Messaging.
- **Configuring a Dialogic-based VoIP gateway** The Microsoft Exchange | Exchange Gateway | Dialogic Web site provides the latest support and configuration information for Dialogic-based VoIP gateways.

We recommend that all customers who plan to deploy Unified Messaging obtain the assistance of a Unified Messaging specialist. An Exchange Unified Messaging specialist will help make sure that there's a smooth upgrade to Unified Messaging from a legacy or third-party voice mail system and help you plan and deploy a new voice mail system with Exchange Unified Messaging. Deploying a new voice mail system or upgrading a legacy voice one requires significant knowledge about VoIP gateways, PBXs, and Unified Messaging. For more information about how to contact a Unified Messaging specialist, see Microsoft Exchange Server 2013 Unified Messaging (UM) Specialists or certified UM partners at Microsoft Pinpoint.

For more information

Configuration notes for supported VoIP gateways, IP PBXs, and PBXs

Connect UM to a supported VoIP gateway

Configuration notes for supported session border controllers

Applies to: *Exchange Online*

Topic Last Modified: *2014-01-20*

Session border controllers (SBCs) enable you to connect your on-premises telephony network to a Microsoft datacenter over a dedicated public WAN connection. An SBC sits on the edge of your on-premises IP network and connects to a second SBC in a Microsoft datacenter.

SBCs require the use of digital certificates to encrypt all traffic between your on-premises organization and the Microsoft datacenter. You must obtain a digital certificate for the network border element, such as a session border controller, that you're using to communicate with Exchange hybrid and online deployments. Digital certificates establish trust between your on-premises organization and the Microsoft datacenter and enable mutual Transport Layer Security (mutual TLS). After this trust is established, the network border elements at your on-premises organization and at the Microsoft datacenter exchange session keys, and use these keys to encrypt the subsequent data traffic.

In hybrid or online deployments, a UM IP gateway represents an SBC. The subject common name in the certificate must match the fully qualified domain name (FQDN) value in the Address box on the UM IP gateway that you create. For example, if you specify the FQDN address sbcexternal.contoso.com on your UM IP gateway, make sure that the subject name and subject alternative name in the certificate contain the same value: sbcexternal.contoso.com. The name that you use is case-sensitive, so make sure the case is the same on both the certificate and the UM IP gateway. If you're using an Acme Packet SBC and the common name doesn't match the UM IP gateway's FQDN, the call will be rejected with a 403 error.

Note:

Because SBCs are designed to sit on the network edge, they also function as a firewall. If you set up an SBC behind your organization's firewall, it can cause configuration problems.

Supported session border controllers

The following SBCs have been successfully tested for interoperability with Exchange hybrid and online deployments. Note that the capabilities and compatibilities of SBCs can vary, and the way you set them up can be different depending on other equipment on your network. Consult with the SBC manufacturer to see whether there are specific configuration notes for Unified Messaging in a hybrid or online deployment.

Vendor	Model	Configuration notes	Comments
Acme Packet	Net-Net 3820 or 4500	Acme Packet SBC in Microsoft Office 365 Unified Messaging	Dedicated SBC
AudioCodes	Mediant 1000B MSBG	MSBG Session Border Controller (SBC) with IP PBX	Dedicated SBC
AudioCodes	Mediant 1000B MSBG	MSBG Gateway and Session Border Controller (SBC) with Legacy PBX	SBC and IP gateway
Ingate	SIParator	Ingate SIParator Configuration Notes	Dedicated SBC
NET	VX1200 & VX1800	NET VX SESSION BORDER CONTROLLER CONFIGURATION NOTES FOR OFFICE 365 UNIFIED MESSAGING (UM)	SBC option for a VoIP gateway product
Sonus	SBC 1000/2000 2.2.1 or later	Configuring Sonus SBC 1000/2000 with Microsoft Office 365 Application Notes	Dedicated SBC

Connect your voice mail system to your telephone network

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-20*

After you've deployed all the required telephony equipment for your organization, including your VoIP gateways, IP PBXs, and SIP-enabled PBXs or Microsoft Lync Server, you need to create all the Unified Messaging (UM) components that will enable your telephony devices to communicate with servers in your organization.

UM components

The UM components enable the integration of Unified Messaging into your directory structure and your existing telephony infrastructure. Your directory stores all the components and settings for UM. Each UM component is necessary to support Unified Messaging. Some UM components are created to represent a telephony hardware device. Others are created to represent a telephony dial plan for an organization or to support a specific feature of Unified Messaging.

There's a tightly integrated and interconnected relationship between the UM components and the features available in Unified Messaging. To successfully plan and deploy Unified Messaging in your organization, you need to fully understand the relationship between each UM component and the others.

For more information about the UM components, see:

- UM dial plans
- UM IP gateways
- UM hunt groups
- Automatically answer and route incoming calls

For more information about setting up voice mail for users, see:

- UM mailbox policies
- Voice mail for users

UM dial plans

Exchange Server 2013 > Unified Messaging > Connect your voice mail system to your telephone network >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

Unified Messaging (UM) dial plans are the main component of Unified Messaging and are required to successfully deploy Unified Messaging voice mail on your network. The following sections discuss UM dial plans and how they're used in a UM deployment.

Overview of UM dial plans

A UM dial plan represents a set of Private Branch eXchanges (PBXs) or IP PBXs that share common user extension numbers. All users' extensions hosted on PBXs or IP PBXs within a dial plan contain the same number of digits. Users can dial one another's telephone extensions without appending a special number to the extension or dialing a full telephone number.

A UM dial plan mirrors a telephony dial plan. A telephony dial plan is configured on PBXs or IP PBXs.

In Unified Messaging, the following UM dial plan topologies can exist:

- A single dial plan that represents a subset of extensions or all extensions for an organization with one PBX or IP PBX.
- A single dial plan that represents a subset of extensions or all extensions for an organization with multiple networked PBXs or IP PBXs.
- Multiple dial plans that represent a subset of extensions or all extensions for an organization with one PBX or IP PBX.
- Multiple dial plans that represent a subset of extensions or all extensions for an organization with multiple PBXs or IP PBXs.

Users who belong to the same dial plan have these characteristics:

- An extension number that uniquely identifies the user mailbox in the dial plan.
- The ability to call or send voice messages to other members in the dial plan using only the extension number.

For more information about how to enable a user for Unified Messaging, see Enable a user for voice mail.

UM dial plans are used in Unified Messaging to make sure that user telephone extensions are unique. In some telephony networks, multiple PBXs or IP PBXs exist. In these telephony networks, there could be two users who have the same telephone extension number. UM dial plans resolve this situation. Putting the two users into two separate UM dial plans makes their extensions unique.

How dial plans work

When you integrate a telephony network with Unified Messaging, there must be one or more hardware devices called Voice over IP (VoIP) gateways or IP PBXs that connect your telephony network to your IP-based packet switched network. VoIP gateways convert circuit-switched protocols from a PBX found in a telephony network to a data-switched protocol such as IP. IP PBXs also convert circuit-switched protocols to a data-switched protocol. Session Border Controllers (SBCs) enable you to connect two IP based networks together over a public or private WAN and are found in UM hybrid or online deployments. Each VoIP gateway, IP PBX, or Session Border Controller (SBC) in your organization is represented by a UM IP gateway. For more information about UM IP gateways, see UM IP gateways.

Unified Messaging requires that you create at least one UM dial plan. Whether you create one or more dial plans, all the Exchange servers in your organization will answer incoming calls. There must also be a single or multiple UM IP gateways associated with the dial plan. In on-premises and hybrid deployments, after you install your Exchange servers and associate a UM IP gateway, all the Exchange servers will answer incoming calls for all dial plans. However, for on-premises or hybrid deployments, when you're integrating Exchange and Lync Server, you must create SIP URI dial plans.

Important:
Each time you create a UM dial plan, a default UM mailbox policy is also created. The UM mailbox policy is named <*Dial Plan Name*> Default Policy. This UM mailbox policy can be deleted or configured differently.

When you create the first UM IP gateway and specify a UM dial plan at the time you create it, a default UM hunt group is also created. Creating these components enables the Exchange servers to receive calls from a VoIP gateway, IP PBX, or SBC and then process those incoming calls for users who are associated with the UM dial plan. In on-premises or hybrid deployments, when a call comes in to the VoIP gateway, IP PBX, or SBC, it forwards the call to a Client Access server. The Client Access server then forwards the call to a Mailbox server and the Mailbox server tries to match the extension number of the user to the associated UM dial plan.

Types of dial plans

A Uniform Resource Identifier (URI) is a string of characters (numbers or alphabetic) that's used to identify or name a resource. In Unified Messaging, the main purpose of a URI is to enable VoIP devices to communicate with other devices using specific protocols. A URI defines the naming and numbering format or scheme used for the calling and called party information contained within a Session Initiation Protocol (SIP) header for an incoming or outgoing call.

The types of UM dial plans you create in Unified Messaging will depend on the URI types supported by the VoIP gateways or IP PBXs in your organization. The URI type is the type of string that's sent from the PBX or IP PBX. When you create a dial plan, you should know the specific URI types that are supported by your PBXs or IP PBXs. There are three formats or URI types that can be configured on UM dial plans:

- Telephone Extension (TeleExtn)
- SIP URI
- E.164

By default, each time you create a dial plan in Unified Messaging, the dial plan will be created to use the telephone extension URI type. After you create a dial plan, you won't be able to change the URI type. You must delete the existing dial plan and create another one with the correct URI type.

Telephone Extension URI type

The Telephone Extension URI type is the most common type of UM dial plan and is used with IP

PBXs and PBXs. When you configure a telephone extension (TelExtn) dial plan, the VoIP gateways, PBXs, and IP PBXs you use must support the TelExtn URI type. Today, most PBXs and IP PBXs support this URI type.

When a call is received by a PBX and the UM-enabled user isn't available to answer the call, the PBX will forward the call to a VoIP gateway. The VoIP gateway—or the IP PBX, if one is used—will translate the call from a circuit-based protocol to an IP based protocol. In the header for the SIP packet received from the VoIP gateway or IP PBX, the calling and called party information will be listed in one of the following formats:

- Tel:512345
- 512345@<*IP address*>

The telephone extension (TelExtn) format used is based on the configuration of the VoIP gateway or IP PBX.

SIP URI type

Session Initiation Protocol (SIP) is a standard protocol for initiating interactive user sessions that involve multimedia elements such as video, voice, chat, and gaming. SIP is a request-to-response based protocol that answers requests from clients and responses from servers. Clients are identified by SIP URIs. Requests can be sent through any transport protocol, such as UDP or TCP. SIP determines the endpoint to be used for the session by selecting the communication media and media parameters.

When you create a new dial plan, you have the option of creating a SIP URI dial plan if your environment has Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server deployed. You can also create a SIP URI dial plan if your organization has IP PBXs or SIP-enabled PBXs. In the latter case, your organization must also support SIP URIs and SIP routing.

A SIP URI is a user's SIP phone number. The SIP URI resembles an email address and is written in the following format: sip:<*user name*>@<*domain or IP address*>:*Port*. When a SIP-enabled IP PBX or PBX is used to send a call to the Exchange servers, the device will send the SIP URI for the calling and called party in the SIP header and will not include extension numbers.

E.164 URI type

E.164 is a standard numbering format that defines the international public telecommunication numbering plan used in the Public Switched Telephone Network (PSTN) and some data networks. E.164 defines the format of telephone numbers. E.164 numbers can have a maximum of 15 digits and are usually written with a plus sign (+) before the digits of the telephone number. To dial an E.164-formatted telephone number from a telephone, the appropriate international call prefix must be included in the number dialed. In an E.164 numbering plan for public telephone systems, each assigned number contains a country code (CC), a national destination code (NDC), and a subscriber number (SN).

When you create a new dial plan, you have the option to create an E.164 dial plan. However, if you create and configure an E.164 dial plan, the PBXs and IP PBXs must support E.164 routing. The SIP header received from a VoIP gateway associated with an E.164 dial plan will include the E.164-formatted telephone number and information about the calling and called party and will be listed in the following format: Tel:+14255550123. For Exchange Online deployments with Exchange Unified Messaging and Lync Server, you must use correctly formatted E.164 numbers for Outlook Voice Access and auto attendant numbers.

VoIP security

Exchange servers communicate with VoIP gateways, IP PBXs, and other Exchange computers in either Unsecured, SIP secured, or Secured mode, depending on how the UM dial plan is configured. In on-premises and hybrid deployments, Client Access and Mailbox servers can operate in any mode configured on a dial plan because the servers listen on TCP port 5060 for Unsecured requests and TCP port 5061 for Secured requests at the same time if they're configured to start in dual mode. Client Access and Mailbox servers answer all incoming calls for all UM dial plans, but these dial plans can have different VoIP security settings.

In on-premises and hybrid deployments, by default, when you create a UM dial plan, it will communicate in Unsecured mode, and the Client Access and Mailbox servers will send and receive data from VoIP gateways, IP PBXs, and SBCs without using encryption. In Unsecured mode, neither the Realtime Transport Protocol (RTP) media channel nor the SIP signaling information is encrypted. You can use the **Get-UMDialPlan** cmdlet in the Shell to determine the security setting for a specific UM dial plan.

In on-premises and hybrid deployments, you can configure a Client Access and Mailbox server to use mutual Transport Layer Security (mutual TLS) to encrypt the SIP and RTP traffic sent and received from other devices and servers. When you configure the dial plan to use SIP secured mode, only the SIP signaling traffic will be encrypted, and the RTP media channels will still use TCP, which isn't encrypted. However, when you configure the dial plan to use Secured mode, both the SIP signaling traffic and the RTP media channels are encrypted. An encrypted signaling media channel that uses Secure Realtime Transport Protocol (SRTP) also uses mutual TLS to encrypt the VoIP data.

You can configure the VoIP security mode either when you're creating a new dial plan or after you've created a dial plan using the EAC or the **Set-UMDialPlan** cmdlet in the Shell. When you configure the UM dial plan to use SIP secured or Secured mode, Client Access and Mailbox servers will encrypt the SIP signaling traffic or the RTP media channels or both. However, to be able to send encrypted data to and from Exchange servers, you must correctly configure the UM dial plan, and VoIP devices such as VoIP gateways, IP PBXs, and SBCs must support mutual TLS.

Outlook Voice Access

There are two types of callers who access the voice mail system using the Outlook Voice Access

number configured on a UM dial plan: unauthenticated callers and authenticated callers. When callers dial the Outlook Voice Access number configured on a dial plan, they're considered anonymous or unauthenticated until they input information including their voice mail extension and a PIN. The only option available to anonymous or unauthenticated callers is the directory search feature. After callers input their voice mail extension and their PIN, they'll be authenticated and given access to their mailbox. After they gain access to the voice mail system, they're using the Outlook Voice Access feature.

Outlook Voice Access is a series of voice prompts that give the caller access to email, voice mail, calendar, and other information. Outlook Voice Access lets authenticated callers navigate their personal information in their mailbox, place calls, or locate users using dual tone multi-frequency (DTMF), also known as touchtone, inputs or voice inputs.

Outlook Voice Access numbers

After you've created a UM dial plan, you need to add at least one Outlook Voice Access number. Outlook Voice Access numbers are also called dial plan pilot numbers. This number is used by Outlook Voice Access users to access their mailboxes and lets them search the directory.

By default, when you create a UM dial plan, no Outlook Voice Access number is configured. To enable users to use the Outlook Voice Access feature, you must configure at least one telephone or extension number. The number of alphanumeric characters in the Outlook Voice Access number can't exceed 20. After you configure this number on the dial plan, the number will be displayed in the voice mail options in Microsoft Outlook, and in Outlook Web App.

You can use the **Outlook Voice Access numbers** box on the UM dial plan to add a telephone number or extension that a user will call to access the voice mail system using Outlook Voice Access. In most cases, you'll enter an extension number or an external telephone number. However, because this field accepts alphanumeric characters, a SIP URI can be used if you're using an IP PBX, a SIP-enabled PBX, Office Communications Server 2007 R2 or Microsoft Lync Server.

Depending on the needs of your organization, you may want to configure one or more Outlook Voice Access number. You can have a single Outlook Voice Access number configured on a single UM dial plan or you can have multiple Outlook Voice Access numbers in a single UM dial plan, but you can't have a single Outlook Voice Access number that spans multiple UM dial plans.

Audio codecs

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-11-20*

In Unified Messaging (UM), an audio codec is used to store voice mail messages. Another codec is used between a VoIP gateway or IP Private Branch eXchange (PBX) and the Mailbox server running the Microsoft Exchange Unified Messaging service or the Client Access server running the Microsoft Exchange Unified Messaging Call Router service. Unified Messaging can use any of the following four audio codecs to create and store voice messages:

- MP3 (default)
- Windows Media Audio (WMA)
- Group System Mobile (GSM) 06.10
- G.711 Pulse Code Modulation (PCM) Linear

Warning:
The G.711 (PCMA and PCMU) and the G.723.1 codecs are VoIP codecs used between a VoIP gateway and the Client Access and Mailbox servers.

Part of planning your UM system involves selecting the correct audio codec based on the needs and requirements of your organization. This topic discusses the audio codecs that UM can use, and you can use it to help plan your UM deployment.

Codecs

The term *codec* is a combination of the words "coding" and "decoding" and is used with digital audio data. A codec is a software program that transforms digital data into an audio file format or audio streaming format. Codecs are used to convert an analog voice signal to a digital version of the voice signal. Codecs can vary in their sound quality, the bandwidth required to use them, and the system requirements needed to do the encoding.

Two types of codecs are used in Unified Messaging:

- The codec used between a VoIP gateway, IP PBX, or SIP-enabled PBX and the Client Access and Mailbox servers or between a PBX and a VoIP gateway.
- The codec used to encode and store voice messages for users.

When you use an ordinary telephone over the Public Switched Telephone Network (PSTN), your voice is transported in an analog format over the telephone line. But with Voice over IP (VoIP), your voice must be converted into digital signals. This conversion process is known as encoding. Encoding is performed by a codec. After the digitized voice has reached its destination, it must then be decoded back to its original analog format so the person on the other end of the call can hear and understand the caller.

VoIP codec

In UM, three types of codecs can be used between VoIP gateways or IP PBXs and the Client Access and Mailbox servers:

- G.711 µ-law
- G.711 A-law

- G.723.1

G.711 is a standard that was developed for use with audio codecs. There are two main algorithms defined in the standard for G.711: the μ-law algorithm that is used in North America and Japan and the A-law algorithm that is used in Europe and other countries. The G.723.1 audio codec is mostly used in VoIP applications and requires a license to be used. G.723.1 is a high-quality, high-compression type of codec.

A Client Access or Mailbox server and a supported VoIP gateway or IP PBX can offer both the G.711 and G.723.1 codecs. By default, the first codec to be used is G.723.1. If you want to use a codec other than G.723.1 between Client Access and Mailbox servers and the VoIP gateway or IP PBX, we recommend that you change the configuration on the VoIP gateway or IP PBX. The following table summarizes some common VoIP codecs.

VoIP codecs

VoIP codec	Bandwidth (Kbps)	Description
G.711	64	This codec requires very low processing. It needs a minimum of 128 kilobits per second (Kbps) for two-way communication.
G.723.1	5.3/6.3	This codec offers high compression with high-quality audio. It requires more processing than the G.711 codec. The G.723.1 codec uses reduced bandwidth but offers poorer-quality audio.

UM voice message storage codec

Unified Messaging dial plans are integral to the operation of UM. By default, when you create a UM dial plan, the UM dial plan uses the MP3 audio codec to create and store voice messages. However, after you create the UM dial plan, you can configure it to use WMA, GSM 06.10, or G.711 PCM Linear audio codecs.

Each audio codec has advantages and disadvantages. The MP3 audio codec was selected as the default audio codec because of its sound quality and compression properties. GSM 06.10 and G.711 PCM Linear audio codecs were included as available options because of their ability to support other types of messaging systems.

When you plan for UM, you must balance the size and the relative quality of the audio file that will be created for voice messages. Generally, the higher the bit rate for an audio file, the higher the quality. You must also consider whether the audio file is compressed. The following table shows the sample bit rate (bit/sec) and compression properties for each audio codec used in UM.

Default UM voice message storage codecs

Voice message storage codec	Bits	Compressed file?
MP3	16 bit	Yes
WMA	16 bit	Yes
G.711 PCM	16 bit	No
GSM 06.10	8-bit	Yes

In UM, the file type created for a voice message depends on the audio codec that's used to create the voice message audio file. The MP3 audio codec creates .mp3 audio files, the WMA audio codec creates .wma audio files, and the GSM 06.10 and G.711 PCM Linear audio codecs create .wav audio files. All these audio files are sent together with an email message to the recipient of the voice message.

Frequently, but not always, coding and decoding the digital data also involves compression or decompression. Audio compression is a form of data compression that reduces the size of audio data files. The audio compression algorithm used by the audio codec compresses the .wma or .wav audio files. In UM, the type of audio compression algorithm that is used is based on the type of audio codec selected in the UM dial plan properties. After the audio file is created and compressed, it's attached to the voice message.

Sometimes information from the digital data is lost during compression and decompression. The higher the compression that is used to compress the audio file, the greater the loss of information during the conversion. However, less disk space is used because the size of the audio file is reduced. Conversely, the lower the compression, the lower the loss of the information. However, more disk space must be used because of the increased size of each audio file.

RTAudio wideband or high fidelity audio for recording voice messages is also available as an audio codec. However, high fidelity audio using RTAudio is available only after you have successfully integrated Unified Messaging with Microsoft Lync Server . To enable RTAudio as the wire codec, either narrow or wideband, the UM dial plan must be configured as a Session Initiation Protocol (SIP) URI-type dial plan and you must set the call answering codec on the dial plan to MP3 or WMA to enable wideband audio (16Khz).

Important:
RTAudio is not available in environments where Lync Server is not deployed. This is because, in environments that haven't integrated Lync Server, the dial plan will be set to Telephone

Extension or E.164 and not to SIP URI.

There are two media streams for each incoming call: inbound to a Client Access server and outbound from a Mailbox server. When the dial plan type is set to SIP URI and the call-answering codec on the dial plan is set to MP3 or WMA, a Client Access server tries to select the RTAudio VoIP codec for the inbound media stream. If negotiation is successful, the RTAudio codec for the inbound stream will be used for call-answering calls or calls that originate from a Lync client or server.

Note:
Calls placed by using the Play on Phone feature will not use the RTAudio codec. The inbound stream for calls placed by using Play on Phone will use the G.711 or G.723.1 codec.

When the RTAudio codec is used, the voice message that is recorded will be recorded in high fidelity and will be stored as an audio file that has an .mp3 or .wma extension depending on how the dial plan is configured. When the voice message is played back to the user in Outlook or Outlook Web App they will hear the voice message in high fidelity audio. If negotiation is unsuccessful, either the G.711 or G.723.1 codec will be used. Both the G.711 and the G.723.1 codecs are narrowband codecs. When they're used as the VoIP codec, the voice message is recorded and stored as a narrowband audio file that has an .mp3 or .wma extension.

The outbound media stream will always be negotiated by using either the G.711 or G.723.1 codec. This means that callers will always hear narrowband audio over the telephone. This also applies to situations when a call is placed by using Microsoft Lync Server 2010 or later.

The audio format and codec that Mailbox servers use to store the audio in voice messages depends not only on the audio codec that's configured on the dial plan but also on the bit rate of the audio that UM negotiates with a SIP peer. If your environment includes Lync Server or SIP endpoints, a Mailbox server will also negotiate the audio codec to use with a SIP peer. For example, when wideband RTAudio is negotiated as the wire codec, a Mailbox server will then use either the 32 Kbps MP3 or WMA 9.2 format when creating voice messages, depending on the dial plan setting. The following table shows the relationship between the voice message storage audio codec and the VoIP or wire audio codec that's used.

Relationship between the storage audio codec and the VoIP or wire audio codec

Audio codec configured on a UM dial plan	VoIP or wire codec (narrowband) - G.723, G.711, or RTAudio (8KHz)	VoIP or wire codec (wideband) - RTAudio (16KHz)
G.711	G.711	Not applicable. A Client Access or Mailbox server doesn't negotiate wideband audio if the dial plan is set to G.711.
WMA	WMA 9 Voice	WMA 9.2

GSM	GSM 6.10	Not applicable. A Client Access or Mailbox doesn't negotiate wideband audio if the dial plan is set to GSM.
MP3	MP3 (16 Kbps)	MP3 (32 Kbps)

Return to top

UM message sizing

You can configure Unified Messaging to use one of the following four audio codecs for creating voice messages: MP3, WMA, GSM 06.10, and G.711 PCM Linear. By default, the MP3 format is selected.

The WMA audio codec is always stored in the Windows Media format, and the attachment is a file that has a .wma file name extension. Audio files encoded using the GSM or G.711 PCM Linear audio codecs are always stored in RIFF/WAV format, and the attachment is a file that has a .wav file name extension.

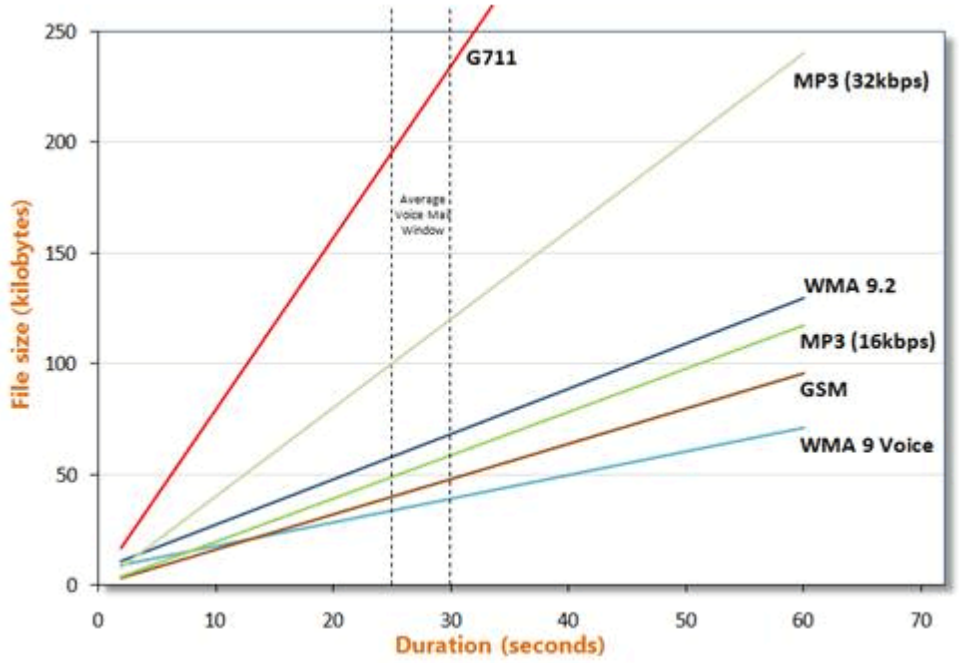
The size of UM voice messages depends on the size of the attachment that holds the voice data. In turn, the size of the attachment depends on the following factors:

- The duration of the voice mail recording
- The audio codec that's used
- The audio file storage format

The following figure shows how the size of the audio file depends on the duration of the voice mail recording for the three audio codecs that you can use in UM.

Note:
In this figure, the average length of a call-answered voice message is approximately 30 seconds.

Audio file size



MP3

By default, the MP3 format is selected and is the default audio file format for voice mail messages. The MP3 format is a common audio file format that's used to greatly reduce the size of the audio file and is most commonly used by personal audio devices or MP3 players. MP3 is a cross-platform type of audio codec and is used for compatibility with many mobile phones and devices and different computer operating systems.

WMA

WMA is the most highly compressed audio codec of the three kinds of codecs. The compression is approximately 11,000 bytes for each 10 seconds of audio. However, the .wma file format has a much larger header section than the .wav file format. The .wma file header section is approximately 7 kilobytes (KB), whereas the header section for the .wav file is less than 100 bytes. Although WMA audio recordings are recorded for longer than 15 seconds, they become smaller than GSM audio recordings. Therefore, for the smallest but highest-quality audio files, use the WMA audio codec.

Note:
If you using push notifications from your on-premises deployment for OWA for Devices, you can’t use the WMA format. OWA for Devices only supports the MP3 file format.

G.711 PCM Linear

The G.711 PCM Linear audio codec creates .wav audio files that are not compressed. Therefore, G.711 PCM Linear .wav audio files occupy the most space for any given duration when they're compared to the GSM and WMA audio codecs. G.711 PCM Linear .wav audio files occupy just over 160,000 bytes for each 10 seconds of audio. G.711 PCM Linear .wav audio files have the highest audio quality of the three audio codecs used by UM. However, the quality of comparable audio

files created using the WMA and GSM audio codecs is acceptable to most users who listen to voice messages.

GSM

The GSM audio codec creates .wav audio files that are compressed. GSM .wav audio files are just over 16,000 bytes for each 10 seconds of audio. However, GSM creates an audio file larger than the audio file created by the WMA audio codec. Therefore, when you are balancing the quality of the voice message and the size, this may not be the best choice.

Return to top

Secondary dial plans

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-06-11*

When you enable a user for Unified Messaging (UM), you're required to assign one extension number and a UM mailbox policy that will link the user to a UM dial plan. After the user is enabled for UM, you can assign additional extension numbers for that user within the same dial plan but the extension numbers within that dial plan must be unique. In some deployments, a user may need to be assigned the same extension number in two separate dial plans. If this is the case, you can link the user to a secondary UM dial plan. This can be useful, for example, if the user has two physical phones or travels between locations.

Contents

Overview

When you enable a user for Unified Messaging, you must define one extension number and one UM mailbox policy that links the user to a single UM dial plan. When you enable the user for UM and they're linked to a UM mailbox policy that's linked to a SIP URI or E.164 dial plan, you must also provide either a SIP address or E.164 number with the user's extension number. UM uses the dial plan and extension, along with the SIP address or E.164 number, to locate the user when a voice

mail message is submitted to the user's mailbox.

If you're using telephone extension dial plans and need to provide the same extension number for a user, you'll need to create a secondary dial plan, enable the user for UM and provide the same extension number for the user. This is because the extension number must be unique within a dial plan,

Note:

There's no limit to the number of secondary extension numbers that you can add for a UM-enabled user.

There may be times when a user travels between locations, has two or more phones, or wants to receive voice mail on one Direct Inward Dial (DID) extension number and receive faxes on a different DID extension number. To achieve this, you must add an additional DID extension to the user's mailbox and, in some cases, add a secondary dial plan.

In some configurations, after you add a second extension on a primary dial plan or add a single or multiple extension numbers to a secondary dial plan, the user can receive voice messages or faxes using one or more of the extension numbers. If you want UM to answer these fax calls and send them to the second DID extension number, you must configure the telephony equipment in your organization to forward the fax calls to the second DID extension number.

The mailbox of a UM-enabled user can be assigned the following:

- A single extension number, Session Initiation Protocol (SIP) address, or E.164 address on a single dial plan
- Multiple extension numbers on a single dial plan
- Multiple extension numbers on two separate dial plans

When a user is enabled for UM, you must specify an extension number and a UM mailbox policy. UM requires the extension number to identify the user when they sign in to Outlook Voice Access to retrieve messages. The UM mailbox policy contains a collection of configuration properties, with values that UM applies to any user who is UM-enabled under that policy. The UM mailbox policy is similar to the "class of service" found in other systems (for example, voice mail or PBX), in the sense that a change to a UM mailbox policy value can affect the behavior for a large number of associated users.

One property on a UM mailbox policy refers to a UM dial plan. This represents a set of telephony-capable extensions. This set has a numbering plan in which duplicate extension numbers aren't allowed.

Therefore, a user's extension number is unique within the UM dial plan in which they're UM-enabled. In fact, the UM dial plan and extension number pair must be unique within the organization. This is one way that UM uniquely identifies a UM-enabled user in an organization. Using a secondary dial plan makes it easier to keep the dial plan and extension number unique within an organization. For example, imagine an organization has two UM dial plans: Dial Plan A and Dial Plan B. A user's extension number in Dial Plan A is 55555 and, in Dial Plan B, it's 66666. When a secondary dial plan is used, the user's extension for Dial Plan A can be 55555 and their extension in Dial Plan B can also

be 55555. In both cases, the user's extension within the dial plan that's used is unique.

The following table defines terms that are used when discussing primary and secondary extensions, Outlook Voice Access numbers, and UM dial plans.

Term	Definition
primary extension	The extension number that's specified when the user is UM enabled.
primary dial plan	The UM dial plan that's specified when the user is UM enabled. The UM-enabled user is associated with the dial plan when the user is linked to the UM mailbox policy.
primary Outlook Voice Access number	The Outlook Voice Access number for the user's primary dial plan. The user's calls are forwarded to this number if there's no answer or their line is busy. It's also the number that the user calls when they want to sign in to Outlook Voice Access.
secondary extension	One or more extension numbers that may be added to a UM-enabled user's configuration.
secondary dial plan	A UM dial plan other than the primary dial plan in which one or more secondary extensions can be configured.
secondary Outlook Voice Access number	The Outlook Voice Access number of a user's secondary dial plan. A user can call this number from their secondary extension number when they want to sign in to Outlook Voice Access.

Return to top

Use of secondary extensions

In most deployments, only one extension is configured per UM-enabled user. However, there are some more advanced deployments that require you to add secondary extensions for your users.

When Microsoft Lync Server is used for Enterprise Voice, Unified Messaging can provide the voice mail system. However, the UM dial plan used for Enterprise Voice must be a SIP URI dial plan that's specific to UM configurations with Lync Server. In these deployments, the user's extension is provided by a Microsoft Unified Communications endpoint, such as Microsoft Office Communicator, running on the user's computer, or Office Communicator Phone Edition, running on a supported IP phone device. Thus, in most cases, the user's primary dial plan must be the same SIP URI dial plan used with Lync Server. But if the user requires more extension numbers, you shouldn't add another secondary extension to the primary dial plan. You must add a secondary dial plan and then add the secondary extension or EUM proxy address to the UM-enabled user.

For more information about adding, removing, or changing extensions, see one of the following:

- Change an extension number
- Add an extension number
- Remove an extension number

If you need to change SIP addresses or E.164 numbers for UM-enabled users, see:

- Add a SIP address
- Change a SIP address
- Remove a SIP address
- Add an E.164 number
- Change an E.164 number
- Remove an E.164 number

Call answering

Unified Messaging provides both of the following:

- **Call answering** Occurs when a user doesn't answer their phone and UM takes the call.
- **Outlook Voice Access** Used by users when they dial in to the voice mail system to access their mailbox.

Two configurations are used frequently:

- A UM-enabled user has two extension numbers (one primary, one secondary) in the primary dial plan. These extensions correspond to different phones on the user's desk and are connected to the same PBX. These different numbers are available to two separate audiences. In this configuration, the primary extension is the "general" work number and the secondary extension is the "task-specific" number, possibly a helpdesk line, or a dedicated fax number.
- A UM-enabled user spends a certain length of time, perhaps three weeks out of four, in their company's main office and the rest of the time in another office at one of the company's remote locations. The two offices have different PBXs, and the extension numbers are unique to each PBX. In this example, the user is configured to have a primary extension in their primary dial plan on the main office PBX and a secondary extension in a secondary dial plan on the PBX of the other office.

In either configuration, voice messages or missed call notification messages that are generated by

unanswered calls to either extension will be sent to the user's Inbox.

Outlook Voice Access

You may want UM-enabled users to be able to sign in to Outlook Voice Access from any extension, primary or secondary. While this is possible, there may be some architectural restrictions that keep this from working identically from all extensions. To sign in to Outlook Voice Access, UM-enabled users must perform the following steps:

1. Call an Outlook Voice Access number.
2. Key in their extension number if they're calling from another phone number.
3. Key in their PIN if they aren't enabled for Enterprise Voice and are calling from a Unified Communications phone, Office Communicator, or Lync Server.

Usage scenarios

- **Single extension with Outlook Voice Access** If the user has a single primary extension, they must always call the Outlook Voice Access number for their primary UM dial plan. If they call from their extension number, they won't be prompted to enter the extension number, and step 2 of the preceding steps will be skipped.
- **Two extensions in the primary dial plan with Outlook Voice Access** If the user has only two extensions, primary and secondary, and both the primary and secondary extension are in the same UM dial plan, they must always call the Outlook Voice Access number of the dial plan. If they call from either the primary or secondary extension, they won't be prompted to enter the extension number, and step 2 of the preceding steps will be skipped. Outlook Voice Access features will work the same way, whichever extension is used to sign in.
- **Extensions in the primary dial plan and in a secondary dial plan with Outlook Voice Access** If the user has only two extensions, primary and secondary, and the primary and secondary extensions are in different UM dial plans (primary and secondary); they should call the Outlook Voice Access number appropriate to their dial plan. From their primary extension, they should call the Outlook Voice Access number of the primary dial plan, and from their secondary extension, they should call the Outlook Voice Access number of the secondary dial plan. If they do this, they won't be prompted to enter the extension number, and step 2 of the preceding steps will be skipped.

Outlook Voice Access features that don't involve outbound dialing (for example "Call the sender" or "Call the office") will work the same way, whichever extension is used to sign in. However, Outlook Voice Access features that do require outbound dialing won't work as expected when the user signs in to the secondary dial plan unless the outbound dialing rules are exactly the same in both dial plans. For the behavior of outbound dialing to be exactly the same, you must ensure that the following properties are configured identically on the primary and secondary dial plans:

- Dialing codes (trunk access, national, and international)
- In-country or region dialing codes
- Dialing rules
- Dialing rule group names

A UM-enabled user is associated with a UM mailbox policy, and this UM mailbox policy is linked with the user's primary dial plan. The UM mailbox policy settings that are associated with the UM-enabled user's primary dial plan will be applied to the user. If a user is associated with a secondary dial plan with a second extension number in the secondary dial plan, the UM mailbox policy settings associated with the primary dial plan will still be applied. In Outlook Voice Access, the same UM mailbox policy settings associated with the primary dial plan are applied whether the user calls in to the primary dial plan or to a secondary dial plan.

The **AllowedInCountryOrRegionGroups** and **AllowedInternationalGroups** properties on the UM mailbox policy contain the names of groups of dialing rules that are configured on the **ConfiguredInCountryOrRegionGroups** and **ConfiguredInternationalGroups** properties of a UM dial plan. When a UM-enabled user calls in to Outlook Voice Access, the outbound calling rules from the UM mailbox policy associated with the primary or secondary dial plan will apply to calls the user makes, depending on whether the UM-enabled user has called in to the primary or secondary dial plan's Outlook Voice Access number.

For example, if a primary dial plan named “Contoso Dial Plan 1” has a dialing rule named “US and Canada” in its **ConfiguredInCountryOrRegionGroups** property, the UM mailbox policy “Contoso UM Policy 1” might also have “US and Canada” in its **AllowedInCountryOrRegionGroups** property. If you want to add a secondary extension in “Contoso Dial Plan 2” for a user in “Contoso UM Policy 2”, you would have to ensure that the **ConfiguredInCountryOrRegionGroups** property of “Contoso Dial Plan 2” also contains a rule named “US and Canada”. Otherwise, if the user signs in to Outlook Voice Access from their secondary extension, UM won't be able to find a rule on the secondary dial plan named “US and Canada”. If this happens, UM will only allow the user to call numbers allowed to any caller to the secondary dial plan, which could be more restrictive.

Return to top

UM features that operate differently for secondary dial plans

There's a set of UM features that can use secondary dial plans but may not work correctly in certain situations. It's important that you understand how each of these features is affected when you configure UM-enabled users to use a secondary dial plan.

Play on Phone

In Outlook Web App, Play on Phone uses the VoIP gateway that's associated with the user’s primary dial plan to make the outbound call. It applies dialing rules from the primary dial plan and the UM mailbox policy that's associated with the user's mailbox.

Directory search (Outlook Voice Access)

A search of the directory for a user who's been authenticated will follow these rules:

- The ability to search for a user and then leave a voice message or call a user will be available only if the user conducting the search is UM enabled and has a primary extension on the same dial plan as the user that's being called. If so, a search by name, alias, and primary extension will locate the user. However, searching by using the secondary extension won't locate the user.
- If the user being searched for is UM enabled and has a secondary extension on the called dial plan, then a search by name, alias, and secondary extension will find the user. However, although options to leave a voice message and call the contact will be offered, the call contact option won't succeed. In this case, a search by primary extension won't find the user.
- To find and be able to either call or leave a voice message for the user they're searching for, the UM-enabled user should use Outlook Voice Access through their primary dial plan's Outlook Voice Access number and search by name, alias, or primary extension. If the searched-for user is called using the secondary dial plan's Outlook Voice Access number, the user will only be found if the search is made by name, alias, or secondary extension. If the primary extension is used, the only option that will be available is for the user to leave a voice mail.

Directory search (Outlook Voice Access)

A search of the directory for a user who hasn't been authenticated will follow these rules:

- The user being searched for will be found and the option to leave a voice message or call the user will be offered only if the user is UM enabled and has a primary extension on the called dial plan. If so, a search by name, alias, and primary extension will find the user. However, a search by secondary extension won't find the user.
- If the user being searched for is UM enabled, has a secondary extension on the called dial plan, and the option **Transfer and search** > **Allow callers to** > **Leave voice messages without ringing a user's phone** is selected on the called dial plan, then a search by name, alias, and secondary extension will find them. However, the option to leave voice mail will be offered to the caller, and there will be no option to call them.
- To find and be able to either call or leave a voice message for a user, the caller must call the Outlook Voice Access number of the user's primary dial plan and search by name, alias, or the user's secondary extension. If the user's secondary Outlook Voice Access number is called, they will only be found if the **Allow callers to search by name of alias** option is set to **In the entire organization**. In this case, only the option to leave a voice message will be provided.

Call the Sender (Outlook Voice Access)

When a user calls in to Outlook Voice Access and chooses the option to Call the Sender, they can send either an email message or a voice mail message to a UM-enabled user. The options available depend on whether the caller is associated with the same dial plan as the sender they're calling. Calls to a UM-enabled user when the caller dials in to an Outlook Voice Access number and the caller is authenticated will follow these rules:

- **Email messages** If the sender of the email message is a UM-enabled user, choosing the option

to call the sender will result in a call to the sender's primary extension that's configured on the user's primary dial plan. In the case where the sender's primary extension is on a dial plan that's different from the caller's, the prompt to "Call the Sender" will only be provided if there's a business, home, or mobile phone configured for the sender and the dialing rules are configured to allow the call.

- **Voice mail messages** If the caller is a UM-enabled user, the option to call the sender will always result in a call to the extension that the sender uses to leave their voice message. If this extension has a number of digits different from the called dial plan, the prompt to call the sender won't be provided unless there are dialing rules in place that would permit the call. For example:
  - The "Call the sender" option will be offered if the sender uses an extension on the dial plan that was used to send the voice message.
  - The "Call the sender" option will be played if the sender uses an extension from a different dial plan than the dial plan that's used with Outlook Voice Access to send the voice message and both dial plans have the same number of digits. The success of the call will depend on whether the VoIP gateway and PBX infrastructure permit the call transfer.
  - The "Call the sender" option won't be played if the sender uses an extension from a different dial plan than the dial plan that's used with Outlook Voice Access to send the voice message, the dial plans have a different number of digits, and there are no outdialing rules that match the sender's extension.

Return to top

UM dial plan procedures

Unified Messaging > Connect your voice mail system to your telephone network > UM dial plans >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-16*

Create a UM dial plan

Manage a UM dial plan

Add Mailbox and Client Access servers to a SIP URI dial plan

Remove Mailbox and Client Access servers from a SIP URI dial plan

Change the audio codec

Configure the maximum call duration

Configure the maximum recording duration

Configure the recording idle time-out value

Configure the VoIP security setting

Configure a dial plan for users who have similar names

Delete a UM dial plan

Create a UM dial plan

Connect your voice mail system to your telephone network > UM dial plans > UM dial plan procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-16*

A Unified Messaging (UM) dial plan contains configuration information related to your telephony network. A UM dial plan establishes a link from the telephone extension number of a user enabled for voice mail to their mailbox. When you create a UM dial plan, you can configure the number of digits in the extension numbers, the Uniform Resource Identifier (URI) type, and the Voice over IP (VoIP) security setting for the dial plan.

Each time you create a UM dial plan, a UM mailbox policy is also created. The UM mailbox policy is named <*DialPlanName*> Default Policy.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to create a UM dial plan

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**, and then click **New +**.
2. On the **New UM dial plan** page, complete the following boxes:
   - **Name** Type the name of the dial plan. A UM dial plan name is required and must be unique.

However, it's used only for display in the EAC and the Shell. If you have to change the display name of the dial plan after it's been created, you must first delete the existing UM dial plan and then create another dial plan that has the appropriate name. If your organization uses multiple UM dial plans, we recommend that you use meaningful names for your UM dial plans. The maximum length of a UM dial plan name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

Although you can include spaces in a UM dial plan name, if you integrate Unified Messaging with Office Communications Server 2007 R2 or Microsoft Lync Server, the dial plan name can't include spaces. Therefore, if you created a dial plan with spaces in the display name, and you're integrating with Office Communications Server 2007 R2 or Lync Server, you must first delete that dial plan and then create another dial plan that doesn't include spaces in the display name.

Important:

Although the box for the name of the dial plan can accept 64 characters, the name of the dial plan can't be longer than 49 characters. If you try to create a dial plan name that contains more than 49 characters, you'll receive an error message. The message will say that the UM mailbox policy couldn't be generated because the UM dial plan name is too long. This happens because, as mentioned earlier, when you create a dial plan a default UM mailbox policy named <*DialPlanName*> Default Policy is also created. When the 15 characters in Default Policy are added to the name of the dial plan, the total characters exceed the limit. The *name* parameter for both the UM dial plan and UM mailbox policy can be 64 characters. However, if the name of the dial plan is longer than 49 characters, the name of the default UM mailbox policy will be longer than 64 characters, and this isn't allowed by the system.

- **Extension length (digits)** Enter the number of digits for the dial plan. The number of digits for extension numbers is based on the telephony dial plan created on a Private Branch eXchange (PBX) or IP PBX. For example, if a user associated with a telephony dial plan dials a four-digit extension to call another user in the same telephony dial plan, you select 4 as the number of digits in the extension.

This is a required box that has a value range from 1 through 20. The typical extension length is from 3 through 7. If your existing telephony environment includes extension numbers, you must specify a number of digits that matches the number of digits in those extensions.

When you create a Session Initiation Protocol (SIP) or an E.164 dial plan and associate a UM-enabled user with the dial plan, you must still input an extension number to be used by the user. This number is used by Outlook Voice Access users when they access their mailbox.

- **Dial plan type** A Uniform Resource Identifier (URI) is a string of characters that identifies or names a resource. The main purpose of this identification is to enable VoIP devices to communicate with other devices over a network using specific protocols. URIs are defined in schemes that define a specific syntax and format and the protocols for the call. In simple terms, this format is passed from the IP PBX or PBX. After you create a UM dial plan, you won't be able to change the URI type without deleting the dial plan, and then re-creating the dial plan to include the correct URI type. You can select one of the following URI types for the dial plan:
  - **Telephone extension** This is the most common URI type. The calling and called party information from the VoIP gateway or IP Private Branch eXchange (PBX) is listed in one of

the following formats: Tel:512345 or 512345@<*IP address*>. This is the default URI type for dial plans.

- **SIP URI** Use this URI type if you must have a Session Initiation Protocol (SIP) URI dial plan such as an IP PBX that supports SIP routing or if you're integrating Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server and Unified Messaging. The calling and called party information from the VoIP gateway. IP PBX, or Communications Server 2007 R2 or Lync Server is listed as a SIP address in the following format: sip:<*username*>@<*domain* or *IP address*>:Port.
- **E.164** E.164 is an international numbering plan for public telephone systems in which each assigned number contains a country code, a national destination code, and a subscriber number. The calling and called party information sent from the VoIP gateway or IP PBX is listed in the following format: Tel:+14255550123.

Warning:

After you create a dial plan, you will be unable to change the URI type without deleting the dial plan, and then re-creating the dial plan to include the correct URI type.

- **VoIP security mode** Use this drop-down list to select the VoIP security setting for the UM dial plan. You can select one of the following security settings for the dial plan:
  - **Unsecured** By default, when you create a UM dial plan, it is set to not encrypt the SIP signaling or RTP traffic. In unsecured mode, the Client Access and Mailbox servers associated the UM dial plan send and receive data from VoIP gateways, IP PBXs, SBCs and other Client Access and Mailbox servers using no encryption. In unsecured mode, neither the Realtime Transport Protocol (RTP) media channel nor the SIP signaling information is encrypted.
  - **SIP secured** When you select **SIP secured**, only the SIP signaling traffic is encrypted, and the RTP media channels still use TCP, which isn't encrypted. With SIP secured, Mutual Transport Layer Security (TLS) is used to encrypt the SIP signaling traffic and VoIP data.
  - **Secured** When you select **Secured**, both the SIP signaling traffic and the RTP media channels are encrypted. Both the secure signaling media channel that uses Secure Realtime Transport Protocol (SRTP) and the SIP signaling traffic use mutual TLS to encrypt the VoIP data.
- **Audio language** Use this list to specify the default language to be used by Outlook Voice Access users. This setting doesn’t apply to the language setting on a UM auto attendant. You can set the language for Outlook Voice Access to be the same as or different from the language that’s used on a UM auto attendant. When a user places a call to a user who is linked with a dial plan, the audio language is the default language that the voice-recorded operator uses. The system prompts that callers hear are played in the same language. The language that is chosen on the UM dial plan is used to read email, voice mail, and calendar items; to say the user’s name if a personal greeting hasn’t been recorded; to transcribe a voice message using the Voice Mail Preview feature; and to enable Automatic Speech Recognition (ASR) to work correctly.
- **Country/Region code** Use this box to type the country/region code number to be used for outgoing calls. This number will precede the telephone number that’s dialed. This box accepts

from 1 through 4 digits. For example, in the United States, the country/region code is 1. In the United Kingdom, it's 44.

3. Click **Save**.

Use the Shell to create a UM dial plan

This example creates a new UM dial plan named `MyUMDialPlan` that uses four-digit extension numbers.

```
New-UMDialplan -Name MyUMDialPlan -NumberofDigits 4
```

This example creates a new UM dial plan named `MyUMDialPlan` that uses five-digit extension numbers and supports SIP URIs.

```
New-UMDialplan -Name MyUMDialPlan -UriType SIPName -
NumberofDigits 5
```

Manage a UM dial plan

Connect your voice mail system to your telephone network > UM dial plans > UM dial plan procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-16*

After you create a Unified Messaging (UM) dial plan, you can view and configure a variety of settings. For example, you can configure the level of Voice over IP (VoIP) security, the audio codec, and dialing restrictions. The settings that you configure on the UM dial plan affect all users who are linked with the dial plan through a UM mailbox policy.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to view or configure UM dial plan settings

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to view or modify, and then click **Edit** .
3. On the **UM Dial Plan** page, click **Configure**. Use the configuration options to view specific dial plan settings and to enable or disable features as described in the following steps.
4. **General** Use this page to view specific dial plan settings or to enable or disable features for UM-enabled users:
   - **Name** This is the name of the dial plan that was created. The maximum length of a UM dial plan name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
   - Although you can include spaces in a UM dial plan name, if you integrate Unified Messaging with Office Communications Server 2007 R2 or Microsoft Lync Server, the dial plan name can’t include spaces. Therefore, if you created a dial plan with spaces in the display name, and you’re integrating with Office Communications Server 2007 R2 or Lync Server, you must first delete that dial plan and then create another dial plan that doesn't include spaces in the display name.

Important:

Although the box for the name of the dial plan can accept 64 characters, the name of the dial plan can't be longer than 49 characters. If you try to create a dial plan name that contains more than 49 characters, you’ll receive an error message. The message will say that the UM mailbox policy couldn't be generated because the UM dial plan name is too long. This happens because, as mentioned earlier, when you create a dial plan a default UM mailbox policy named <*DialPlanName*> Default Policy is also created. When the 15 characters in Default Policy are added to the name of the dial plan, the total characters exceed the limit. The *name* parameter for both the UM dial plan and UM mailbox policy can be 64 characters. However, if the name of the dial plan is longer than 49 characters, the name of the default UM mailbox policy will be longer than 64 characters, and this isn't allowed.

   - **Extension length (digits)** This is the number of digits in the extension numbers for users who are associated with this dial plan. For example, if a user associated with a dial plan dials a 4-digit extension to call another user in the same dial plan, select 4 as the number of digits in the extension.

The number of digits for extension numbers is based on the telephony dial plan created on an IP PBX or PBX. This is a required field that has a value range from 1 through 20. The typical extension length is from 3 through 7 digits. If your existing telephony environment includes extension numbers, you must specify a number of digits that matches the number of digits in those extensions

when you create the UM dial plan.

- **Dial plan type** A Uniform Resource Identifier (URI) is a string of characters that identifies or names a resource. The main purpose of this identification is to enable VoIP devices and PBXs to communicate with other devices over a network using specific protocols. URIs are defined in schemes that define a specific syntax and format and the protocols for the call. In simple terms, this format is passed from the IP PBX or PBX and the type of dial plan you create must match that format. After you create a UM dial plan, you won't be able to change the dial plan type without deleting the dial plan, and then re-creating the correct type of dial plan. You can select one of the following dial plan types:
  - **Telephone extension** This is the most common dial plan type. The calling and called party information from the VoIP gateway or IP Private Branch eXchange (PBX) is listed in one of the following formats: Tel:512345 or 512345@<*IP address*>. This is the default type for dial plans.
  - **SIP URI** Use this dial plan type if you must have a Session Initiation Protocol (SIP) URI dial plan such as an IP PBX that supports SIP routing, a SIP-enabled PBX, or if you're integrating Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server and Unified Messaging. The calling and called party information from the VoIP gateway. IP PBX, SIP-enabled PBX, or Communications Server 2007 R2 or Lync Server is listed as a SIP address in the following format: sip:<*username*>@<*domain* or *IP address*>:Port.
  - **E.164** E.164 is an international numbering plan for public telephone systems in which each assigned number contains a country code, a national destination code, and a subscriber number. The calling and called party information sent from the VoIP gateway and PBX or IP PBX is listed in the following format: Tel:+14255550123.

Note:

After you create a dial plan, you won't be able to change the dial plan type without deleting the dial plan, and then re-creating the correct type of dial plan.

- **VoIP security mode** Use this drop-down list to select the VoIP security setting for the UM dial plan. You can select one of the following security settings for the dial plan:
  - **Unsecured** By default, when you create a UM dial plan, it's set to not encrypt the SIP signaling or RTP traffic. In Unsecured mode, the Exchange servers associated with the UM dial plan send and receive data from VoIP gateways, IP PBXs, SBCs, and other Exchange servers using no encryption. In Unsecured mode, neither the Realtime Transport Protocol (RTP) media channel nor the SIP signaling information is encrypted.
  - **SIP secured** When you select **SIP secured**, only the SIP signaling traffic is encrypted, and the RTP media channels still use TCP, which isn't encrypted. With SIP secured, mutual Transport Layer Security (TLS) is used to encrypt the SIP signaling traffic and VoIP data.
  - **Secured** When you select **Secured**, both the SIP signaling traffic and the RTP media channels are encrypted. Both the secure signaling media channel that uses Secure Realtime Transport Protocol (SRTP) and the SIP signaling traffic use mutual TLS to encrypt the VoIP data.

5. **Dial codes** Use this page to configure the dial codes for a UM dial plan. Several dial code settings can be configured on the dial plan. These include incoming and outgoing calling

options. You can configure the following:

- **Dial codes for outgoing calls** Use these settings to specify the dialing codes for outgoing calls that can be made by UM-enabled users. These outgoing calls are calls that are placed using Outlook Voice Access or from a voice mail message.
  - **Outside line access code** Use this field to type the number or numbers used to access an outside telephone number for outgoing external calls. This number will precede the telephone number dialed. This is also called a trunk access code. This field accepts from 1 through 16 digits. For many organizations, this number is 9. By default, this field isn't populated.

Frequently, this setting is used in telephony environments where a PBX or IP PBX is located onsite or maintained in an organization. It may not have to be configured if your organization's telephony environment is maintained by an external business or vendor.

  - **International access code** Use this field to type the number code used to access international telephone numbers for outgoing calls. This number will precede the telephone number dialed. By default, this field isn't populated. This field accepts from 1 through 4 digits. For example, the international access code for the United States is 011. For Europe, it's 00.
  - **National number prefix** Use this field to type the number code used to dial telephone numbers that are out of an area code but within the country/region. This number will precede the telephone number dialed. By default, this field isn't populated. This field accepts from 1 through 4 digits. For example, 0 is used in Europe, and 1 is used in North America.
  - **Country/Region code** Use this field to type the country/region code number used for outgoing calls. This number will precede the telephone number dialed. By default, this field isn't populated. This field accepts from 1 through 4 digits. For example, in the United States, the country/region code is 1. In the United Kingdom, it's 44.

- **Number formats for dialing between UM dial plans** Use these settings to configure calls between users in separate dial plans when they place calls between the dial plans.
  - **Country/Region number format** Use this field to specify how a user's telephone number should be dialed by the Exchange servers when users are in a different dial plan that has the same country code. This is used by auto attendants and when an Outlook Voice Access user searches and tries to call the user in the directory.

This entry consists of a number prefix and a variable number of characters (for example, 020*xxxxxxx*).To determine the telephone number, Unified Messaging will append the last *x* digits from the telephone number specified in the directory to the prefix specified.

  - **International number format** Use this field to specify how a user's telephone number should be dialed by Unified Messaging when the users are in different dial plans that have different country codes. This is used by an auto attendant and when an Outlook Voice Access user searches and tries to call the user in the directory.

This entry consists of a number prefix and a variable number of characters (for example, 4420*xxxxxxx*). To determine the telephone number, Unified Messaging will append the last *x* digits from the telephone number specified in the directory to the prefix specified.

- **Number formats for incoming calls within the same dial plan** Use this field to add or remove a number format for incoming calls that are placed between users in the same dial plan. This field accepts both numbers and the letter “x” as a wild card character. No other letters can be used in this field.

For incoming calls within the same dial plan add a number format. For example, to add a number format for 5-digit extensions, enter, 142570xxxxx and click **+**. To remove a number format, click **–**.

6. **Outlook Voice Access** Use this page to configure Outlook Voice Access settings for the UM dial plan. Outlook Voice Access enables users to access their individual mailboxes to retrieve email, voice messages, contacts, and calendaring information using a telephone. You can view or configure the following:
   - **Welcome greeting** This display-only field shows the name of the sound file that will be used for the welcome greeting.
     - **Default greeting** The welcome greeting is used when an Outlook Voice Access user or another caller calls the Outlook Voice Access number and does a directory search. This audio file is the default greeting for a UM dial plan. However, you may want to change this welcome greeting and provide another welcome greeting specific to your company, such as, "Welcome to Outlook Voice Access for Contoso, Ltd."

If you decide to customize this greeting, you must first record the customized greeting, save it as a .wav file, and then configure the dial plan to use this customized greeting. The file name and path must not exceed 255 characters.

You can add a customized greeting by clicking **Change**, and then clicking **Browse** to select a previously recorded custom greeting and specify the audio file (.wav) to use for the welcome greeting. If you don't specify an audio file, Outlook Voice Access users will hear a default welcome greeting that says, "Welcome, you are connected to Microsoft Exchange."

   - **Informational announcement** When enabled, this optional recording plays immediately after the business or non-business hours welcome greeting. An informational announcement may state the organization's security polices for accessing the system, for example, "When you gain access to our system using Outlook Voice Access, you have agreed to the terms of our business agreement and all security policies for our organization apply. Access to our system is monitored and gaining illegal access will be prosecuted.” An informational announcement can also provide information that's required for compliance with company policy, for example, "Calls may be monitored for training purposes." If it's important that callers hear the whole informational announcement, it can be marked as uninterruptible.

By default, there's no informational announcement configured on UM dial plans. To enable an informational announcement and use a custom audio file specific to your organization, click **Change** and then click **Browse**.

   - **Allow announcement to be interrupted** Select this check box to enable the Outlook Voice Access user to interrupt the informational announcement. You should do this if you have long informational announcements. Outlook Voice Access users may become frustrated if the informational announcement is long and they can't interrupt it to access the options provided by the UM dial plan.
   - **Outlook Voice Access numbers** Use this field to add a telephone or extension number or a

SIP URI that an Outlook Voice Access user will call to access the voice mail system using Outlook Voice Access. In most cases, you enter an extension number or an external telephone number. However, because this field accepts all alphanumeric characters, a SIP URI can be used if you're using an IP PBX, Office Communications Server 2007 R2, or Microsoft Lync Server.

By default, when a dial plan is created, no Outlook Voice Access numbers are defined. To enable Outlook Voice Access users to call into Outlook Voice Access, you must configure at least one telephone number. The number of alphanumeric characters can't exceed 20.

When you configure this number on the dial plan, this number will be displayed in Microsoft Office Outlook 2007 or later versions and Outlook Web App for voice mail options.

To add a new Outlook Voice Access number, enter the number in the box and click **+**. To remove an Outlook Voice Access number, click **–**.

7. **Settings** Use this page to configure dial plan settings for Unified Messaging. When you configure settings on this page, you can control how Outlook Voice Access users and external callers calling into an auto attendant linked to the dial plan locate users in your organization, the audio codec that is used for voice mail messages, the number of sign-in failures, and time-out values. You can configure the following:
   - **Primary way to search for names** Use this list to select the primary way that callers can locate a user when they dial in to the system.

By default, **Last First** is selected. This means that when users are searching for a user in the directory, they will enter the user's last name first and then the first name.

When an Outlook Voice Access user calls in to an Outlook Voice Access number to access their mailbox, a caller calls in to an Outlook Voice Access number to perform a directory search, or a caller calls in to an auto attendant linked to a UM dial plan, they can search for a user in the directory by spelling their name or alias.

You must select one of the supported methods to be able to use the dial-by-name primary method. The following methods are supported:

- **Last First (default)**
- **First Last**
- **SMTP address**

   - **Secondary way to search for names** Use this list to select the secondary way that callers can locate a user when they dial in to the system.

By default, **SMTP address** is selected. This means that when users search for a user in the directory, they will enter the user's email alias or SMTP address.

When an Outlook Voice Access user calls in to an Outlook Voice Access number to access their mailbox, a caller calls in to an Outlook Voice Access number to perform a directory search, or a caller calls in to an auto attendant linked to a UM dial plan, they can search for a user in the directory by spelling their name or alias. When you select one of these options, callers can use the primary way to search for names or the secondary way to search for names to locate users in the directory.

You aren't required to select one of the four methods that are supported. However, if you don't select a secondary way to search for users, callers will be given only one way to search for a user.

The following options are available:

- **Last First**
- **First Last**
- **SMTP address** (default)
- **None**

- **Audio codec** Use this list to select the audio codec that will be used by the dial plan. When a caller places a call to a user who is associated with the dial plan and leaves a voice message, Unified Messaging uses the audio codec that you select from this list to record voice messages that will be sent to voice mail-enabled users. The following audio codecs are supported:
  - **MP3** (default)
  - **WMA** (Windows Media Audio)
  - **G711** (Pulse Code Modulation (PCM) Linear)
  - **GSM** (Group System Mobile 06.10)

By default, the MP3 format is selected. The MP3 format is a common audio file format that's used to greatly reduce the size of the audio file and is most commonly used by personal audio devices or MP3 players. MP3 is a cross-platform type of audio codec and is used for compatibility with many mobile phone and devices and various computer operating systems.

WMA is used because it's highly compressed and has high-quality format properties. G.711 PCM Linear is a telephone-quality audio codec format that's the least compressed and has the lowest-quality format. GSM 06.10 is an audio codec format that's used by mobile phone vendors and is the standard for digital mobile phone services.

If you're concerned about users' disk quotas, select WMA as the audio codec. Voice files saved in .wma format are approximately half the size of the same voice recording made using one of the other audio codecs.

- **Operator extension** Use this text box to enter the telephone number or an extension number for the dial plan's operator. This is different than an operator extension that is configured on a UM auto attendant. However, you can put in the same phone or extension number for both types of operators.

You can configure this setting to transfer calls to an auto attendant if one is configured, to a human operator, to external telephone numbers, or to extension numbers.

When a caller who is using the telephone keypad presses 0, or says "reception" or "operator," or the **Number of input failures before disconnecting** threshold is exceeded, the caller is transferred to the telephone or extension number that you specify in this text box.

This telephone number can be a number external to the organization or an internal telephone extension number. For example, if the extension number for the receptionist or operator is 81964 and your organization has only one dial plan, enter 81964.

By default, this setting is blank. If you don't enter a number in this text box, the ability to transfer calls to the operator is disabled and callers are politely disconnected because there's no one to answer the call.

We recommend that you populate this text box with a telephone number that transfers callers to an operator if they can't locate a specific user in the directory.

- **Number of sign-in failures before disconnecting** Use this text box to enter the number of sequential unsuccessful logon attempts allowed before a caller is disconnected.

The value of this setting can be from 1 through 20. Setting this value too low can frustrate users. For most organizations, this value should be set to the default of three attempts.

- **Timeouts and retries** These settings apply to Outlook Voice Access users and external callers that dial into a UM auto attendant.
  - **Maximum call duration (minutes)** Use this text box to enter the maximum number of minutes that an incoming call can be connected to the system without being transferred to a valid extension number before the call is ended. For most organizations, this value should be set to the default of 30 minutes.

This setting applies to all kinds of calls. This includes incoming Outlook Voice Access calls, voice calls internal to your organization, and voice and incoming fax calls external to your organization.

The value of this setting can be from 10 through 120. Setting this value too low can cause incoming calls to be disconnected before they are completed. For example, if your organization receives many large fax messages, you may want to consider increasing this value from the default so that all the pages for fax messages are received.

  - **Maximum recording duration (minutes)** Use this text box to enter the maximum number of minutes allowed for each voice recording when a caller leaves a voice mail message. For most organizations, this value should be set to the default of 20 minutes.

The value of this setting can be from 1 through 100. Setting this value too low can cause long voice messages to be disconnected before they are completed. Setting this value too high lets users save lengthy voice messages in their Inboxes.

This setting is important if you have implemented strict disk quotas for users. This value must be less than the value set for the **Maximum call duration (minutes)** setting.

  - **Recording idle time out (seconds)** Use this text box to enter the number of seconds of silence that the system allows when a voice message is being recorded before the call is ended. For most organizations, this value should be set to the default of 5 seconds.

The value of this setting can be from 2 through 10. Setting this value too low can cause the system to disconnect callers before they are finished leaving their voice messages. Setting this value too high allows lengthy silences in voice messages.

  - **Number of input failures before disconnecting** Use this text box to configure the number of times that callers can enter incorrect menu choices before they are disconnected. For most organizations, this value should be set to the default of three attempts. This is an important setting for speech-enabled UM dial plans.

Examples of incorrect data include when a caller requests an extension number that isn't found in the system, the system can't locate the user's extension number to transfer the call, or the caller presses a menu option that isn't valid.

The value of this setting can be from 1 through 20. Setting this value too low may prematurely disconnect the caller.

- **Audio language** Use this list to specify the default language to be used by Outlook Voice Access users. This setting doesn’t apply to the language setting on a UM auto attendant. You

can set the language for Outlook Voice Access to be the same as or different from the language that's used on a UM auto attendant. When a user places a call to a user who is linked with a dial plan, the audio language is the default language that the voice-recorded operator uses. The system prompts that callers hear are played in the same language. The language that is chosen on the UM dial plan is used to read email, voice mail, and calendar items; to say the user's name if a personal greeting hasn't been recorded; to transcribe a voice message using the Voice Mail Preview feature; and to enable Automatic Speech Recognition (ASR) to work correctly.

For on-premises deployments, adding other languages lets Outlook Voice Access use a language other than U.S. English. For example, if an Outlook Voice Access user calls in using an Outlook Voice Access number from a desk telephone, the user is greeted with a prerecorded operator's voice in English. Even if the same user selects a different language, such as French, in Outlook Web App, the menus are still read in U.S. English. For the user to be able to hear the prerecorded operator menus in French, you must install the appropriate language pack.

Note:
For Exchange Online, all languages are available.

8. **Dialing rules** Use this page to specify dialing rules for in-country/region and international calls placed by UM-enabled users. Each entry defined on the dialing rule determines the types of calls that users within a specific dialing rule group can make. After you use the **Dialing rules** page to configure dialing rules, you must configure the UM dial plan, a UM mailbox policy, or a UM auto attendant to use the appropriate dialing rule. After you configure the UM mailbox policy to use a dialing rule group, the dialing restrictions configured apply to all UM-enabled users who are associated with the UM mailbox policy. For example, you can configure a dialing rule group that doesn't require users who are associated with the dial plan to dial an outside line access code when they place a call to an in-country/region telephone number. You can configure the following:
   - **In-country/region dialing rules** Use this box to add, remove, or edit in-country/region dialing rule groups used by UM mailbox policies. To create a dialing rule, click **+**. To edit an existing dialing rule, click ✎. To remove a dialing rule, click **–**. When you create a dialing rule, add the following information on the **New dialing rule** page:
     - **Dialing rule name** Use this text box to enter the name for the dialing rule you are creating. You can use the same name to collect several rules in a group and then enable or disable them under **Dialing authorization**. The name can be up to 32 characters long.
     - **Number pattern to transform (number mask)** Use this text box to enter the number pattern to transform before dialing, for example 91425xxxxxxx. If a user enters a number that matches this pattern, UM will transform the number dialed into a dialed number before placing the call. You can only enter numbers and the wildcard character, "x".
     - **Dialed number** Use this text box to enter the number you want to dial that matches the number pattern you set in the **Number pattern to transform (number mask)**. The dialed number is used to determine the actual dial string sent to the VoIP gateway or IP PBX. This number can be different from the number obtained by Unified Messaging for the outgoing call. However, your PBX or IP PBX can also be configured to omit the area code for local

calls and can be configured for private voice numbering plans. Any wildcard characters (*x*) in the dial string are replaced with the digits from the original number that were matched by the number mask on the dialing rule. An example of a valid dialed number is 9*xxxxxxx*. This field can contain only numbers and the character *x*.

- **Comment** Use this text box to put in a comment or description for the dialing rule that you're adding or modifying. By default, this text box is blank.

Note:

If you are integrating with Office Communications Server 2007 R2 or Microsoft Lync Server, you'll probably find it unnecessary to configure dialing rules or dialing rule groups in Unified Messaging. Office Communications Server 2007 R2 and Lync Server are designed to perform call routing and number translation for users in your organization, and will also do this when the calls are made on behalf of users.

- **International rules** Use this text box to add, remove, or edit international dialing rule groups used by UM mailbox policies.
  - **Dialing rule name** Use this text box to enter the name for the dialing rule you are creating. You can use the same name to collect several rules in a group and then enable or disable them under **Dialing authorization**. The name can be up to 32 characters long.
  - **Number pattern to transform (number mask)** Use this text box to enter the number pattern to transform before dialing, for example 91425xxxxxxx. If a user enters a number that matches this pattern, UM will transform the number dialed into a dialed number before placing the call. You can only enter numbers and the wildcard character, "x".
  - **Dialed number** Use this text box to enter the number you want to dial that matches the number pattern you set in **Number pattern to transform (number mask)**. The dialed number is used to determine the actual dial string sent to the VoIP gateway or IP PBX. This number can be different from the number obtained by Unified Messaging for the outgoing call. However, your PBX or IP PBX can also be configured to omit the area code for local calls and can be configured for private voice numbering plans. Any wildcard characters (*x*) in the dial string are replaced with the digits from the original number that were matched by the number mask on the dialing rule. An example of a valid dialed number is 9*xxxxxxx*. This field can contain only numbers and the character *x*.
  - **Comment** Use this text box to put in a comment or description for the dialing rule that you're adding or modifying. By default, this text box is blank.

Note:

For on-premises deployments, if you are integrating with Office Communications Server 2007 R2 or Microsoft Lync Server, you'll probably find it unnecessary to configure dialing rules or dialing rule groups in Unified Messaging. Office Communications Server 2007 R2 or Lync Server are designed to perform call routing and number translation for users in your organization, and will also do this when the calls are made on behalf of users.

9. **Dialing authorization** Use this page to select dialing rules for callers who call in to an Outlook Voice Access number configured on a UM dial plan. You can restrict the type of calls placed by callers when an unauthenticated user or an Outlook Voice Access user calls in to an Outlook Voice Access number configured on a dial plan by configuring dialing rule groups and dialing

restrictions. You can configure the following:

- **Calls in the same UM dial plan** Select this check box to let users who call in to an Outlook Voice Access number configured on a dial plan place or transfer calls to an extension number associated with a UM-enabled user who is within the same dial plan. By default, this setting is enabled.

When you disable this setting, users who call in to the Outlook Voice Access number won't be able to place or transfer calls to any users who aren't UM-enabled, to other extension numbers, or to UM-enabled users who are associated with the same dial plan. This is because the **Allow calls to any extension** setting is disabled by default.

- **Allow calls to any extension** When this setting is disabled, users who call in to an Outlook Voice Access number on the dial plan can't place calls to users who aren't UM-enabled or to other extension numbers not associated with a UM-enabled user. However, they can place a call or transfer a call to extension numbers associated with UM-enabled users. This is because the **Calls in the same UM dial plan** setting is enabled by default. The **Allow calls to any extension** setting is disabled by default.

When this setting is enabled, users who call in to an Outlook Voice Access number configured on the dial plan can place calls to users who aren't UM-enabled, to other extension numbers not associated with a UM-enabled user, and to UM-enabled users. This is because the **Calls in the same UM dial plan** setting is enabled by default.

You can enable this setting in an environment where not all users have been UM-enabled. This setting is also useful when you want to allow users who call in to a Outlook Voice Access number configured on a dial plan to call extension numbers that aren't associated.

- **Authorized in-country/region dialing rule groups** Use this section to add or remove allowed in-country/region dialing rules. By default, there are no in-country/region dialing rules configured on UM dial plans.

In-country/region dialing rule groups are used to allow or restrict the telephone numbers within a country or region that any user who has dialed in to the subscriber access number can dial. This helps prevent unnecessary or unauthorized telephone calls and charges.

To add in-country/region dialing rules, you must first create the appropriate in-country/region dialing rule on the dial plan, and then add the appropriate dialing rule entries on the dialing rule. After you create the required dialing rules on the dial plan, you must then add the dialing rule to the list of dialing authorizations on the **Dialing authorization** page on the dial plan.

In-country/region dialing rule groups can be used to allow or restrict access to telephone numbers within a country or region. This is applied to all users who have called in to an Outlook Voice Access number.

- **Authorized international dialing rule groups** Use this section to add or remove allowed international dialing rules. By default, there are no international dialing rules configured on UM dial plans.

International dialing rules are used to allow or restrict the telephone numbers outside a country or region that any user who has dialed in to the Outlook Voice Access number can dial. This helps prevent unnecessary or unauthorized telephone calls and charges.

To add international dialing rule groups, you must first create the appropriate international dialing rules on the dial plan, and then add the appropriate dialing rule entries. After you create the required dialing rules on the dial plan, you must then add the dialing rule to the list of dialing authorizations on the **Dialing authorization** page on the dial plan.

International dialing rule groups can be used to allow or restrict access to telephone numbers outside a country or region. This is applied to all users who have called in to an Outlook Voice Access number.

10. **Transfer & search** Use this page to configure the UM dial plan features. Several features can be configured on the UM dial plan. These include transferring calls, sending voice messages, and searching for users. You can configure the following:
    - **Allow callers to** Use these settings to determine how users who call in to an Outlook Voice Access number can contact users. You can configure the following:
        - **Transfer to users** Select this check box to enable Outlook Voice Access users to transfer calls to users. By default, this option is enabled. This lets users associated with the dial plan transfer calls to users in the same UM dial plan. After you select this check box, you can set the group of users callers can search for by selecting the appropriate option under the **Allow callers to search for users by name or alias** section on this page.

If you disable this option, Outlook Voice Access won't allow callers to be transferred to any users in the dial plan.

        - **Leave voice messages without ringing a user's phone** Select this check box to enable callers to send voice messages to users. By default, this option is enabled. This lets Outlook Voice Access users who are associated with the dial plan send voice messages to users in the same UM dial plan. After you select this check box, you can set the group of users callers can search for by selecting the appropriate option under the **Allow callers to search for users by name or alias** section on this page.

If you disable this option, Outlook Voice Access won't invite callers to send a voice message during a system prompt.

    - **Allow callers to search for users by name or alias** Use these options to determine a grouping of users that can be searched. By default, the **In this dial plan only** option is selected. However, you can change the grouping of users. Choose from the following options:
        - **In this dial plan only** Use this option to allow callers who connect to Outlook Voice Access to locate and contact users who are within the dial plan that they are a member of.
        - **In the entire organization** Use this option to allow callers who connect to Outlook Voice Access to locate and contact anyone who is listed in the entire organization. This includes all users who are mailbox-enabled or UM-enabled users in all dial plans.
        - **Only on this auto attendant** Use this list to allow Outlook Voice Access users to connect to a UM auto attendant and then potentially connect to another auto attendant you have configured. You must create this auto attendant to allow callers to be transferred to another auto attendant that's specified.
        - **Only for this extension** Use this option to allow Outlook Voice Access users to connect to an extension number that you specify in the field for this option. This field accepts only numeric digits. The number of digits that you define in this field must match the number of

digits configured on the dial plan associated with the auto attendant.

- **Information to include for users with the same name** Use this field to select how the dial plan differentiates between users who have the same or similar names. When a caller is prompted to enter letters or say the person's name to find a particular user in the organization, sometimes more than one name matches the caller's input. If there are two users with the same name, UM will use one of the following ways to add additional information to the user's name. For example, if you select **Department**, when an Outlook Voice Access user calls in to Outlook Voice Access and searches for a user and there are duplicate or similar names in the directory, the caller will hear the user's name and department, for example:
  - System: "Welcome to Outlook Voice Access. Please enter your PIN and press the pound key."
  - Caller inputs their PIN followed by the # key.
  - System: "Please say voice mail, email, calendar, personal contacts, directory, or personal options."
  - Caller: "Directory"
  - System: "Directory search. Please note, for the following tasks the system requires you to use your telephone keypad rather than speaking. Use the keypad to spell the name of the person you're trying to find, last name first, or to spell the first part of their email address, press the pound key twice, if you know the extension, press the pound key."
  - Caller uses the key pad and inputs "smithtony" and presses the # key.
  - System: "For Tony Smith, research, press 1. For Tony Smith, administration, press 2. For Tony Smith, technical support, press 3."
  - Caller presses the appropriate key on the keypad and the call is transferred to the user.

By default, all UM auto attendants associated with this dial plan inherit this setting. However, you can change this setting on each UM auto attendant you create.

Select one of the following methods for providing callers with more information to help them locate the correct user in the organization:

- **None** No additional information is given when matches are listed. By default, this method is selected.
- **Title** The voice mail system includes each user's title when matches are listed.
- **Department** The voice mail system includes each user's department when matches are listed.
- **Location** The voice mail system includes each user's location when matches are listed.
- **Prompt for alias** The voice mail system prompts the caller for the user's alias.

11.After you configure the required settings, click **Save** to save your changes.

Use the Shell to configure UM dial plan settings

This example configures a UM dial plan named `MyDialPlan` to use 9 for the outside line access code.

```
Set-UMDialplan -Identity MyDialPlan -OutsideLineAccessCode
9
```

This example configures a UM dial plan named `MyDialPlan` to use a welcome greeting.

```
Set-UMDialplan -Identity MyDialPlan -WelcomeGreetingEnabled
$true -WelcomeGreetingFilename welcome.wav
```

This example configures a UM dial plan named `MyDialPlan` with dialing rules.

```
$csv=import-csv "C:\MyInCountryGroups.csv"
Set-UMDialPlan -Identity MyDialPlan -
ConfiguredInCountryGroups $csv
Set-UMDialPlan -Identity MyDialPlan -AllowedInCountryGroups
"local, long distance"
```

Use the Shell to view UM dial plan settings

This example displays a list of all the UM dial plans.

```
Get-UMDialplan
```

This example displays a formatted list of all of the settings on a UM dial plan named `MyUMDialPlan`.

```
Get-UMDialplan -Identity MyUMDialPlan | Format-List
```

Add Mailbox and Client Access servers to a SIP URI dial plan

Connect your voice mail system to your telephone network > UM dial plans > UM dial plan procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-16*

You can add Client Access and Mailbox servers to SIP URI dial plans. Client Access and Mailbox servers can't be associated with Telephone Extension or E.164 dial plans, but the servers will answer all incoming calls.

If you're deploying Microsoft Lync Server, to enable outbound calling to work correctly, you must manually add all Client Access and Mailbox servers to all SIP URI dial plans that you've created for Lync Server.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a SIP URI dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to add a Mailbox server to a SIP URI dial plan

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Mailbox server you want to modify, and then click **Edit** .
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Service settings** > **Associated dial plans**, click **Add +**.
5. In the **Select a UM Dial Plan** window, select the SIP URI dial plan, click **Add**, click **OK**, and then click **Save**.

Use the Shell to add a Mailbox server to a SIP URI dial plan

This example adds the Mailbox server named `MyMailboxServer` to a SIP URI dial plan named `MySIPDialPlan` and prevents it from accepting new calls. It also sets the startup mode to Dual mode, which enables the Mailbox server to accept TCP and TLS requests.

```
Set-UMService -Identity MyMailboxServer -DialPlans
MySIPDialPlan -Status Disabled -UMStartupMode Dual
```

This example adds the Mailbox server named `MyMailboxServer` to two SIP dial plans, named `MySIPDialPlan` and `MySIPDialPlan2`, and sets the following:

- Allows both IPv4 and IPv6 addresses.
- Sets the maximum number of incoming calls to 50.
- Configures the SIP access service for Lync Server.

```
Set-UMService -Identity MyMailboxServer -DialPlans
```

```
MySIPDialPlan, MySIPDialPlan2 -IPAddressFamily Any -
MaxCallsAllowed 50 -SipAccessService
northamerica.lyncpoolna.contoso.com
```

Use the EAC to add a Client Access server to a SIP URI dial plan

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Client Access server you want to modify, and then click **Edit** .
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Call Router settings** > **Associated dial plans**, click **Add +**.
5. In the **Select a UM Dial Plan** window, select the SIP URI dial plan, click **Add**, click **OK**, and then click **Save**.

Use the Shell to add a Client Access server to a SIP URI dial plan

This example adds the Client Access server named `MyClientAccessServer` to a SIP URI dial plan named `MySIPDialPlan`. It also sets the startup mode to Dual mode, which enables the Client Access server to accept TCP and TLS requests.

```
Set-UMCallRouterSettings -DialPlans MySIPDialPlan -Server
MyClientAccessServer -UMStartupMode Dual
```

This example adds the Client Access server named `MyClientAccessServer` to two SIP dial plans, named `MySIPDialPlan` and `MySIPDialPlan2`, and allows the server to use both IPv4 and IPv6 addresses.

```
Set-UMCallRouterSettings -DialPlans MySIPDialPlan,
MySIPDialPlan2 -IPAddressFamily Any -Server
MyClientAccessServer
```

Remove Mailbox and Client Access servers from a SIP URI dial plan

Connect your voice mail system to your telephone network > UM dial plans > UM dial plan procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-16*

You can remove Client Access and Mailbox servers from SIP URI dial plans. When you're deploying Microsoft Lync Server, to enable outbound calling to work correctly, you must manually add all Client Access and Mailbox servers to the SIP URI dial plans that you've created for Lync Server. However, you may need to remove a Client Access or Mailbox server from your Lync deployment, for example, if you're performing maintenance or taking the server offline.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a SIP URI dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to remove a Mailbox server from a SIP URI dial plan

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Mailbox server you want to modify, and then click **Edit** .
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Service settings** > **Associated dial plans**, locate the SIP URI dial plan to remove, click **Remove** —, and then click **Save**. If you want to remove more than one SIP URI dial plan, press and hold the CTRL key, select the dial plans you want to remove, and then click **Save**.

Use the Shell to remove a Mailbox server from a SIP URI dial plan

This example removes the Mailbox server named `MyMailboxServer` from a SIP URI dial plan named

MySIPDialPlan.

```
$dp= Get-UMDialPlan "MySIPDialPlan"
$s=Get-UMService MyMailboxServer
$s.dialplans-=$dp.identity
Set-UMService -id MyMailboxServer -dialplans:$s.dialplans
```

In this example, there are three SIP URI dial plans: SipDP1, SipDP2 and SipDP3. This example removes the Mailbox server named `MyMailboxServer` from the SipDP3 dial plan.

```
Set-UMService -id MyMailboxServer -DialPlans SipDP1,SipDP2
```

In this example, there are two SIP URI dial plans: SipDP1 and SipDP2. This example removes the Mailbox server named `MyMailboxServer` from the SipDP2 dial plan.

```
Set-UMService -id MyMailboxServer -DialPlans SipDP1
```

This example removes the Mailbox server named `MyMailboxServer` from all SIP dial plans.

```
Set-UMService -id MyUMServer -DialPlans $null
```

Use the EAC to remove a Client Access server from a SIP URI dial plan

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Client Access server you want to modify, and then click **Edit** .
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Call Router settings** > **Associated dial plans**, locate the SIP URI dial plan to remove, click **Remove** —, and then click **Save**. If you want to remove more than one SIP URI dial plan, press and hold the CTRL key, select the dial plans you want to remove, and then click **Save**.

Use the Shell to remove a Client Access server from a SIP URI dial plan

This example removes the Client Access server named `MyClientAccessServer` from a SIP URI dial plan named `MySIPDialPlan`.

```
$dp= Get-UMDialPlan "MySIPDialPlan"
$s=Get-UMCallRouterSettings MyClientAccessServer
$s.dialplans-=$dp.identity
Set-UMCallRouterSettings -id MyClientAccessServer -
```

```
dialplans:$s.dialplans
```

In this example, there are three SIP URI dial plans: SipDP1, SipDP2 and SipDP3. This example removes the Client Access server named `MyClientAccessServer` from the SipDP3 dial plan.

```
Set-UMCallRouterSettings -id MyClientAccessServer -
DialPlans SipDP1,SipDP2
```

In this example, there are two SIP URI dial plans: SipDP1 and SipDP2. This example removes the Client Access server named `MyClientAccessServer` from the SipDP2 dial plan.

```
Set-UMCallRouterSettings -id MyClientAccessServer -
DialPlans SipDP1
```

This example removes the Client Access server named `MyClientAccessServer` from all SIP dial plans.

```
Set-UMCallRouterSettings -id MyClientAccessServer -
DialPlans $null
```

Change the audio codec

Connect your voice mail system to your telephone network > UM dial plans > UM dial plan procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

Unified Messaging can use one of four codecs for creating voice mail messages: MP3, Windows Media Audio (WMA), Group System Mobile (GSM) 06.10, and G.711 Pulse Code Modulation (PCM) Linear. By default, when you create a Unified Messaging (UM) dial plan, the UM dial plan uses the MP3 audio codec to record voice messages. The MP3 audio format is a popular audio format that is used across multiple operating systems, email clients, and MP3 players. After the UM dial plan is created, you can configure the UM dial plan to use one of the other audio formats including the WMA, GSM 06.10, or G.711 PCM Linear audio codecs. To listen to the voice message, a mobile phone or computer must have a compatible audio software application installed.

For additional tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging

permissions topic.

- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the audio codec on a Unified Messaging dial plan

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. In **Settings**, under **Audio codec**, use the drop-down list to select one the following:
   - MP3
   - WMA
   - GSM
   - G711
5. Click **Save**.

Use the Shell to change the audio codec on a Unified Messaging dial plan

This example sets the audio codec on a UM dial plan named `MyUMDialPlan` to G.711.

```
Set-UMDialPlan -Identity MyUMDialPlan -AudioCodec G711
```

This example sets the audio codec on a UM dial plan named `MyUMDialPlan` to WMA.

```
Set-UMDialPlan -Identity MyUMDialPlan -AudioCodec Wma
```

Configure the maximum call duration

Connect your voice mail system to your telephone network > UM dial plans > UM dial plan

procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-09*

You can specify the maximum number of minutes that an incoming call can be connected to the system without being transferred to a valid extension number before the call is ended. For most organizations, this value should be set to the default: 30 minutes. This setting applies to all calls, including incoming Outlook Voice Access calls, voice calls internal to your organization, voice calls into Unified Messaging (UM) auto attendants, and fax calls placed from outside your organization.

This value can be set to a number from 10 through 120. Setting this value too low can cause incoming calls to be disconnected before they're completed. For example, if your organization receives many large fax messages, you may want to consider increasing this value from the default so that all the pages of fax messages are received.

For additional tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the maximum call duration

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. In **Settings**, under **Maximum call duration (minutes)**, enter the number in minutes.
5. Click **Save**.

Use the Shell to configure the maximum call duration

This example sets the maximum call duration to 10 minutes on a UM dial plan named `MyUMDialPlan`.

```
Set-UMDialPlan -identity MyUMDialPlan -MaxCallDuration 10
```

Configure the maximum recording duration

Connect your voice mail system to your telephone network > UM dial plans > UM dial plan procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-09*

You can specify the maximum number of minutes allowed for each voice recording when a caller leaves a voice mail message. This value can be set to a number from 1 through 100. For most organizations, this value should be set to the default of 20 minutes. Setting this value too low can cause long voice messages to be disconnected before they're completed. Setting this value too high lets users save lengthy voice messages in their Inboxes.

This setting is important if you've implemented strict disk quotas for users. It must be set to a lower value than the one set for **Maximum call duration (minutes)**.

For additional tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the maximum recording duration

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. In **Settings**, under **Maximum recording duration (minutes)**, enter the number in minutes.
5. Click **Save**.

Use the Shell to configure the maximum recording duration

This example sets the maximum recording duration to 10 minutes for a UM dial plan named `MyUMDialPlan`.

```
Set-UMDialPlan -identity MyUMDialPlan -MaxRecordingDuration
10
```

Configure the recording idle time-out value

Connect your voice mail system to your telephone network > UM dial plans > UM dial plan procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-11*

You can specify the number of seconds of silence that the system allows when a voice message is being recorded before the call is ended. For most organizations, this value should be set to the default of 5 seconds.

This value can be set from 2 through 10. Setting this value too low can cause the system to disconnect callers before they've finished leaving their voice messages. Setting this value too high allows lengthy silences in voice messages.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the recording idle time-out value

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. In **Settings**, under **Recording idle time out (seconds)**, enter the number in seconds.
5. Click **Save**.

Use the Shell to configure the recording idle time-out value

This example sets the recording idle time-out value to 10 for a UM dial plan named `MyUMDialPlan`.

```
Set-UMDialPlan -identity MyUMDialPlan -RecordingIdleTimeout
10
```

Configure the VoIP security setting

Connect your voice mail system to your telephone network > UM dial plans > UM dial plan procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can enable Voice over IP (VoIP) security for a Unified Messaging (UM) dial plan. By default, when a UM dial plan is created, it will use Unsecured mode or no encryption. Exchange servers can answer calls for single or multiple UM dial plans and can answer calls for dial plans that have different VoIP security settings.

When you configure a UM dial plan to use Session Initiation Protocol (SIP) secured or Secured mode, the Exchange servers that answer calls for the UM dial plan will encrypt the SIP signaling traffic (for SIP secured mode) or both the Realtime Transport Protocol (RTP) media channels and the SIP signaling traffic (for Secured mode).

Important:

For on-premises and hybrid deployments, when you configure the SipTCPListeningPort, SipTLSListeningPort, or the UMStartUpMode on a Client Access server running the Microsoft Exchange Unified Messaging Call Router service or a Mailbox server running the Microsoft Exchange Unified Messaging service, you will need to configure the Windows Firewall rules correctly to allow SIP and RTP network traffic.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure VoIP security on a UM dial plan

1. In the EAC, navigate to **Unified Messaging** > **UM Dial Plans**, select the UM dial plan on which you want to change the VoIP security, and then click **Edit** ✎.
2. On the **UM Dial Plan** page, click **Configure**.
3. In **General**, under **VoIP security mode**, select one of the following options:
   - **SIP secured**
   - **Unsecured** (default)
   - **Secured**

4. Click **Save**.

Use the Shell to configure VoIP security on a UM dial plan

This example configures a UM dial plan named `MySecureDialPlan` to encrypt both SIP and RTP traffic.

```
Set-UMDialPlan -identity MySecureDialPlan -VoIPSecurity
Secured
```

This example configures a UM dial plan named `MySecureDialPlan` to encrypt SIP but not encrypt RTP traffic.

```
Set-UMDialPlan -identity MySecureDialPlan -VoIPSecurity
SIPsecured
```

This example configures a UM dial plan named `MySecureDialPlan` to not encrypt SIP and RTP traffic.

```
Set-UMDialPlan -identity MySecureDialPlan -VoIPSecurity
Unsecured
```

Configure a dial plan for users who have similar names

Connect your voice mail system to your telephone network > UM dial plans > UM dial plan procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

You can configure a Unified Messaging (UM) dial plan to specify the information that is provided for callers when users have the same or similar names. UM uses this setting to differentiate between users who have the same or similar names and provide this information to callers. When a caller or an Outlook Voice Access user is prompted to enter letters to find a particular user, sometimes more than one name matches the caller's input. You can use one of the available options for providing the caller with more information to help them locate the user they're trying to reach.

You can set this setting on both UM dial plans and UM auto attendants. When a UM auto attendant is created, it inherits this setting from the dial plan associated with the auto attendant. By default, this setting isn't configured for dial plans, so no additional information will be given to callers to help them locate the correct user.

Note:
For the information that will be included for users with similar names to work correctly, you must provide the title, department, and location information for the recipients in your Microsoft Exchange organization.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure a UM dial plan for users with similar names

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM dial plan** page, click **Configure** > **Transfer & search**, and under **Information to include for users with the same name**, select one of the following options:
   - **Title** The dial plan includes each user's title when it finds two or more users with similar names.
   - **Department** The dial plan includes each user's department when it finds two or more users with similar names.
   - **Location** The dial plan includes each user's location when it finds two or more users with similar names.
   - **None** The dial plan won't include any additional information when users have similar names. Although this is the default setting, we recommend that you include one of the available options for callers. If you don't, callers won't be able to tell the difference between two or more users with similar names.
   - **Prompt For alias** The dial plan prompts the caller for the user's alias. An alias is the part of

the user's email or SMTP address that is before the at (@) symbol.

3. Click **Save**.

Use the Shell to configure a UM dial plan for users with similar names

This example sets the information to include with users with similar names to prompt for the user's alias on a UM dial plan named `MyDialPlan`.

```
Set-UMDialplan -Identity MyDialPlan -
MatchedNameSelectionMethod PromptForAlias
```

This example sets the information to include with users with similar names to department on a UM dial plan named `MyDialPlan`.

```
Set-UMDialplan -Identity MyDialPlan -
MatchedNameSelectionMethod Department
```

This example sets the information to include with users with similar names to location on a UM dial plan named `MyDialPlan`.

```
Set-UMDialplan -Identity MyDialPlan -
MatchedNameSelectionMethod Location
```

Delete a UM dial plan

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-11*

You can delete an existing Unified Messaging (UM) dial plan. When you delete the UM dial plan, it will no longer be available for UM IP gateways, UM mailbox policies, and UM hunt groups. You can't delete a UM dial plan if it's referenced by or associated with UM mailbox policies, UM auto attendants, UM IP gateways, or UM hunt groups.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to delete an existing dial plan

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to delete, and then click **Delete** .
3. On the warning page, click **Yes**.

Use the Shell to delete an existing dial plan

This example deletes a UM dial plan named `MyUMDialPlan`.

```
RemoveUMDialplan -identity MyUMDialPlan
```

UM IP gateways

Exchange Server 2013 > Unified Messaging > Connect your voice mail system to your telephone network >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-06-24*

A Unified Messaging (UM) IP gateway represents a physical Voice over IP (VoIP) gateway, IP Private Branch eXchange (PBX), or session border controller (SBC) hardware device. Before a VoIP gateway, IP PBX, or SBC can be used to answer incoming calls and send outgoing calls for voice mail users, a UM IP gateway must be created in the directory service.

Contents

IPv6 support for UM IP gateways

Enabling and disabling a UM IP gateway

Overview of UM IP gateways

Traditionally, *gateway* is a term that describes a physical device that connects two incompatible networks. With Exchange Unified Messaging and other unified messaging solutions, the VoIP gateway is used to translate between the Public Switched Telephone Network (PSTN)/Time Division Multiplex (TDM) or circuit-switched based telephony network and an IP or packet-switched data network. An IP PBX also translates between the PSTN network and a packet-switched network, so when an IP PBX is used, a VoIP gateway isn't required. A VoIP gateway is only required if you are connecting a legacy PBX hardware device to your UM deployment.

Note:

A packet-switched network is a network in which packets (messages or fragments of messages) are individually routed between devices such as routers, switches, VoIP gateway, IP PBXs and SBCs. This contrasts with a circuit-switched network that sets up a dedicated connection between the two nodes for their exclusive use for the duration of the communication.

Exchange Unified Messaging relies on the ability of the VoIP gateway to translate TDM or telephony circuit-switched based protocols, such as Integrated Services Digital Network (ISDN) or QSIG, from a PBX to protocols based on VoIP or IP, such as Session Initiation Protocol (SIP), Realtime Transport Protocol (RTP), or T.38 for real-time facsimile transport.

IP PBXs are also used when connecting a circuit-switched telephony network to a data or packet-switched network. They are also used to translate circuit-switched protocols to protocols based on VoIP or IP, such as SIP, RTP, and Secure RTPC (SRTP).

Session Border Controllers (SBCs) are somewhat different than VoIP gateways and IP PBXs. Instead of connecting a circuit-switched network to a packet-switched network, they're used to connect two data networks over a public network like the Internet or over a private WAN connection. In Unified Messaging, SBCs are used in a hybrid deployment of UM in which UM uses some components that are located on-premises and others, such as mailboxes, that are located in the cloud.

VoIP device configurations

Although there are many types and manufacturers of PBXs, VoIP gateways, IP PBXs, and SBCs, there are basically three types of VoIP device configurations:

- **IP PBX** A single device that translates between the PSTN/TDM or circuit-switched based telephony network and an IP or packet-switched data network
- **PBX (legacy) and a VoIP gateway** Two separate components that together translate between the PSTN/TDM or circuit-switched telephony network and an IP or packet-switched data network
- **SBC** Single or multiple devices that connect two types of IP-based networks such as a LAN and a

datacenter.

To support Unified Messaging, one or both types of IP/VoIP device configurations are used when connecting a telephony network infrastructure to a data network infrastructure or connecting an on-premises deployment with a UM deployment in the cloud.

UM IP gateways

The UM IP gateway contains one or more UM hunt groups and configuration settings. UM hunt groups are used to link a UM IP gateway to a UM dial plan. The combination of the UM IP gateway and a UM hunt group establishes a link between a VoIP gateway, IP PBX, or SBC and a UM dial plan. By creating multiple UM hunt groups, you can associate a single UM IP gateway with multiple UM dial plans.

After you create a UM IP gateway, the Exchange servers linked to the UM IP gateway will send a SIP OPTIONS request to the VoIP gateway, IP PBX, or SBC to ensure that the device is responsive. If the VoIP gateway, IP PBX, or SBC doesn't respond to the request, an Exchange server will log an event with ID 1400 stating that the request failed. If this happens, make sure that the VoIP gateway, IP PBX, or SBC is available and online and that the Unified Messaging configuration is correct.

A Mailbox server communicates only with VoIP gateways, IP PBXs, or SBCs listed as trusted SIP peers. In some cases, if two VoIP gateways, IP PBXs, or SBCs are configured to use the same IP address, an event with ID 1175 will be logged. Unified Messaging protects against unauthorized requests by retrieving the internal URL of the Unified Messaging Web services virtual directory and then uses the URL to build the list of FQDNs for the trusted SIP peers. When two FQDNs are resolved to the same IP address, this event is logged.

IPv6 support for UM IP gateways

Internet Protocol version 6 (IPv6) is the most recent version of the Internet Protocol (IP). IPv6 is intended to correct many of the shortcomings of IPv4, which was the previous version of the IP. In Microsoft Exchange Server 2010 on-premises and hybrid deployments, IPv6 was supported only when IPv4 was also used.

In Exchange 2013 on-premises and hybrid deployments, UM-related components and speech services run only on Client Access and Mailbox servers. Because the UM architecture has changed and now requires Unified Communications Managed API (UCMA) v4.0 to support both IPv4 and IPv6 as well as other Exchange features, the Client Access and Mailbox servers that have Unified Messaging components and services fully support IPv6 networks and doesn’t require IPv4.

In on-premises, hybrid, and Exchange Online deployments, both enterprise and Exchange Online UM administrators can use IPv6 when they connect UM to IPv6-capable devices, including devices such as routers, IP gateways, IP PBXs, and Microsoft Office Communications Server 2007 R2 and Microsoft Lync servers. However, for interoperability and backward compatibility, IPv4 can be used instead without additional configuration changes if the *IPAddressFamily* parameter is set to Any on

UM IP gateways.

Exchange UM must still communicate directly with SIP peers (VoIP gateways, IP PBXs, and SBCs) that may not support IPv6 in their software or firmware. If they don't support IPv6, UM must be able to communicate directly with SIP peers that use IPv4. For hosted voice mail, UM communicates with customer equipment through SBCs, Lync Server 2010, or Lync Server 2013. In hosted environments, IPv6 SIP-aware clients such as SBCs and Lync servers can be deployed to handle the IPv6-to-IPv4 conversion process.

For on-premises and hybrid deployments after you install your Client Access and Mailbox servers, and for Exchange Online UM deployments, you need to create UM IP gateways. If you need your UM IP gateways to support IPv6, you must also:

1. Create a new UM IP gateway or configure an existing UM IP gateway with an IPv6 address for each of the IP gateways, IP PBXs, or SBCs on your network. When you're creating and configuring the required UM IP gateways, you must add the IPv6 address or the Fully Qualified Domain Name (FQDN) for the UM IP gateway. If you're adding the FQDN to the UM IP gateway, you must have created the correct DNS records to resolve the UM IP gateway FQDN to the IPv6 address. If you have an existing UM IP gateway, you can use the **Set-UMIPgateway** cmdlet to configure the IPv6 address or FQDN.
2. Configure the *IPAddressFamily* parameter on each UM IP gateway. To enable the VoIP gateway to accept IPv6 packets, you must set the UM IP gateway to either accept both IPv4 and IPv6 connections, or accept only IPv6 connections, by using the **Set-UMIPgateway** cmdlet.
3. After you've configured your UM IP gateways, you must also configure the VoIP gateways, IP PBXs, and SBCs on your network to support IPv6. For details, see your hardware vendor for a list of devices that support IPv6 and how to correctly configure them.

Note:

The maximum number of UM IP gateways per dial plan is 200. If you create more than 200 the UM service won't start.

Enabling and disabling a UM IP gateway

By default, a UM IP gateway is left in an enabled state after it's created. However, the UM IP gateway can be enabled or disabled. If you disable a UM IP gateway, you can set it to force all Exchange servers to drop existing calls. Alternatively, you can set it to force the Exchange servers associated with the UM IP gateway to stop handling any new calls presented by the VoIP gateway, IP PBX, or SBC.

If you're integrating Unified Messaging with Office Communications Server R2 or Microsoft Lync Server, you must allow only one UM IP gateway to make outgoing calls for users, and disable outbound calling on all other UM IP gateways associated with your SIP URI dial plans. Use either the Shell or the EAC to disable outbound calling.

When selecting the UM IP gateway through which to allow outgoing calls for on-premises and hybrid deployments, choose the one that's likely to handle the most traffic. Don't allow outgoing

traffic through a UM IP gateway that connects to a pool of Lync Server Directors. This is necessary to ensure that outbound calls to external users placed by a Mailbox server running the Microsoft Exchange Unified Messaging service (for example, in Play-on-Phone scenarios) reliably traverse the corporate firewall.

UM IP gateway procedures

Unified Messaging > Connect your voice mail system to your telephone network > UM IP gateways >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-05-03*

Create a UM IP gateway

Manage a UM IP gateway

Enable a UM IP gateway

Disable a UM IP gateway

Configure a fully qualified domain name

Configure the IP address

Configure the listening port

Delete a UM IP gateway

Create a UM IP gateway

Connect your voice mail system to your telephone network > UM IP gateways > UM IP gateway procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-16*

When you create a Unified Messaging (UM) IP gateway, you enable Exchange servers to connect to a new Voice over IP (VoIP) gateway, a Private Branch eXchange (PBX) enabled for Session Initiation Protocol (SIP), an IP PBX, or a session border controller (SBC). Immediately after you create a UM IP gateway, you should create a new UM hunt group and then associate the UM hunt group with the UM IP gateway. You can associate the UM IP gateway with one or more UM dial plans by creating one or more UM hunt groups.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to create a UM IP gateway

1. In the EAC, navigate to **Unified Messaging** > **UM IP gateways**, and then click **New +**.
2. On the **New UM IP gateway** page, enter the following information:
   - **Name** Use this box to specify a unique name for the UM IP gateway. This is a display name that appears in the EAC. If you have to change the display name of the UM IP gateway after it's been created, you must first delete the existing UM IP gateway, and then create another UM IP gateway that has the name that you want. The UM IP gateway name is required, but it's used for display purposes only. Because your organization may use multiple UM IP gateways, we recommend that you use meaningful names for your UM IP gateways. The maximum length of a UM IP gateway name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
   - **Address** You can configure a UM IP gateway with either an IP address or a fully qualified domain name (FQDN). Use this box to specify the IP address configured on the VoIP gateway, SIP-enabled PBX, IP PBX, or SBC, or an FQDN. This box accepts only FQDNs that are valid and formatted correctly.

You can enter alphabetical and numeric characters in this box. IPv4 addresses, IPv6 addresses, and FQDNs are supported. If you want to use mutual Transport Layer Security (mutual TLS) between a UM IP gateway and a dial plan operating in either SIP secured or Secured mode, you must configure the UM IP gateway with an FQDN. You must also configure it to listen on port 5061 and verify that any VoIP gateways or IP PBXs have also been configured to listen for mutual TLS requests on port 5061. To configure a UM IP gateway, run the following command: `Set-UMIPGateway -identity MyUMIPGateway -Port 5061`.

If you use an FQDN, you must also make sure that you've correctly configured a DNS host record for the VoIP gateway so that the host name will be correctly resolved to an IP address. Also, if you use

an FQDN instead of an IP address, and the DNS configuration for the UM IP gateway is changed, you must disable and then enable the UM IP gateway to make sure that configuration information for the UM IP gateway is updated correctly.

- **UM dial plan** Click **Browse** to select the UM dial plan that you want to associate with the UM IP gateway. When you select a UM dial plan to associate with a UM IP gateway, a default UM hunt group is also created and associated with the UM dial plan that you selected. If you don't select a UM dial plan, you must manually create a UM hunt group and then associate that UM hunt group with the UM IP gateway that you create.

3. Click **Save**.

Use the Shell to create a UM IP gateway

This example creates a UM IP gateway named `MyUMIPGateway` that enables Exchange servers to start accepting calls from a VoIP gateway, a PBX enabled for SIP, an IP PBX, or an SBC that has an IP address of 10.10.10.1.

```
New-UMIPGateway -Name MyUMIPGateway -Address 10.10.10.1
```

This example creates a UM IP gateway named `MyUMIPGateway` that enables Exchange servers to start accepting calls from a VoIP gateway, a PBX enabled for SIP, an IP PBX, or an SBC that has an FQDN of MyUMIPGateway.contoso.com and listens on port 5061.

```
New-UMIPGateway -Name MyUMIPGateway -Address
"MyUMIPGateway.contoso.com" -Port 5061
```

This example creates a UM IP gateway named `MyUMIPGateway` and prevents the UM IP gateway from accepting incoming calls or sending outgoing calls, sets an IPv6 address, and allows the UM IP gateway to use IPv4 and IPV6 addresses.

```
New-UMIPGateway -Identity MyUMIPGateway -Address
fe80::39bd:88f7:6969:d223%11 -IPAddressFamily Any -Status
Disabled -OutcallsAllowed $false
```

Manage a UM IP gateway

Connect your voice mail system to your telephone network > UM IP gateways > UM IP gateway procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

After you create a Unified Messaging (UM) IP gateway, you can view or configure a variety of

settings. For example, you can configure the IP address or a fully qualified domain name (FQDN), configure outgoing call settings, and enable or disable Message Waiting Indicator.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to view or configure UM IP gateway properties

1. In the EAC, navigate to **Unified Messaging** > **UM IP Gateways**. In the list view, select the UM IP gateway that you want to manage, and then click **Edit** .
2. Use the **UM IP Gateway** page to view and configure settings for the UM IP gateway. You can view or configure the following settings:
   - **Status** This display-only field shows the status of the UM IP gateway.
   - **Name** Use this box to specify a unique name for the UM IP gateway. This is a display name that appears in the EAC. If you have to change the display name of the UM IP gateway after it's been created, you must first delete the existing UM IP gateway, and then create another UM IP gateway that has the appropriate name. The UM IP gateway name is required, but it's used for display purposes only. Because your organization may use multiple UM IP gateways, we recommend that you use meaningful names for your UM IP gateways. The maximum length of a UM IP gateway name is 64 characters, and it can include spaces.
   - **Address** You can configure a UM IP gateway with either an IP address or a fully qualified domain name (FQDN). Use this box to specify the IP address or FQDN configured on the VoIP gateway, SIP-enabled PBX, IP PBX, or SBC.

You can enter alphabetical and numeric characters in this box. IPv4 addresses, IPv6 addresses, and FQDNs are supported. If you use an FQDN, you must also make sure that you have correctly configured a DNS host record for the VoIP gateway so that the host name will be correctly resolved to an IP address. Also, if you use an FQDN instead of an IP address, and the DNS configuration for the UM IP gateway is changed, you must disable and then enable the UM IP gateway to make sure that configuration information for the UM IP gateway is updated correctly.

If you want to use mutual Transport Layer Security (mutual TLS) between a UM IP gateway and a dial plan operating in either SIP secured or Secured mode, you must configure the UM IP gateway with an FQDN. You must also configure it to listen on port 5061 and verify that any IP gateways or IP PBXs have also been configured to listen for mutual TLS requests on port 5061. To configure a UM IP gateway, run the following command: `Set-UMIPGateway -identity MyUMIPGateway -Port 5061`.

- **Allow outgoing calls through this UM IP gateway** Select this check box to allow the UM IP gateway to accept and process outgoing calls. This setting doesn't affect call transfers or incoming calls from a VoIP gateway.

By default, when the UM IP gateway is created, this setting is enabled. If you disable this setting, users associated with the dial plan won't be able to make outgoing calls through the VoIP gateway, IP PBX, or SBC defined in the **Address** field.

- **Allow message waiting indicator** Select this check box to allow voice mail notifications to be sent to users for calls taken by the UM IP gateway. This setting allows the UM IP gateway to receive and send SIP NOTIFY messages for users. This setting is enabled by default and allows message waiting notifications to be sent to users.

Message Waiting Indicator can refer to any mechanism that indicates the existence of a new or unheard message. The indication that a new voice message has arrived can be found in the Inbox in clients such as Outlook and Outlook Web App. It can take the form of a Short Messaging Service (SMS) or text message sent to a registered mobile phone, an outbound call made from an Exchange server to a preconfigured number, or a lighted desktop phone lamp for a user.

Use the Shell to configure UM IP gateway properties

This example modifies the IP address of a UM IP gateway named `MyUMIPGateway`.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address 10.10.10.1
```

This example prevents the UM IP gateway named `MyUMIPGateway` from accepting incoming calls and prevents outgoing calls.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
voipgateway.contoso.com -Status 2 -OutcallsAllowed $false
```

This example enables the UM IP gateway to function as a VoIP gateway simulator and can be used with the **Test-UMConnectivity** cmdlet.

```
Set-UMIPGateway -Identity MyUMIPGateway -Simulator $true
```

Important:
There is a period of latency before all changes that you make to the configuration of a UM IP gateway replicate to all Exchange servers in the same UM dial plan as the UM IP gateway.

This example prevents the UM IP gateway named `MyUMIPGateway` from accepting incoming calls and prevents outgoing calls, sets an IPv6 address, and allows the UM IP gateway to use IPv4 and IPV6 addresses.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
fe80::39bd:88f7:6969:d223%11 -IPAddressFamily Any -Status
Disabled -OutcallsAllowed $false
```

Use the Shell to view UM IP gateway properties

This example displays a formatted list of all the UM IP gateways in the Active Directory forest.

```
Get-UMIPGateway |Format-List
```

This example displays the properties for a UM IP gateway named `MyUMIPGateway`.

```
Get-UMIPGateway -Identity MyUMIPGateway
```

This example displays all the UM IP gateways including VoIP gateway simulators in the Active Directory forest.

```
Get-UMIPGateway -IncludeSimulator $true
```

Enable a UM IP gateway

Connect your voice mail system to your telephone network > UM IP gateways > UM IP gateway procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

By default, when a Unified Messaging (UM) IP gateway is created, its status is set to enabled. However, you might need to disable the UM IP gateway to take it offline and not allow it to take incoming or outgoing calls. After you create a UM IP gateway, you can control its operation and functionality by setting its status variable to enabled or disabled.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created and has been disabled. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable a UM IP gateway

1. In the EAC, navigate to > **Unified Messaging** > **UM IP Gateways**, select the UM IP gateway you want to enable, and then click the **Up arrow** ↑.
2. On the **Warning** page, click **Yes**.

Use the Shell to enable a UM IP gateway

This example enables a UM IP gateway named `MyUMIPGateway`.

```
Enable-UMIPGateway -Identity MyUMIPGateway
```

Disable a UM IP gateway

Connect your voice mail system to your telephone network > UM IP gateways > UM IP gateway procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-13*

By default, when you create a Unified Messaging (UM) IP gateway, the status of the UM IP gateway is enabled. After the UM IP gateway is created, you can disable the operation of the gateway by

setting its status to disabled. After you disable the UM IP gateway, the Voice over IP (VoIP) gateway, IP Private Branch eXchange (PBX), or session border controller (SBC) that it's configured to use can no longer process incoming Unified Messaging calls.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created and is enabled. For detailed steps, see Create a UM IP gateway and Enable a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to disable a UM IP gateway

1. In the EAC, navigate to **Unified Messaging** > **UM IP Gateways**, select the UM IP gateway you want to disable, and then click the **Down arrow** ↓.
2. On the **Warning** page, click **Yes**.

Use the Shell to disable a UM IP gateway

This example disables a UM IP gateway named `MyUMIPGateway` and stops it from accepting incoming calls from a VoIP gateway, IP PBX, or SBC.

```
Disable-UMIPGateway -Identity MyUMIPGateway
```

This example disables a UM IP gateway named `MyUMIPGateway` and disconnects all current calls immediately.

```
Disable-UMIPGateway -Identity MyUMIPGateway -Immediate
$true
```

Configure a fully qualified domain name

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-09*

You can configure a Unified Messaging (UM) IP gateway with either an IP address or a fully qualified domain name (FQDN). When you create a UM IP gateway, you must define the IP address or the FQDN configured on the VoIP gateway, IP PBX, or session border controller (SBC) that you're using. You can change the IP address or FQDN after the UM IP gateway is created.

If you create a UM IP gateway using an FQDN, you must create the appropriate HOST (A) records in your DNS forward lookup zone. If you create a UM IP gateway using an FQDN, and the DNS configuration for the UM IP gateway is changed, you must disable and then enable the UM IP gateway to make sure that its configuration information is updated correctly.

If you want to use mutual Transport Layer Security (mutual TLS) between a UM IP gateway and a dial plan operating in either SIP secured or Secured mode, you must configure the UM IP gateway with an FQDN. You must also configure it to listen on port 5061 and verify that the VoIP gateway, IP PBX, or SBC has also been configured to listen for mutual TLS requests on port 5061. To configure a UM IP gateway, run the following command: `Set-UMIPGateway -identity MyUMIPGateway -Port 5061`.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure an FQDN

1. In the EAC, navigate to **Unified Messaging** > **UM IP Gateways**, select the UM IP gateway that you want to modify, and then click **Edit** .
2. On the **UM IP gateway** page, in **Address**, enter the FQDN for the VoIP gateway, PBX enabled for SIP, IP PBX, or SBC.
3. Click **Save**.

Important:
When you use an FQDN instead of an IP address on the UM IP gateway, verify that the correct DNS records have been created.

Use the Shell to configure an FQDN

This example configures a UM IP gateway named `MyUMIPGateway` with an FQDN named voipgateway.contoso.com.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
voipgateway.contoso.com
```

This example configures a UM IP gateway named `MySBC` with an FQDN of sbc.contoso.com and listens for SIP requests on TCP port 5061.

```
Set-UMIPGateway -Identity MySBC -Address sbc.contoso.com -
Port 5061
```

Configure the IP address

Connect your voice mail system to your telephone network > UM IP gateways > UM IP gateway procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-11*

Before you create a Unified Messaging (UM) IP gateway, you must first set the IP address or the fully qualified domain name (FQDN) on the VoIP gateway, IP PBX, or session border controller (SBC) that you're using. Then, when you create the UM IP gateway, you set the IP address or FQDN. You can change the IP address or FQDN later.

You can configure the IP address or FQDN using either the EAC or the Shell. In the EAC, the **Address**

box on the **UM IP gateway** page can accept an IPv4 IP address, an IPv6 address, or an FQDN. You can also use the *Address* parameter on the **Set-UMIPGateway** cmdlet in the Shell to set an IPv4 IP address, an IPv6 address, or an FQDN. If you create a UM IP gateway using an FQDN, you must create the appropriate HOST A records in your DNS forward lookup zone. If the DNS configuration for the UM IP gateway is changed, you must disable and then enable the UM IP gateway to make sure that its configuration information is updated correctly.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the IP address on a UM IP gateway

1. In the EAC, navigate to **Unified Messaging** > **UM IP Gateways**, select the UM IP gateway that you want to modify, and then click **Edit** .
2. On the **UM IP gateway** page, in the **Address** box, enter the IP address for the VoIP gateway, IP PBX, or session border controller (SBC).
3. Click **Save** to save your changes.

Important:

If you use an FQDN instead of an IP address on the UM IP gateway, verify that the correct DNS records have been created.

Use the Shell to configure the IP address on a UM IP

gateway

This example configures a UM IP gateway named `MyUMIPGateway` with an IP address of 10.10.10.1.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address 10.10.10.1
```

This example configures a UM IP gateway named `MyUMIPGateway` with an IP address of 10.10.10.10 and listens for SIP requests on TCP port 5061.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
10.10.10.10 -Port 5061
```

This example prevents the UM IP gateway named `MyUMIPGateway` from accepting incoming and outgoing calls, sets an IPv6 address, and allows the UM IP gateway to use IPv4 and IPV6 addresses.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
fe80::39bd:88f7:6969:d223%11 -IPAddressFamily Any -Status
Disabled -OutcallsAllowed $false
```

Configure the listening port

Connect your voice mail system to your telephone network > UM IP gateways > UM IP gateway procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can configure the TCP port that's used to listen for Session Initiation Protocol (SIP) requests on a Unified Messaging (UM) IP gateway. By default, when you create a UM IP gateway, the TCP SIP listening port number is set to 5060. The TCP SIP listening port can't be configured or changed by using the EAC. You must configure the TCP SIP listening port number by using the **Set-UMIPGateway** cmdlet.

You may have to configure the TCP listening port number to 5061 if you want to:

- Set the VoIP security setting on a UM dial plan to SIP Secured.
- Set the VoIP security setting on a UM dial plan to Secured.
- Integrate with Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server.
- Use mutual Transport Layer Security (mutual TLS) to encrypt network data between Exchange servers and a VoIP gateway, Private Branch eXchange (PBX) enabled for SIP, IP PBX, or session border controller (SBC).

If you want to use mutual TLS between a UM IP gateway and a dial plan operating in either SIP

Secured or Secured mode, when you create the UM IP gateway you must configure it with a fully qualified domain name (FQDN) and then use the Shell to configure the UM IP gateway to listen on TCP port 5061. You must also verify that any VoIP gateways, PBXs enabled for SIP, IP PBXs, and SBCs have also been configured to listen for mutual TLS requests on port 5061.

Important:

When you create a UM IP gateway using an FQDN, you must create the appropriate HOST (A) records in your DNS forward lookup zone. If you create a UM IP gateway using an FQDN, and the DNS configuration for the UM IP gateway is changed, you must disable and then enable the UM IP gateway to make sure that the UM IP gateway's configuration information is updated correctly.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to configure the TCP listening port

This example configures a UM IP gateway named `MyUMIPGateway` that has an FQDN of mTLS.MyUMIPGateway.contoso.com and listens for SIP requests on TCP port 5061.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
mTLS.MYUMIPGateway.contoso.com -Port 5061
```

This example configures a UM IP gateway named `MyUMIPGateway` that has an FQDN of SIPSecured.MyUMIPGateway.contoso.com and listens for SIP requests on TCP port 5061.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
SIPSecured.MyUMIPGateway.contoso.com -Port 5061
```

This example configures a UM IP gateway named `MyUMIPGateway` that has an FQDN of MyOCSUMIPGateway.contoso.com and listens for SIP requests on TCP port 5061.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
MyOCSUMIPGateway.contoso.com -Port 5061
```

Delete a UM IP gateway

Connect your voice mail system to your telephone network > UM IP gateways > UM IP gateway procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

When you delete a Unified Messaging (UM) IP gateway, Exchange servers can no longer accept incoming calls from the Voice over IP (VoIP) gateway, Session Initiation Protocol (SIP)–enabled Private Branch eXchange (PBX), IP PBX, or session border controller (SBC) associated with the UM IP gateway.

Important:
You should delete a UM IP gateway only when you fully understand the implications of disabling communication with a VoIP gateway, IP PBX, or SBC.

For additional tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to delete a UM IP gateway

1. In the EAC, navigate to **Unified Messaging** > **UM IP Gateways**, select the UM IP gateway you want to delete, and then click **Delete** 🗑.
2. On the **Warning** page, click **Yes**.

Use the Shell to delete a UM IP gateway

This example deletes the UM IP gateway named `MyUMIPGateway`.

```
Remove-UMIPGateway -Identity MyUMIPGateway
```

UM hunt groups

Exchange Server 2013 > Unified Messaging > Connect your voice mail system to your telephone network >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-18*

A telephony hunt group provides a way to distribute telephone calls from a single number to multiple extensions or telephone numbers. In Unified Messaging (UM), a UM hunt group is a logical representation of a telephony hunt group, and it links a UM IP gateway to a UM dial plan.

Looking for management tasks related to Unified Messaging hunt groups? See UM hunt group procedures.

Contents

What is a hunt group?

Hunt group is a term used to describe a group of Private Branch eXchange (PBX) or IP PBX extension numbers that are shared by users. Hunt groups are used to efficiently distribute calls into or out of a specific business unit. Creating and defining a hunt group minimizes the chance that a caller who places an incoming call will receive a busy signal when the call is received.

Hunt groups are used to locate an open line, extension, or channel when an incoming call is received. Calls are "rolled over" to the next available line when a primary phone line is busy or isn't answered. The calling party gets a busy signal or is sent to voice mail only if no extensions in the group are open. For example, a PBX or IP PBX might be configured to have 10 extension numbers for the sales department. The 10 sales extension numbers would be configured as one hunt group.

The settings for a simple hunt group include a name, an extension number, a list of available group members, and a hunt group selection method. The hunt group selection method determines the order in which incoming calls are presented to the members of the hunt group.

There are multiple algorithms or methods that a PBX or IP PBX can use to locate an open line, extension, or channel. These include:

- **Group hunt or ring all extensions** When an incoming call is received on the hunt group extension number, the PBX or IP PBX rings all extension numbers in the group.
- **Start with lowest number or linear hunting** This is the default setting on most PBXs and IP PBXs. With this method, calls are routed to the first idle line in sequential order, starting with the first line in the group. This configuration is most often found on multiline phones at small businesses.
- **Round-robin or circular hunting** With this method, calls are routed to the first idle line, starting with the line after the one that last handled a call. When calls are distributed using the "round-robin" method, if a call is delivered to line 1, the next call goes to line 2, the next to line 3, and so on. This process continues even if one of the previous lines becomes free. When the end of the hunt group is reached, the hunting starts over at the first line. Lines are skipped only if they are still busy on a previous call. Circular or round-robin hunting spreads call disruption evenly throughout all the calls, minimizing the possibility for a major disruption in service.
- **Most-idle or uniform-distribution hunting** With this method, the call is routed to the first available line in the group that has been idle the longest. This method uses the length of time that the person taking the call has been busy instead of whether the line is available. This method is typically used in large call centers where the incoming calls are being answered by people and the load is distributed evenly across the group of extension numbers.

You can configure one or more hunt groups. Each hunt group must include a minimum of two lines. If a number is already being used in one hunt group, it won't be available in another.

Following are examples of simple telephony hunt groups and how they work.

Example 1

Extension 300 (pilot number) is programmed so that when a call comes in, it rings extension 301, then 302, then 303, then 304.

1. Extension 301 is busy.
2. Extension 302 rings and isn't answered.
3. Extension 303 answers the call.
4. Extension 304 is free and waiting for an incoming call.

Example 2

Extension 1000 (pilot number) is programmed so that when a call comes in, it rings all the extensions 2000 through 2003 at the same time:

1. Extension 2000 is free.
2. Extension 2001 is free.
3. Extension 2002 is free.
4. Extension 2003 answers the incoming call.

Return to top

What is a pilot number?

In a telephony network, a PBX or an IP PBX can be configured to have a single hunt group or multiple hunt groups. Each hunt group created on a PBX or IP PBX must have an associated *pilot number*. Using a pilot number helps to eliminate busy signals and to route incoming calls to the extension numbers that are available. The PBX or IP PBX uses the pilot number to locate the hunt group and in turn to locate the telephone extension number on which the incoming call was received and the extensions that are assigned to the hunt group. Without a defined pilot number, the PBX or IP PBX can't locate where the incoming call was received.

A pilot number is the address, extension, or location of the hunt group inside the PBX or IP PBX. It's generally a blank extension number or one extension number from a hunt group of extension numbers that doesn't have a person or telephone associated with it. For example, you might configure a hunt group on a PBX or IP PBX to contain extension numbers 4100, 4101, 4102, 4103, 4104, and 4105. The pilot number for the hunt group is configured as extension 4100. When a call is received on extension number 4100, the PBX or IP PBX looks for the next available extension number to determine where to deliver the call. In this example, the PBX or IP PBX will use its programmed search algorithm to look at extension numbers 4101, 4102, 4103, 4104, and 4105.

Using a pilot number helps eliminate busy signals and helps route incoming calls to the extension numbers that are available. In Unified Messaging, the PBX or IP PBX pilot number is used as the target. If none of the extension numbers in the hunt group answer an incoming call, the call is routed to a Mailbox server running the Microsoft Exchange Unified Messaging service.

Return to top

What is a UM hunt group?

Unified Messaging hunt groups are critical to the operation of the UM system. A UM hunt group is a logical representation of an existing PBX or IP PBX hunt group. It's used to link a UM IP gateway with a UM dial plan. A single UM hunt group can also link multiple UM IP gateways with a UM dial plan. By default, when you create a UM IP gateway and associate it with a UM dial plan, a UM hunt group is created, and you can also create other hunt groups. You must create at least one UM hunt group.

UM hunt groups are used to locate the PBX or IP PBX hunt group from which an incoming call is

received. A pilot number defined for a hunt group on the PBX or IP PBX must also be defined for the UM hunt group. The pilot number is used to match the information presented for incoming calls using the Session Initiation Protocol (SIP) signaling information on the voice message. The pilot number enables Exchange servers to interpret the call together with the correct dial plan so that the call can be routed correctly. The absence of a hunt group prevents Exchange servers from knowing the location of the incoming call. Knowing the location of incoming calls enables the Exchange servers to accept the call header information that's passed from the VoIP gateway, IP PBX, or SIP-enabled PBX. It's very important that you configure your UM hunt groups correctly, because incoming calls that don't match the pilot number defined on the UM hunt group won't be answered, and routing of incoming calls will fail.

In on-premises and hybrid deployments when you create a UM hunt group, you're enabling all Client Access and Mailbox servers, regardless of whether they've been added to a UM dial plan, to communicate with a VoIP gateway, IP PBX, or SIP-enabled PBX. This is because all Client Access and Mailbox servers answer incoming calls for all dial plans, instead of for a specific UM dial plan like the UM server did in previous versions of Exchange. If you delete the UM hunt group, the associated UM IP gateway won't be able to answer incoming calls from a VoIP gateway, IP PBX, or SIP-enabled PBX or place outgoing calls through the VoIP gateway, IP PBX or SIP-enabled PBX using the specified pilot number.

However, for on-premises and hybrid deployments if you're integrating UM with Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server, you must add all Client Access and Mailbox servers to all SIP URI dial plans that have been created to work with Communications Server 2007 R2 or Lync Server. This enables call routing and outdialing to work correctly.

For more information about UM IP gateways, see UM IP gateways.

UM hunt group procedures

Unified Messaging > Connect your voice mail system to your telephone network > UM hunt groups >

Applies to: *Exchange Online*

Topic Last Modified: *2012-11-05*

Create a UM hunt group

View a UM hunt group

Delete a UM hunt group

Create a UM hunt group

Connect your voice mail system to your telephone network > UM hunt groups > UM hunt group procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-16*

A Unified Messaging (UM) hunt group is a logical representation of a Private Branch eXchange (PBX) or IP PBX hunt group. A UM hunt group acts as a connection or link between a UM IP gateway and a UM dial plan.

Note:
If you associate a UM dial plan with the UM IP gateway when you create a UM IP gateway, a UM hunt group will also be created.

Note:
If you want to change the settings for a UM hunt group, you must delete the hunt group and then create another hunt group that has the appropriate settings.

For additional management tasks related to UM hunt groups, see UM hunt group procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM hunt groups" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to create a UM hunt group

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial

plan you want to modify, and then click **Edit** ✎.

2. On the **UM Dial Plan** page, under **UM Hunt Groups**, click **New +**.
3. On the **New UM hunt group** page, enter the following information:
   - **Name** Use this box to create the display name for the UM hunt group. A UM hunt group name is required and must be unique, but it's used only for display purposes in the EAC and the Shell. If you have to change the display name of the hunt group after it's been created, you must first delete the existing hunt group and then create another hunt group that has the appropriate name.

If your organization uses multiple hunt groups, we recommend that you use meaningful names for your hunt groups. The maximum length of a UM hunt group name is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

   - **UM IP gateway** Use this box to specify the UM IP gateway to be used. Click **Browse** to select the UM IP gateway, and then click **OK**.
   - **Pilot identifier** Use this box to specify a string that uniquely identifies the pilot identifier configured on the PBX or IP PBX.

An extension number or a Session Initiation Protocol (SIP) Uniform Resource Identifier (URI) can be used in this box. Alphanumeric characters are accepted in this box. For legacy PBXs, a numeric value is used as a pilot identifier. However, some IP PBXs can use SIP URIs.

4. Click **Save**.

Use the Shell to create a UM hunt group

This example creates a UM hunt group named `MyUMHuntGroup` that has a pilot identifier of 12345.

```
New-UMHuntGroup -Name MyUMHuntGroup -PilotIdentifier 12345
-UMDialplan MyUMDialPlan -UMIPGateway MyUMIPGateway
```

This example creates a UM hunt group named `MyUMHuntGroup` that has multiple pilot identifiers.

```
New-UMHuntGroup -Name MyUMHuntGroup -PilotIdentifier
5551234,55555 -UMDialplan MyUMDialPlan -UMIPGateway
MyUMIPGateway
```

View a UM hunt group

Connect your voice mail system to your telephone network > UM hunt groups > UM hunt group procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

When you view the properties for a Unified Messaging (UM) hunt group, you can view the properties associated with a single UM hunt group or with all UM hunt groups associated with a single UM IP gateway. If neither parameter is specified, all UM hunt groups will be returned. You can't use the EAC to view UM hunt group properties; you must use the Shell.

After a UM hunt group has been created, the configured settings can't be changed. If you want to change a configuration setting such as the pilot identifier on a UM hunt group, you must delete the existing UM hunt group and create a new UM hunt group that has the correct settings.

For additional tasks related to UM hunt groups, see UM hunt group procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM hunt groups" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM gateway has been created. For detailed steps, see Create a UM IP gateway.
- Before you perform this procedure, confirm that a UM hunt group has been created. For detailed steps, see Create a UM hunt group.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to view the properties of a UM hunt group

This example displays all the UM hunt groups in the Active Directory forest.

```
Get-UMHuntGroup
```

This example displays the details of a UM hunt group named `MyUMHuntGroup` in a formatted list.

```
Get-UMHuntGroup -identity MyUMIPGateway\MyUMHuntGroup |
Format-List
```

Note:

When you're using the **Get-UMHuntGroup** cmdlet, you can't enter only the name of the UM hunt group. You must also include the name of the UM IP gateway that's associated with the UM hunt group.

Delete a UM hunt group

Connect your voice mail system to your telephone network > UM hunt groups > UM hunt group procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

After you delete a Unified Messaging (UM) hunt group, the UM IP gateway associated with the UM hunt group will no longer service or answer incoming calls. If deleting the UM hunt group leaves the UM IP gateway without any remaining configured hunt groups, the UM IP gateway can't handle or process UM calls.

For additional tasks related to UM hunt groups, see UM hunt group procedures.

Warning:

If you want to change the UM hunt group settings, you must delete the hunt group and then create another hunt group that has the appropriate settings.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM hunt groups" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- Before you perform these procedures, confirm that a UM hunt group has been created. For detailed steps, see Create a UM hunt group.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to delete a UM hunt group

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, click the UM dial

plan you want to change, and on the toolbar, click **Edit** .

2. On the **UM Dial Plan** page, under **UM Hunt Groups**, select the hunt group you want to delete, and on the toolbar, click **Delete** .
3. On the **Warning** page, click **Yes**.

Use the Shell to delete a UM hunt group

This example deletes a UM hunt group named `MyUMHuntGroup`.

```
Remove-UMHuntGroup -identity MyUMHuntGroup
```

UM services

Exchange Server 2013 > Unified Messaging > Connect your voice mail system to your telephone network >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-18*

Client Access servers running the Microsoft Exchange Unified Messaging Call Router service and Mailbox servers running the Microsoft Exchange Unified Messaging service enable you to deploy Unified Messaging (UM) and voice mail functionality for users in your organization.

Looking for management tasks related to UM services? See UM services procedures.

Contents

Client Access and Mailbox servers

In Exchange 2013, the server roles found in Exchange 2007 and Exchange 2010 are combined into two types of servers, and all components or services from those server roles are run on the same physical server or on two separate servers called Client Access and Mailbox.

In this new model, the Client Access server is responsible for Autodiscover, Secure Sockets Layer (SSL), authentication, redirection, and proxying. The Client Access server is the entry point for any inbound calls or Session Initiation Protocol (SIP) requests for Unified Messaging. The routing logic and SIP REDIRECT is implemented as a service that's automatically included on a Client Access server. This service is known as the Microsoft Exchange Unified Messaging Call Router service. It runs on each Client Access server in your organization.

When a Client Access server receives a SIP INVITE for an incoming call, the Microsoft Exchange Unified Messaging Call Router service redirects the incoming call to the Mailbox server. Then a media channel (RTP or SRTP) is created between the VoIP gateway, IP PBX, or session border controller (SBC) and the Mailbox server that hosts the user's mailbox. Although the Client Access server acts as a SIP redirector, it only handles SIP requests from VoIP gateways, IP PBXs, or SBCs. It doesn't receive any media traffic. Media traffic that uses RTP or SRTP is only passed between the Mailbox server and SIP peers such as VoIP gateways, IP PBXs, or SBCs. When you deploy Exchange 2013 and UM, you have to configure your VoIP gateways, IP PBXs, or SBCs to point to the Client Access servers that you've installed so that incoming calls will be routed correctly for UM.

In Exchange 2013, the Mailbox server handles the same processes as the Unified Messaging server role handled in Exchange 2007 and Exchange 2010. The Mailbox server runs both the Microsoft Exchange Unified Messaging service and UM worker processes.

When you're installing your Client Access and Mailbox servers and deploying Unified Messaging, you don't have to associate or add Client Access or Mailbox servers to UM dial plans. Client Access and Mailbox servers answer all incoming calls and then use the UM dial plans to locate users.

However, if you're integrating Unified Messaging with Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server, both the SIP and the RTP or SRTP media channels for incoming calls are handled by Lync servers and the Mailbox server. In a Lync integrated environment, you don't have VoIP gateways, IP PBXs, or SBCs. To Lync, the Mailbox server that's running the Microsoft Exchange Unified Messaging service looks just like an Exchange 2010 UM server. The Mailbox server and the Client Access server are considered trusted peers because both servers must be added to a SIP dial plan. Lync routes the incoming call using the Inbound Routing component, which uses SIP to communicate with the Client Access server and then route the call to a Mailbox server.

Return to top

Server configuration settings

In Exchange 2013, all the UM components and configuration settings that applied to a single computer running the Unified Messaging server role in Exchange 2010 are still available. However, some of those components and configuration settings are found on a Client Access server and others are available on a Mailbox server. In some cases, the same setting is available on both. The following list shows the parameters and settings that are available on both a Client Access server and a Mailbox server.

- [-DialPlans <MultiValuedProperty>]
- [-MaxCallsAllowed <Int32>]
- [-SipTcpListeningPort <Int32>]
- [-SipTlsListeningPort <Int32>]
- [-Status <Enabled | Disabled | NoNewCalls>]
- [-UMStartupMode <TCP | TLS | Dual>]

For the Mailbox server, you'll use the **Set-UMService**, **Get-UMService**, **Enable-UMService**, and

Disable-UMService cmdlets to view or configure UM properties for the Microsoft Exchange Unified Messaging service. A different set of cmdlets, **Set-UMCallRouterSettings** and **Get-UMCallRouterSettings**, are used to view or configure the Microsoft Exchange Unified Messaging Call Router service properties on a Client Access server. This ensures that the existing **Get-UMServer**, **Set-UMServer**, **Enable-UMServer**, and **Disable-UMServer** cmdlets from Exchange 2007 and Exchange 2010 will work in a coexistence deployment with Exchange 2013 Mailbox servers. This also ensures that the cmdlets will work when the Mailbox and Client Access servers are installed on the same or different computers.

Return to top

Server operation

When Client Access and Mailbox servers are installed, they're automatically enabled so they can answer incoming and outgoing voice calls and route voice mail messages to the intended recipients in your Exchange organization.

You can allow the Microsoft Exchange Unified Messaging service on a Mailbox server or the Microsoft Exchange Unified Messaging Call Router service on a Client Access server to answer new calls, or prevent it from doing so. By default, a Mailbox or Client Access server is in an enabled state after it's installed. When you're setting the Mailbox or Client Access server to accept incoming voice, fax, auto attendant, and Outlook Voice Access calls, you use the **Set-ServerComponentState** cmdlet.

Configuring Maintenance Mode for an Exchange 2013 Mailbox or Client Access server lets you take the server out of service. For a Mailbox server, out-of-service means that the server won't host any active databases, all transport queues are empty, and the server won't accept any incoming calls from Client Access servers, VoIP gateways, IP PBXs, SIP-enabled PBXs, or SBCs. For a Client Access server, out-of-service means that the server won't accept any incoming calls from VoIP gateways, IP PBXs, SIP-enabled PBXs or SBCs.

In Exchange 2007 and Exchange 2010, there was a status parameter that could be used to control the operational status of a Unified Messaging server. In Exchange 2013, no status parameter is available for that purpose on the **Set-UMService** cmdlet for a Mailbox server or the **Set-UMCallRouterSettings** cmdlet on a Client Access server.

Although the Client Access and Mailbox servers are set to enabled when they're installed, neither server can correctly process and route incoming calls to UM-enabled users until a UM dial plan is linked with at least one UM IP gateway.

After a dial plan is linked with a UM IP gateway, the Client Access and Mailbox servers locate all UM IP gateways that are associated with the UM dial plan and VoIP gateways, IP PBXs, and SBCs. To detect and identify any configuration changes on either UM dial plans or UM IP gateways, the Client Access or Mailbox servers check the configuration every 10 minutes.

If the UM IP gateway identifies any changes to the configuration, the Client Access or Mailbox

server reacts accordingly, and either starts using or stops using the appropriate VoIP gateway, IP PBX, or SBC. After the Client Access and Mailbox servers answer incoming calls for users linked with a UM dial plan and they're correctly communicating with VoIP gateways, IP PBXs, and SBCs, you can run a set of diagnostic operations to verify that they're operating correctly and that connectivity between the Exchange servers and VoIP gateways, IP PBXs, or SBCs is working correctly.

Return to top

UM protocols, ports, and services

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-04-25*

Microsoft Exchange 2013 Unified Messaging (UM) requires that several TCP and User Datagram Protocol (UDP) ports be used to establish communication between servers running Exchange 2013 and other devices. By allowing access through these IP ports, you enable Unified Messaging to function correctly. This topic discusses the TCP and UDP ports used in Exchange 2013 Unified Messaging.

Unified Messaging protocols and services

Exchange 2013 Unified Messaging features and services rely on static and dynamic TCP and UDP ports to ensure correct operation of Client Access servers running the Microsoft Exchange Unified Messaging Call Router service and Mailbox servers running the Microsoft Exchange Unified Messaging service.When Exchange 2013 is installed, static inbound Windows Firewall rules are added for Exchange. If you change the TCP ports that are used by Client Access and Mailbox servers, you may also need to reconfigure the Windows Firewall rules to allow Unified Messaging to work correctly.

Important:
On Exchange 2013 Client Access and Mailbox servers running UM components and services, Exchange setup creates inbound firewall rules that allow inbound communication without any TCP port restrictions. The following inbound rules for UM services are added:

1. **SESWorker (GFW) (TCP-In)**
2. **UMCallRouter (GFW) (TCP-In)**
3. **UMCallRouter (TCP-In)**
4. **UMService (GFW) (TCP-In)**
5. **UMService (TCP-In)**
6. **UMWorkerProcess – RPC (TCP-In)**

7. UMWorkerProcess (GFW) (TCP-In)
8. UMWorkerProcess (TCP-In)

Session Initiation Protocol

Session Initiation Protocol (SIP) is a protocol used for initiating, modifying, and ending an interactive user session that involves multimedia elements such as video, voice, instant messaging, online games, and virtual reality. It's one of the leading signaling protocols for Voice over IP (VoIP), together with H.323. Most VoIP standards-based solutions use either H.323 or SIP. However, several proprietary designs and protocols also exist. The VoIP protocols typically support features such as call waiting, conference calling, and call transfer.

SIP clients such as VoIP gateways and IP Private Branch eXchanges (PBXs) can use TCP and UDP port 5060 to connect to SIP servers, including Client Access servers running the Microsoft Exchange Unified Messaging Call Router service. SIP is used only for setting up and tearing down voice or video calls. All voice and video communications occur over Realtime Transport Protocol (RTP).

Real-Time Transport Protocol

RTP defines a standard packet format for delivering audio and video over a specific network, such as the Internet. RTP carries only voice/video data over the network. Call setup and teardown are generally performed by SIP.

RTP doesn't require a standard or static TCP or UDP port to communicate with. RTP communications occur on an even number UDP port, and the next higher odd number port is used for TCP communications. Although there are no standard port range assignments, RTP is generally configured to use ports between 1024 and 65535, and Mailbox servers running the Microsoft Exchange Unified Messaging service follow this convention. It's difficult for RTP to traverse firewalls because it uses a dynamic port range.

Unified Messaging Web services

The Unified Messaging Web services installed on Mailbox servers use IP for network communication between a client, the Mailbox server, and computers running other Exchange 2013 server roles. Several Exchange 2013 Outlook Web App and Outlook 2013 client features rely on Unified Messaging Web services to operate correctly.

The following Unified Messaging client features rely on Unified Messaging Web services:

- Voice mail options available with Exchange 2013 Outlook Web App, including the Play on Phone feature and the ability to reset a PIN.
- The Play on Phone feature found in an Outlook client.

UM ports

The Microsoft Exchange Unified Messaging Call Router service found on a Client Access server uses SIP over either Transmission Control Protocol (TCP) or mutual Transport Layer Security (mutual TLS) to communicate with Mailbox servers that are running the Microsoft Exchange Unified Messaging service. To avoid TCP/User Datagram Protocol (UDP) port conflicts, the Microsoft Exchange Unified Messaging Call Router service and Microsoft Exchange Unified Messaging service default to and listen on different TCP ports. They can accept both unsecured and secured connections, depending on whether mutual TLS is used with SIP and RTP traffic. By default, a Client Access server listens for SIP requests on both TCP port 5060 in Unsecured mode and TCP port 5061 in SIP Secured mode when mutual TLS is used. These ports are configurable using the **Set-UMCallRouterSettings** cmdlet. The Microsoft Exchange Unified Messaging Call Router service on the Client Access server doesn't handle media (RTP or SRTP) traffic, so only TCP ports and no UDP ports are used. By default, a Mailbox server listens for SIP requests on both TCP port 5062 in Unsecured mode and TCP port 5063 in SIP Secured mode when mutual TLS is used. These ports aren't configurable using Exchange Management Shell cmdlets. The Microsoft Exchange Unified Messaging service on the Mailbox server will accept connections from a Client Access server on SIP ports 5062 and 5063. After the Client Access server redirects the SIP request to a Mailbox server, an RTP or SRTP media channel is created using a VoIP gateway, IP PBX, or SBC, and the Microsoft Exchange Unified Messaging worker process on the Mailbox server.

The following table summarizes the Exchange 2013 ports and protocols, and whether the ports can be changed.

UM listening ports

Protocol	TCP port	UDP port	Can the ports be changed?
SIP (Client Access server – Microsoft Unified Messaging Call Router service)	5060 (unsecured), 5061 (secured). The service listens on both ports.	Not applicable	Yes, using the **Set-UMCallRouterSettings** cmdlet.
SIP (Mailbox server – Microsoft Exchange Unified Messaging service)	5062 (unsecured), 5063 (secured). The service listens on both ports.	Not applicable	Ports can't be changed.
SIP (Mailbox server - UM worker process)	5065 and 5067 for TCP (unsecured). 5066 and 5068 for mutual TLS (secured).	Not applicable	Ports can't be changed.

RTP (Mailbox server - UM worker process)	Not applicable	Ports between 1024 and 65535.	The range of ports can be changed through the registry (however, this isn't a supported configuration): HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft Speech Server\2.0\AudioConnectionMinPort HKLM\SOFTWARE\Microsoft\Microsoft Speech Server\2.0\AudioConnectionMaxPort

Lync Server and UM ports

Exchange 2013 Unified Messaging supports Network Address Translation (NAT) traversal and allows for the RTP media to a Mailbox server to be tunneled through a NAT firewall. However, for this to work, you must also have Microsoft Office Communications Server 2007 R2 and Microsoft Lync Server 2010 or Microsoft Lync 2013 deployed in your environment. If you deploy both Exchange 2013 and Communications Server 2007 R2 or Microsoft Lync Server 2010 or Lync 2013 on your network, this deployment will enable Mailbox servers running the Microsoft Exchange Unified Messaging service to communicate with endpoints outside a NAT firewall. The Mailbox server is associated with a Communications Server 2007 R2, Microsoft Lync Server 2010, or a Lync 2013 pool and obtains the appropriate authentication tokens from the A/V Authentication service on a computer serving that particular Communications Server 2007 or Lync Server pool.

The A/V Authentication service is used to allow RTP voice media to traverse NAT devices and firewalls. This is necessary because media gateways handle signaling only and cannot transport voice securely across a NAT device or firewall. When you configure a Mediation Server in Communications Server 2007 R2, Lync Server 2010, or Lync 2013, you specify the A/V Edge server on which the A/V Authentication service is running so that the Mediation Server will know where to forward the incoming media packets.

For more information about how to deploy Communications Server 2007 R2 or Lync Server 2010 or

2013 and Exchange 2013 Unified Messaging, see the following:

- Deploying Exchange 2013 UM and Lync Server overview
- Checklist: Integrate Exchange 2013 UM with Lync Server

UM services procedures

Unified Messaging > Connect your voice mail system to your telephone network > UM services >

Topic Last Modified: *2013-02-15*

Manage UM settings on a Mailbox server

Manage UM settings on a Client Access server

Allow or prevent call answering on a Mailbox server

Allow or prevent call answering on a Client Access server

Configure the startup mode on a Mailbox server

Configure the startup mode on a Client Access server

Configure the number of incoming calls on a Mailbox server

Start the Microsoft Exchange Unified Messaging service

Stop the Microsoft Exchange Unified Messaging service

Start the Microsoft Exchange Unified Messaging Call Router service

Stop the Microsoft Exchange Unified Messaging Call Router service

Set the TCP listening port on a Client Access server

Set the TLS listening port on a Client Access server

Manage UM settings on a Mailbox server

Connect your voice mail system to your telephone network > UM services > UM services procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-11*

After you install a Mailbox server that is running the Microsoft Exchange Unified Messaging service,

you can configure several options, including the number of concurrent calls, the TCP and Transport Layer Security (TLS) listening ports, the status, and the UM startup mode.

Important:
It's not required that Mailbox servers be added to a UM dial plan before it can process calls for Unified Messaging (UM), except when you're integrating UM and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server. By default, all Mailbox servers in an organization are available to answer incoming calls.

For additional management tasks related to Unified Messaging and Mailbox servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox server (UM service)" entry in the Unified Messaging permissions topic.
- Verify that the Mailbox server is installed, either on the same computer as the Client Access server or on a separate computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to configure Unified Messaging properties on a Mailbox server

This example removes a Mailbox server named `MyMailboxServer` from all Session Initiation Protocol (SIP) dial plans.

```
Set-UMService -Identity MyMailboxServer -DialPlans $null
```

This example adds the Mailbox server named `MyMailboxServer` to a UM SIP dial plan named `MySIPDialPlanName` and also sets the maximum number of incoming voice calls.

```
Set-UMService -Identity MyMailboxServer -DialPlans
MySIPDialPlanName -MaxCalls 150
```

This example sets the startup mode to Dual mode on a Mailbox server named `MyUMServer`.

```
Set-UMService -Identity MyMailboxServer -DialPlans
MySIPDialPlanName -UMStartUpMode -Dual
```

Use the Shell to view Mailbox server properties

This example displays a list of all the Mailbox servers.

```
Get-UMService
```

This example displays a formatted list of properties for the Mailbox server named `MyMailboxServer`.

```
Get-UMService -Identity MyMailboxServer | Format-List
```

Manage UM settings on a Client Access server

Connect your voice mail system to your telephone network > UM services > UM services procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-09*

After you install a Client Access server that is running the Microsoft Exchange Unified Messaging Call Router service, you can configure several options, including the number of concurrent calls, the TCP and Transport Layer Security (TLS) listening ports, the status, and the startup mode.

Note:

It's not required that Client Access servers be added to a UM dial plan before it can process calls for Unified Messaging (UM), except when you're integrating UM and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server. By default, all Client Access servers in an organization are available to answer incoming calls.

For additional tasks related to Unified Messaging and Client Access servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Client Access server (UM call router service)" entry in the Unified Messaging permissions topic.

- Verify that the Client Access server is installed, either on the same computer as the Mailbox server or on a separate computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to configure Unified Messaging properties on a Client Access server

This example removes a Client Access server named `MyClientAccessServer` from all Session Initiation Protocol (SIP) dial plans.

```
Set-UMCallRouterSettings -DialPlans $null - Server
MyClientAccessServer
```

This example adds a Client Access server named `MyClientAccessServer` to a SIP dial plan named `MySIPDialPlan` and also sets the maximum number of incoming voice calls.

```
Set-UMCallRouterSettings -DialPlans MySIPDialPlan -MaxCalls
150 -Server MyClientAccessServer
```

This example sets the SIP TCP listening port to 5077 and the startup mode to Dual mode on a Client Access server named `MyClientAccessServer`.

```
Set-UMCallRouterSettings  -Server MyClientAccessServer-
SipTCPListeningPort 5077 -UMStartUpMode -Dual
```

Use the Shell to view Client Access server properties

This example displays a list of all the Client Access servers.

```
Get-UMCallRouterSettings
```

This example displays a formatted list of properties for the Client Access server.

```
Get-UMCallRouterSettings | Format-List
```

Allow or prevent call answering on a Mailbox server

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-18*

You can allow the Microsoft Exchange Unified Messaging service on a Mailbox server to answer new calls or prevent it from doing so. By default, a Mailbox server is in an enabled state after it's installed. When you're setting the Mailbox server to accept incoming voice, fax, auto attendant and Outlook Voice Access calls, you use the **Set-ServerComponentState** cmdlet.

Configuring Maintenance Mode for a Mailbox server lets you take the server out of service. For a Mailbox server, out-of-service means that the server won't host any active databases, all transport queues are empty, and the server won't accept any incoming calls from Client Access servers, VoIP gateways, IP PBXs, SIP-enabled PBXs, or session border controllers (SBCs).

In Exchange 2007 and Exchange 2010, there was a status parameter that could be used to control the operational status of a Unified Messaging server. In Exchange 2013, no status parameter is available for that purpose on the **Set-UMService** cmdlet for a Mailbox server.

Important:
It's not required that Client Access and Mailbox servers be added to a UM dial plan before they can process calls for Unified Messaging, except when you're integrating UM and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server. By default, all Client Access and Mailbox servers in an organization are available to answer incoming calls.

For additional management tasks related to Mailbox servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange Server Configuration Settings" entry in the Unified Messaging permissions topic.
- Verify that the Mailbox server is installed, either on the same computer as the Client Access server or on a separate computer.
- If you're putting a Mailbox server into Maintenance Mode, verify that there's enough redundancy of all database copies to allow the server to go out of service.
- Before taking a server out of Maintenance Mode, verify the health of the server and make sure it's ready to go into service.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to allow or prevent call answering on a Mailbox server

This example enables a Mailbox server `UMMBXr-05x.contoso.com` to answer incoming voice, fax, auto attendant, and Outlook Voice Access calls from VoIP gateways, IP PBXs, SIP-enabled PBXs, and SBCs, and writes the change to the registry on the UMMBX-05x server.

```
Set-ServerComponentState -Component UnifiedMessaging -
Identity UMMBX-05x.contoso.com -Requester Maintenance -
State Active -LocalOnly
```

This example prevents a Mailbox server `UMMBX-05x.contoso.com` from answering incoming voice, fax, auto attendant, and Outlook Voice Access calls from VoIP gateways, IP PBXs, SIP-enabled PBXs, and SBCs, and writes the change only to Active Directory.

```
Set-ServerComponentState -Component UnifiedMessaging -
Identity UMMBX-05x.contoso.com -Requester Maintenance -
State Inactive -RemoteOnly
```

Allow or prevent call answering on a Client Access server

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-18*

You can allow the Microsoft Exchange Unified Messaging Call Router service on a Client Access server to answer new calls or prevent it from doing so. By default, a Client Access server is in an enabled state after it's installed. When you're setting the Client Access server to accept incoming voice, fax, auto attendant and Outlook Voice Access calls, you use the **Set-ServerComponentState** cmdlet.

Configuring Maintenance Mode for a Client Access server lets you take the server out of service. For a Client Access server, out-of-service means that the server won't accept any incoming calls from VoIP gateways, IP PBXs, SIP-enabled PBXs, or session border controllers (SBCs).

In Exchange 2007 and Exchange 2010, there was a status parameter that could be used to control the operational status of a Unified Messaging server. In Exchange 2013, no status parameter is available for that purpose on the **Set-UMCallRouterSettings** cmdlet on a Client Access server.

Important:

It's not required that Client Access and Mailbox servers be added to a UM dial plan before they can process calls for Unified Messaging, except when you're integrating UM and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server. By default, all Client Access and Mailbox servers in an organization are available to answer incoming calls.

For additional management tasks related to Client Access servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange Server Configuration Settings" entry in the Unified Messaging permissions topic.
- Verify that the Client Access server is installed, either on the same computer as the Mailbox server or on a separate computer.
- If you're putting a Client Access server into Maintenance Mode, verify that there's enough healthy capacity in the Client Access array to allow the server to go out of service.
- Before taking a server out of Maintenance Mode, verify the health of the server and make sure it's ready to go into service.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to allow or prevent call answering on a Client Access server

This example enables a Client Access server `UMCallRouter-05x.contoso.com` to answering incoming voice, fax, auto attendant, and Outlook Voice Access calls from VoIP gateways, IP PBXs, SIP-enabled PBXs, and SBCs, and writes the change to the registry on the UMCallRouter-05x server.

```
Set-ServerComponentState -Component UnifiedMessaging -
Identity UMCallRouter-05x.contoso.com -Requester
```

```
Maintenance -State Active -LocalOnly
```

This example prevents a Client Access server `UMCallRouter-05x.contoso.com` from answering incoming voice, fax, auto attendant, and Outlook Voice Access calls from VoIP gateways, IP PBXs, SIP-enabled PBXs, and SBCs, and writes the change only to Active Directory.

```
Set-ServerComponentState -Component UnifiedMessaging -
Identity UMCallRouter-05x.contoso.com -Requester
Maintenance -State Inactive -RemoteOnly
```

Configure the startup mode on a Mailbox server

Connect your voice mail system to your telephone network > UM services > UM services procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-15*

You can specify the startup mode for the Microsoft Exchange Unified Messaging service on a Mailbox server. By default, the Mailbox server will start up in TCP mode, but if you're using Transport Layer Security (TLS) to encrypt Voice over IP (VoIP) traffic, you must configure the Mailbox server to use TLS or Dual mode. We recommend that Mailbox servers be configured to use Dual as the startup mode. This is because all Client Access servers and Mailbox servers can answer incoming calls for all UM dial plans, and those dial plans can have different security settings (Unsecured, SIP secured, or Secured). If you change the startup mode, you must restart the Microsoft Exchange Unified Messaging service for the change to take effect.

Important:
By default, Mailbox servers are available to answer incoming calls. You don't have to add a Mailbox server to a UM dial plan to process UM calls unless you're integrating UM and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server.

For additional management tasks related to Unified Messaging and Mailbox servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox server (UM service)" entry in the Unified Messaging permissions topic.

- Verify that the Mailbox server is installed, either on the same computer as the Client Access server or on a separate computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the startup mode on a Mailbox server

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Exchange server you want to modify, and then click **Edit** .
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Service settings** > **UM startup mode**, select one of the following from the drop-down list:
   - **TCP** Use this option if you aren't using mTLS and are using only Unsecured dial plans.
   - **TLS** Use this option if you are using mTLS and using only SIP Secured or Secured dial plans.
   - **DUAL** Use this option if you are using mTLS and using Unsecured, SIP Secured, and Secured dial plans.
5. After you select the UM startup mode, click **Save**.

Use the Shell to configure the startup mode on a Mailbox server

This example sets the startup mode for a Mailbox server named `MyUMServer1` to Dual mode.

```
Set-UMService -Identity MyUMServer1 -UMStartUpMode Dual
```

This example sets the startup mode for a Mailbox server named `MyUMServer1` to TLS mode.

```
Set-UMService -Identity MyUMServer1 -UMStartUpMode TLS
```

Configure the startup mode on a Client Access server

Topic Last Modified: *2013-02-15*

You can specify the startup mode for the Microsoft Exchange Unified Messaging Call Router service on a Client Access server. By default, the Client Access server will start up in TCP mode, but if you're using Transport Layer Security (TLS) to encrypt Voice over IP (VoIP) traffic, you must configure the Client Access server to use TLS or Dual mode. We recommend that Client Access servers be configured to use Dual as the startup mode. This is because all Client Access and Mailbox servers can answer incoming calls for all UM dial plans, and those dial plans can have different security settings. If you change the startup mode, you must restart the Microsoft Exchange Unified Messaging Call Router service for the change to take effect.

Important:

By default, Client Access servers are available to answer incoming calls. You don't have to add a Client Access server to a UM dial plan to process UM calls unless you're integrating UM and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server.

For additional management tasks related to Unified Messaging and Client Access servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Client Access server (UM call router service)" entry in the Unified Messaging permissions topic.
- Verify that the Client Access server is installed, either on the same computer as the Mailbox server or on a separate computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the startup mode on a Client Access server

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Exchange server you want to modify, and then click **Edit** ✎.

3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Call Router settings** > **UM startup mode**, select one of the following from the drop-down list:
   - **TCP** Use this option if you aren't using mTLS and are using only Unsecured dial plans.
   - **TLS** Use this option if you are using mTLS and using only SIP Secured or Secured dial plans.
   - **DUAL** Use this option if you are using mTLS and using Unsecured, SIP Secured, and Secured dial plans.
5. After you select the UM startup mode, click **Save**.

Use the Shell to configure the startup mode on a Client Access server

This example sets the startup mode for a Client Access server named `UMCallRouter1` to Dual mode.

```
Set-UMCallRouterSettings -Server UMCallRouter1 -
UMStartUpMode Dual
```

This example sets the startup mode for a Client Access server named `UMCallRouter1` to TLS mode.

```
Set-UMCallRouterSettings -Server UMCallRouter1 -
UMStartUpMode TLS
```

Configure the number of incoming calls on a Mailbox server

Connect your voice mail system to your telephone network > UM services > UM services procedures >

Applies to: *Exchange Server 2013, Exchange Server*

Topic Last Modified: *2013-02-23*

You can configure the number of incoming concurrent connections that a Mailbox server that's running the Microsoft Exchange Unified Messaging service will accept. This includes all incoming calls including Outlook Voice Access, call answering, auto attendants, and fax calls. When you increase the number of concurrent connections on a Mailbox server, more system resources are required than if you decrease the number of concurrent calls. Decreasing the number of concurrent calls is especially important on slower computers on which Unified Messaging services are installed. The range for the number of concurrent voice calls is 0 to 200. The default setting is 100.

For additional tasks related to Unified Messaging and Mailbox servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox server (UM service)" entry in the Unified Messaging permissions topic.
- Verify that you've correctly installed the Client Access and Mailbox servers.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the number of incoming calls on a Mailbox server

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Exchange server you want to modify, and then click **Edit** .
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Service settings**, under **Maximum number of calls allowed**, enter a number from 0 to 200, and click **Save**.

Use the Shell to configure the number of incoming calls on a Mailbox server

This example sets the number of incoming voice, Outlook Voice Access, and fax calls that can be accepted by a Mailbox server named `MyMailboxServer1` to 50.

```
Set-UMService -Identity MyMailboxServer1 -MaxCallsAllowed
50
```

Start the Microsoft Exchange Unified Messaging service

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-16*

You can use the Services snap-in in Microsoft Management Console (MMC) or cmd.exe at a command prompt to start the Microsoft Exchange Unified Messaging service on a Mailbox server. By default, the Microsoft Exchange Unified Messaging service is started after a Mailbox server is installed. However, there may be times when you have to restart the Microsoft Exchange Unified Messaging service manually, for example, when you've taken the Mailbox server offline and have to bring it back online.

When the Microsoft Exchange Unified Messaging service is started on a Mailbox server, the Mailbox server is available to answer and process incoming UM calls.

For additional management tasks related to Mailbox servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- To perform the following procedures, you must log on to the Mailbox server by using an account that's a member of the local Administrators group.
- Verify that the Mailbox server is installed, either on the same computer as the Client Access server or on a separate computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the MMC Services snap-in to start the Microsoft Exchange Unified Messaging service

1. Click **Start**, and then click **Control Panel**.
2. In Control Panel, double-click **Administrative Tools**.
3. In **Administrative Tools**, double-click **Services**.
4. In the **Services** details pane, right-click **Microsoft Exchange Unified Messaging**, and then click **Start**.

Use a command prompt to start the Microsoft Exchange Unified Messaging service

1. Click **Start**, and then click **Run**.
2. In the **Open** box, type the following command, and then press Enter.

```
net start MSExchangeUM
```

Stop the Microsoft Exchange Unified Messaging service

Connect your voice mail system to your telephone network > UM services > UM services procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-16*

You can use the Services snap-in in Microsoft Management Console (MMC) or cmd.exe at a command prompt to stop the Microsoft Exchange Unified Messaging service on a Mailbox server. There may be times when you need to stop this service, for example, when you have to take the Mailbox server offline. When you stop the Microsoft Exchange Unified Messaging service, the Mailbox server won't be able to accept and process incoming calls.

For additional management tasks related to Mailbox servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- To perform the following procedures, you must log on to the Mailbox server by using an account that's a member of the local Administrators group.
- Verify that the Mailbox server is installed, either on the same computer as the Client Access server or on a separate computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the MMC Services snap-in to stop the Microsoft Exchange Unified Messaging service

1. Click **Start**, and then click **Control Panel**.
2. In Control Panel, double-click **Administrative Tools**.
3. In **Administrative Tools**, double-click **Services**.
4. In the **Services** details pane, right-click **Microsoft Exchange Unified Messaging**, and then click **Stop**.

Use a command prompt to stop the Microsoft Exchange Unified Messaging service

1. Click **Start**, and then click **Run**.
2. In the **Open** box, type the following command, and then press Enter.

```
net stop MSExchangeUM
```

Start the Microsoft Exchange Unified Messaging Call Router service

Connect your voice mail system to your telephone network > UM services > UM services procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-16*

You can use the Services snap-in in Microsoft Management Console (MMC) or cmd.exe at a command prompt to start the Microsoft Exchange Unified Messaging Call Router service on a Client Access server. By default, the Microsoft Exchange Unified Messaging Call Router service is started after a Client Access server is installed. However, there may be times when you have to restart or stop the Microsoft Exchange Unified Messaging Call Router service manually, for example, when you've taken the Client Access server offline and have to bring it back online.

When the Microsoft Exchange Unified Messaging Call Router service is started on a Client Access server, the Client Access server is available to answer and process incoming UM calls.

For additional management tasks related to Client Access servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- To perform the following procedures, you must log on to the Client Access server by using an account that's a member of the local Administrators group.
- Verify that the Client Access server is installed, either on the same computer as the Mailbox server or on a separate computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the MMC Services snap-in to start the Microsoft Exchange Unified Messaging Call Router service

1. Click **Start**, and then click **Control Panel**.
2. In Control Panel, double-click **Administrative Tools**.
3. In **Administrative Tools**, double-click **Services**.
4. In the **Services** details pane, right-click **Microsoft Exchange Unified Messaging Call Router**, and then click **Start**.

Use a command prompt to start the Microsoft Exchange Unified Messaging Call Router service

1. Click **Start**, and then click **Run**.
2. In the **Open** box, type the following command, and then press Enter.

```
net start MSExchangeUMCR
```

Stop the Microsoft Exchange Unified Messaging Call Router service

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-16*

You can use the Services snap-in in Microsoft Management Console (MMC) or cmd.exe at a command prompt to stop the Microsoft Exchange Unified Messaging Call Router service on a Client Access server. There may be times when you need to stop this service, for example, when you have to take the Client Access server offline. When you stop the Microsoft Exchange Unified Messaging Call Router service, the Client Access server won't be able to accept and process incoming calls.

For additional management tasks related to Client Access servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- To perform the following procedures, you must log on to the Client Access server by using an account that's a member of the local Administrators group.
- Verify that the Client Access server is installed, either on the same computer as the Mailbox server or on a separate computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the MMC Services snap-in to stop the Microsoft Exchange Unified Messaging Call Router service

1. Click **Start**, and then click **Control Panel**.
2. In Control Panel, double-click **Administrative Tools**.
3. In **Administrative Tools**, double-click **Services**.
4. In the **Services** details pane, right-click **Microsoft Exchange Unified Messaging Call Router**, and then click **Stop**.

Use a command prompt to stop the Microsoft Exchange Unified Messaging Call Router service

1. Click **Start**, and then click **Run**.
2. In the **Open** box, type the following command, and then press Enter.

```
net stop MSExchangeUMCR
```

Set the TCP listening port on a Client Access server

Connect your voice mail system to your telephone network > UM services > UM services procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-09*

You can configure the TCP port that's used to listen for SIP requests on a Client Access server running the Microsoft Exchange Unified Messaging Call Router service. By default, when you install a Client Access server, the SIP TCP listening port number is set to 5060 and the Client Access server starts in TCP mode. The SIP TCP listening port can't be configured by using the EAC. You must configure the SIP TCP listening port number using the **Set-UMCallRouterSettings** cmdlet.

You may have to configure the TCP listening port to 5061 if your VoIP gateways, IP PBXs, or session border controllers (SBCs) are configured to use a TCP port other than the SIP standard 5060.

You can only configure Client Access server TCP and TLS ports. You can't configure the ports for an Exchange 2013 Mailbox server. However, you can use the **Set-UMService** cmdlet to configure the TCP and TLS listening ports for Exchange 2010 UM servers.

For additional tasks related to Unified Messaging and Client Access servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Client Access server (UM call router service)" entry in the Unified Messaging permissions topic.
- Verify that you have correctly installed Client Access and Mailbox servers.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the TCP listening port on a Client Access server

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Exchange server you want to modify, and then click **Edit** .
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Call Router settings**, under **TCP listening port**, enter the number for the TCP port, and click **Save**.

Use the Shell to configure the TCP listening port on a Client Access server

This example sets the TCP listening port on a Client Access server named `MyClientAccessServer` to 5566.

```
Set-UMCallRouterSettings -Server MyClientAccessServer -
SipTCPListeningPort 5566
```

Set the TLS listening port on a Client Access server

Connect your voice mail system to your telephone network > UM services > UM services procedures >

Topic Last Modified: *2013-02-17*

You can configure the Transport Layer Security (TLS) port that's used to listen for SIP requests on a Client Access server running the Microsoft Exchange Unified Messaging Call Router service. By default, when you install a Client Access server, the SIP TLS listening port number is set to 5061.

You may have to configure the TLS listening port to 5061 if you want to:

- Set the VoIP security setting on a UM dial plan to SIP Secured.
- Set the VoIP security setting on a UM dial plan to Secured.
- Integrate with Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server.
- Use mutual Transport Layer Security (mutual TLS) to encrypt network data between Client Access servers, Mailbox servers running the Microsoft Exchange Unified Messaging service, and VoIP

gateways, Private Branch eXchanges (PBXs) enabled for Session Initiation Protocol (SIP), IP PBXs, or session border controllers (SBCs).

If you want to use mutual TLS between a UM IP gateway and a dial plan operating in either SIP Secured or Secured mode, when you create the UM IP gateway you must configure it with a fully qualified domain name (FQDN) and then configure the UM IP gateway to listen on TLS port 5061. You must also verify that any VoIP gateways, PBXs enabled for SIP, IP PBXs, or SBCs have also been configured to listen for mutual TLS requests on port 5061.

You can only configure Client Access server TCP and TLS ports. You can't configure the ports for an Exchange 2013 Mailbox server. However, you can use the **Set-UMService** cmdlet to configure the TCP and TLS listening ports for Exchange 2010 UM servers.

For additional tasks related to Unified Messaging and Client Access servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Client Access server (UM call router service)" entry in the Unified Messaging permissions topic.
- Verify that you have correctly installed Client Access and Mailbox servers.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the TLS listening port on a Client Access server

1. In the EAC, navigate to **Servers** > **Servers**.
2. In the list view, select the Exchange server you want to modify, and then click **Edit** ✎.
3. On the **Exchange Server** page, click **Unified Messaging**.
4. Under **UM Service settings**, under **TLS listening port**, enter the number for the TLS port, and then click **Save**.

Use the Shell to configure the TLS listening port on a Client

Access server

This example sets the TLS listening port on a Client Access server named `MyClientAccessServer` to 5561.

```
Set-UMCallRouterSettings -Server MyClientAccessServer -
SipTlsListeningPort 5561
```

Automatically answer and route incoming calls

Exchange Server 2013 > Unified Messaging >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-08-26*

Microsoft Exchange Unified Messaging (UM) enables you to create a single or multiple UM auto attendants, depending on the needs of your organization. Unlike other Unified Messaging components, such as UM dial plans and UM IP gateways, you aren't required to create UM auto attendants. However, auto attendants help internal and external callers locate users or departments that exist in an organization and transfer calls to them. This topic discusses the UM auto attendant feature found in Unified Messaging.

Contents

Auto attendants

In telephony or Unified Messaging environments, an automated attendant or auto attendant menu system transfers callers to the extension of a user or department without the intervention of a receptionist or an operator. In many auto attendant systems, a receptionist or operator can be reached by pressing or saying zero. The automated attendant is a feature in most modern Private

Branch eXchanges (PBXs), IP PBXs, and Unified Messaging solutions.

Some auto attendant systems use message-only information menus and voice menus so an organization can provide business hours, directions to the premises, information about job opportunities, and answers to other frequently asked questions. After the message plays, callers are forwarded to the receptionist or operator, or they can return to the main menu.

In more complex auto attendant systems, the menu system can be used to search for other auto attendant menus, locate a user in the system, or transfer to another outside telephone line. The menu system can also be used to let the caller interact with the system in certain situations, such as when a student enrolls for a college class or checks a grade, or when you activate a credit card over the telephone.

Although auto attendants can be very useful, if they aren't designed and configured correctly, they can confuse and frustrate callers. For example, specifically in large organizations, when auto attendants aren't designed correctly, callers can be led through a lengthy series of questions and menu prompts before they are finally transferred to a person to answer their questions.

UM auto attendants

Unified Messaging enables you to create one or more UM auto attendants depending on the needs of your organization. UM auto attendants can be used to create a voice menu system for an organization that lets external and internal callers move through the UM auto attendant menu system to locate and place or transfer calls to company users or departments in an organization.

When anonymous or unauthenticated users call an external business telephone number, or when internal callers call a defined extension number, they are presented with a series of voice prompts that help them place a call to a user or locate a user in the organization and then place a call to that user. The UM auto attendant is a series of voice prompts or .wav files that callers hear instead of a human operator when they call an organization that has Unified Messaging. The UM auto attendant lets callers move through the menu system, place calls, or locate users by using dual tone multi-frequency (DTMF) or voice inputs. However, for Automatic Speech Recognition (ASR) or voice inputs to be used, you must enable ASR on the UM auto attendant.

A UM auto attendant has the following features:

- It provides corporate or informational greetings.
- It provides custom corporate menus. You can customize these menus to have more than one level.
- It provides a directory search function that enables a caller to search the organization's directory for a name.
- It enables a caller to connect to the telephone of, or leave a message for, members of the organization.

There is no limit to the number of UM auto attendants you can create. Each Unified Messaging auto attendant can support an unlimited number of extensions. A UM auto attendant can reference one, and only one, UM dial plan. UM auto attendants can also reference or link to other UM auto

attendants.

An incoming call received from an external telephone number or an internal telephone extension is passed between Exchange servers, and then sent to a UM auto attendant. The UM auto attendant is configured by the administrator to use prerecorded voice (.wav) files that are played over the telephone to the caller and that enable the caller to move through the Unified Messaging menu system. You can customize all the .wav files used when you configure a UM auto attendant to meet the needs of your organization.

Return to top

Auto attendants with multiple languages

There are situations in which you may have to provide callers with auto attendants that have different languages. The language setting available on a UM auto attendant enables you to configure the default prompt language on the auto attendant. When you are using the default system prompts for the auto attendant, this is the language that the caller will hear when the auto attendant answers the incoming call. This language setting affects only the default system prompts provided. This language setting doesn't affect custom prompts configured on an auto attendant.

For on-premises and hybrid deployments, when you install the U.S. English version, U.S. English is the only language available to configure on UM auto attendants. If you install a localized version, for example, Japanese, you can configure the auto attendant that you create to use Japanese or U.S. English for the default language. Additional UM language packs can be installed on a Unified Messaging server to enable you to use other default languages on an auto attendant.

For example, if you have a business that's based in the United States but requires a menu system that gives callers the options of U.S. English, Spanish, and French, you must first install the UM language packs that you need. In this case, if you have installed the U.S. English version, you would install the UM language packs for Spanish and French. However, because a Unified Messaging auto attendant can have only one language configured at a time, you would create four auto attendants: a main auto attendant configured to use U.S. English and then one auto attendant for each language: U.S. English, Spanish, and French. You would then configure the main auto attendant to have the appropriate key mappings or menu navigation to access the other auto attendants that you created for each language. In this example, the main auto attendant would answer the incoming call and the caller would hear, "Welcome to Contoso, Ltd. For English, press or say 1. For Spanish, press or say 2. For French, press or say 3."

Tip:

In Exchange UM, authenticated and non-authenticated Outlook Voice Access users can’t search for users in the directory using speech inputs in any language. However, callers that call into an auto attendant can use speech inputs in multiple languages to navigate auto attendant menus and search for users in the directory.

Return to top

Non-business hours and business hours custom greetings

After you create a UM auto attendant, a default system prompt will be used for the non-business hours main menu prompt greeting heard by callers after the non-business hours welcome greeting is played. Although the system prompts mustn't be replaced or changed, you probably want to customize the greetings and menu prompts used with UM auto attendants. Frequently, in addition to configuring a customized non-business hours welcome greeting, you also want to create and configure a custom non-business hours main menu prompt greeting. After you configure a custom non-business hours main menu prompt greeting, you must enable key mappings on the UM auto attendant for non-business hours.

A custom non-business hours main menu prompt greeting is a list of options callers hear during non-business hours. To let callers hear a non-business hours main menu prompt greeting, you first must configure the business and non-business hours schedule by using the EAC or the **Set-UMAutoAttendant** cmdlet in the Shell. For example, "You have reached Trey Research after normal business hours. If you are experiencing a medical emergency, please hang up and dial 911. To leave a message for one of our doctors, press 1. To leave a message for one of our physical therapists, press 2. To leave a general message for one of our front office coordinators, press 3. To be connected with an after hours operator, press 0."

By default, when you create a UM auto attendant, the business and non-business hours greetings or prompts aren't configured and no menu navigation entries are defined for business or non-business hours main menu prompts. To correctly configure customized non-business hours main menu greetings and prompts, you must:

1. Configure business and non-business hours on the **Business hours** page.
2. Create the greeting file that will be used for the non-business hours welcome greeting.
3. Configure the non-business hours welcome greeting on the **Greetings** page.
4. Create the greeting file that will be used for the non-business hours main menu prompt greeting.
5. Configure the non-business hours main menu prompt greeting on the **Greetings** page.
6. Enable menu navigation and add menu navigation entries on the **Menu navigation** page.

Return to top

Menu navigation entries

If you use the default main menu prompt greeting and define a menu navigation entry or multiple menu navigation entries, the UM Text-to-Speech (TTS) engine will synthesize a main menu prompt. However, the TTS engine will only synthesize a main menu prompt if the default greeting is configured and at least one menu navigation entry has been defined. The TTS engine will not synthesize a main menu prompt if you're using a custom main menu prompt, for example, "For the sales department, press 1. For the support department, press 2." To create this main menu prompt, you must create two menu navigation entries: one named "Sales Department" and another named "Support Department", and then configure the key mapping entry to play an audio file, transfer to

an extension number, or send the caller to another auto attendant.

When you configure menu navigation entries, you define the options and the operations that will be performed if a caller speaks a phrase while they're using a speech-enabled auto attendant or presses a key on the telephone keypad while they're using an auto attendant that isn't speech-enabled. To configure menu navigation entries for an auto attendant, you must:

- Enable business hours menu navigation.
- Add menu navigation entries.
- Type the name of the menu navigation entry.
- Select an option in the **When this key is pressed** list, and use the **Play the following audio file** box to upload the audio file to play.
- Configure the action you want performed:
  - Transfer to this extension
  - Transfer to this UM auto attendant
  - Leave a voice message for this user
  - Announce business location
  - Announce business hours

Return to top

Auto attendant examples

The following examples demonstrate how you can use UM auto attendants with Unified Messaging:

- **Example 1** At a company called Contoso, Ltd., external customers can use three external telephone numbers: 425-555-0111 (Corporate Offices), 425-555-0122 (Product Support), and 425-555-0133 (Sales). The Human Resources, Administration, and Accounting departments have internal telephone extensions and must be accessed from the Corporate Offices UM auto attendant.

To create a UM auto attendant structure that supports this scenario, create and configure three UM auto attendants that have the appropriate external telephone numbers. Create three other UM auto attendants for each department in the Corporate Offices. You then configure each UM auto attendant based on your requirements, such as the greeting type or other navigational information.

- **Example 2** At a company called Contoso, Ltd., external customers call one main number for the business, 425-555-0100. When an external caller calls the external number, the UM auto attendant answers and prompts the caller by saying, "Welcome to Contoso, Ltd. Please press or say “One” to be transferred to corporate administration. Please press or say “Two” to be transferred to product support. Please press or say “Three” to be transferred to corporate information. Please press or say “Zero” to be transferred to the operator." To create a UM auto attendant structure that supports this scenario, you create a UM auto attendant that has customized extensions that route the call to the appropriate extension number.

Return to top

DTMF interface

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-08-05*

In Unified Messaging (UM), callers can use dual tone multi-frequency (DTMF), also referred to as touchtone, and voice inputs to interact with the system. The methods that callers can use depend on how the UM dial plans and auto attendants are configured.

The DTMF interface enables callers to use the telephone keypad to locate users and navigate the UM voice mail menu system when they call an Outlook Voice Access number configured on a dial plan or when they call a telephone number configured on an auto attendant. This topic discusses the DTMF interface and how it's used by callers to locate users and to navigate the UM voice mail menu system.

Contents

DTMF overview

DTMF requires a caller to press a key on the telephone keypad that corresponds to a Unified Messaging menu option or to input a user's name or email alias by using the letters on the keys to spell the name or alias. Callers might use DTMF because Automatic Speech Recognition (ASR) hasn't been enabled or because they tried to use voice commands and failed. In either case, DTMF inputs are used to navigate menus and search for users.

By default, in UM, DTMF inputs are used on dial plans and are the default caller interface for UM auto attendants.

Callers can use DTMF inputs for:

- Dial plan dial-in access by using Outlook Voice Access.
- Dial plan directory lookups and searches to locate users.
- Auto attendants that aren't speech-enabled.

- Auto attendants that are speech-enabled that do or don't have a DTMF fallback auto attendant configured.
- DTMF fallback auto attendants (not speech-enabled).

UM dial plans and dial by name

When you create a UM dial plan, you can configure the primary and secondary input method that callers will use to look up names when they search for a user or want to contact a user. These settings are located on the dial plan's **Settings** page and are called **Primary way of searching for names** and **Secondary way of searching for names**. The following options are available for both the primary and secondary ways of searching for names:

- Last First
- First Last
- SMTP address

Additionally, **None** is an available option for the secondary way of searching for names.

By default, **Last First** is selected as the primary way of searching for names and **SMTP address** is selected as the secondary way of searching for names Therefore, when a caller dials in to an Outlook Voice Access number configured on the UM dial plan, the dial plan's welcome message is played and the operator says something like, "Welcome to Contoso Outlook Voice Access. To access your mailbox, enter your extension. To contact someone, press the pound key." After the caller presses the # key, the system responds with "Spell the name of the person you are calling, last name first, or to spell their email alias, press the pound key twice." In this scenario, depending on how your dial plan is configured, the system then prompts the caller to enter the user's last name and then the user's first name (Last First) or to spell the email alias, excluding the domain name. For example, if the user's email alias is tsmith@contoso.com, the caller would enter tsmith.

If you want to change this configuration because the default setting doesn't meet your needs, you can change it to enable callers to enter the user's email alias first or the user's first name followed by the last name. In this case, you would configure the **Primary way of searching for names** with the **SMTP address** setting and configure the **Secondary way of searching for names** with the **First Last** setting. The settings for the dial by name methods will also apply to any UM auto attendants that are associated with the dial plan. For callers to be able to enter the name of the user by using DTMF inputs or the keys on the telephone keypad, a DTMF map and values for the user must exist within your organization’s directory.

For more information about how to change the dial by name primary and secondary methods on a UM dial plan, see Configure the primary way for Outlook Voice Access users to search and Configure the secondary way for Outlook Voice Access users to search.

Return to top

DTMF maps

In an Exchange organization, an attribute named **msExchUMDtmfMap** is associated with each user created in the directory. Unified Messaging uses this attribute to map the user's first name, last name, and email alias to a set of numbers. This mapping is referred to as a DTMF map. A DTMF map enables a caller to enter the digits on the telephone keypad that correspond to the letters of the user's name or email alias. This attribute contains the values needed to create a DTMF map for the user's first name followed by the last name, for the user's last name followed by the first name, and for the user's email alias.

The following table shows the DTMF map values that would be stored in Active Directory on the **msExchUMDtmfMap** attribute for a UM-enabled user named Tony Smith with an alias of tsmith@contoso.com.

DTMF values stored for a UM-enabled user named Tony Smith

Directory entry	User's name
• firstNameLastName:866976484	tonysmith
• lastNameFirstName:764848669	smithtony
• emailAddress:876484	tsmith

Names and email aliases may contain other characters that aren't alphanumeric, such as commas, hyphens, underscores, or periods. Characters such as these won't be used in a DTMF map for a user. For example, if the email alias for Tony Smith is tony-smith@contoso.com, the DTMF map value would be 866976484, and the hyphen wouldn't be included. However, if a user's email alias contains a number or numbers, for example, tonysmith123@contoso.com, the numbers would be used in the DTMF map that's created. The DTMF map for tonysmith123 would be 866976484123.

A DTMF map must exist for a user for callers to be able to enter the user's name or email alias. However, not all users will have a DTMF map associated with their user account.

Return to top

DTMF maps for users who aren't enabled for Unified Messaging

Users, including mailbox-enabled users, aren't enabled for Unified Messaging by default. The **msExchUMDtmfMap** attribute is populated with the values needed for DTMF maps for users who haven’t been enabled for UM. By default, the following DTMF maps are created for all users when a mailbox is created for them:

1. emailAddress
2. firstNameLastName
3. lastNameFirstName

If a user doesn’t have DTMF map values defined for their account, callers won’t be able to contact

the user when they press a telephone key from a UM auto attendant menu or perform a directory search. Also, UM-enabled users won't be able to send messages or transfer calls to users who don't have a DTMF map unless they can use Automatic Speech Recognition (ASR). To enable callers to transfer calls or contact users who aren't UM-enabled by using the telephone keypad, you need to create the necessary values for the DTMF map for those users. You can use the **Set-User** cmdlet with the *-CreateDtmfMap* parameter to create and update a single user's DTMF map or update a DTMF map for a user if the name of the user was changed after a DTMF map was created. Optionally, you can create an Exchange Management Shell script by using this cmdlet to update the DTMF map values for multiple users.

For more information about the **Set-User** cmdlet, see Set-User.

Return to top

DTMF maps for users who are enabled for Unified Messaging

By default, a DTMF map is created for a user when they're enabled for Unified Messaging. This makes it possible for calls to be transferred to the UM-enabled user from external callers, from users who aren't enabled for UM, and from other UM-enabled users who use the telephone keypad to spell the user's name or email alias.

After the DTMF map values have been created for a UM-enabled user, callers can use the directory search feature. Callers use directory search when they use the telephone keypad in the following situations:

- To identify or search for a user when they call in to an Outlook Voice Access number.
- To locate or transfer calls to a UM-enabled user when they call in to a UM auto attendant.

For more information about how to enable a user for Unified Messaging, see Enable a user for voice mail.

Sometimes a user's first name, last name, or email alias changes after the user is enabled for UM. The user's DTMF map values aren't updated automatically. If a caller enters the user's new name or email alias and the user's DTMF map hasn't been updated to reflect the change to the name or email alias, the caller won't be able to locate the user in the directory, send a message to the user, or transfer calls to the user. If you have to update a user's DTMF map after the user has been enabled for UM, you can use the **Set-User** cmdlet with the *-CreateDtmfMap* parameter. You can also create an Exchange Management Shell script using this cmdlet if you want to update the DTMF maps for multiple UM-enabled users.

Caution:

We recommend that you don't manually change the DTMF values for users by using a tool such as ADSI Edit because it might result in inconsistent configurations or other errors. We recommend that you use only the **Set-UMService** cmdlet or the **Set-User** cmdlet to create or update DTMF maps for users.

For more information

Adsiedit Overview

UM auto attendant procedures

Exchange Server 2013 > Unified Messaging > Automatically answer and route incoming calls >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-01-30*

Set up a UM auto attendant

Create a UM auto attendant

Add an auto attendant extension number

Configure business hours

Create a holiday schedule

Enter a business name

Set a business location

Configure the time zone

Enable a customized business hours greeting

Enable a customized business hours menu prompt

Enable a customized non-business hours greeting

Enable a customized non-business hours menu prompt

Enable an informational announcement

Create menu navigation

Create business hours navigation menus

Create non-business hours navigation menus

Manage a UM auto attendant

Configure a DTMF fallback auto attendant

Enable a UM auto attendant

Disable a UM auto attendant

Delete a UM auto attendant

Enable or disable automatic speech recognition

Enable or prevent transferring calls from an auto attendant

Enable or disable sending voice messages to users

Enable or disable directory lookups

Configure the group of users that can be contacted

Configure an auto attendant for users who have similar names

Set up a UM auto attendant

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

In addition to allowing users access to voice mail, Unified Messaging (UM) allows you to create one or more UM auto attendants depending on the needs of your organization. UM auto attendants can be used to create a voice menu system for an organization that lets external and internal callers locate, place, or transfer calls to company users or departments in an organization.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

Auto attendants

In telephony or Unified Messaging environments, an automated attendant or auto attendant menu system transfers callers to the extension of a user or department without the intervention of a receptionist or an operator. In many auto attendant systems, a receptionist or operator can be reached by pressing or saying zero. Some auto attendant systems use message-only information menus and voice menus so an organization can provide business hours, directions to the premises, information about job opportunities, and answers to other frequently asked questions. After the message plays, callers are forwarded to the receptionist or operator, or they can return to the main menu.

Although auto attendants can be very useful, if they aren't designed and configured correctly, they can confuse and frustrate callers. For example, especially in large organizations, when auto attendants aren't designed correctly, callers can be led through a lengthy series of questions and menu prompts before they’re finally transferred to a person to answer their questions.

How do I set up an auto attendant?

In the Exchange Administration Center (EAC), you set up and manage UM auto attendants to automatically answer calls to your organization and allow callers to self-select different options using the keys on their telephone. You can have just one UM auto attendant that provides basic menu navigation for callers to your organization, or you can have multiple nested and branching auto attendants that provide a richer experience for your callers. However, in both cases, you must plan and set up your auto attendants carefully.

To plan and create a new UM auto attendant structure, you need to do the following:

1. Decide whether you want to allow users to interact with the auto attendant using speech inputs.
2. Decide which language you want to use for your main auto attendant and whether you need to create other auto attendants to support more languages.
3. Decide on the business and non-business hours for the auto attendant and set the business hours using **Business hours**. Though it's not required, you can also decide on the holiday schedule for this auto attendant.

Note:
You should also set the time zone on the attendant.

4. Decide whether you want standard system-generated business and non-business hours greetings or to create custom recordings for them.

If you want to use custom greetings, plan and record your business and non-business hour greetings to play to callers during business and non-business hours. If you need to, you can also create a custom informational announcement greeting. For example, for your business hours greeting you could use "Welcome to Contoso. For English, press or say 1, for Spanish, press or say 2." For your non-business hours greeting, you could record the following script: "Welcome to Contoso. Our office is currently closed. We will be open on Monday at 8:00 am."

5. Plan your auto attendant structure based on your business needs. For example, one organization may be a multinational business with offices in both Germany and the UK, and thus need an auto attendant structure based on multiple languages. Another organization might have its corporate office at one site, Sales located at another site, and Customer Service located at a third site, and thus need an auto attendant that directly relates to the structure of the organization.
6. Decide if you'll need DTMF fallback auto attendants or other auto attendants to use when auto attendant voice commands don't work.
7. Plan the menu navigation for business hours and non-business hours. For each auto attendant, including DTMF auto attendants, you'll need to plan and configure menu prompts and menu navigation entries. You'll need to do this for both business and non-business hours.
8. The following is an example of a worksheet you could use to plan non-business hours menu navigation.

Key	Prompt/Navigation menu entry name	Response to record
1	Language selection to use	"Press or say 1 to use English."

	English.	
2	Account balance	“Press or say 2 to get your account balance.”
3	Transfer to Sales	"Press or say 3 to be transferred to our sales department."
4	Transfer to customer service	“Press or say 4 to be transferred to the next customer service representative.”
5	Business hours	No response needed.
6	Business location	No response needed.

9. Using your menu navigation plan, record prompts that inform callers what they can do. For example, depending on the auto attendant structure for the non-business hours menu navigation shown in the table, you might record the following script: "To leave a message for Sales, press one. For our business hours, press two. For our address, press three."
10. Determine how callers will access your organization. Consider how they will search for and contact users in your organization. Also consider how to transfer callers, including how they'll get to a live person or organization representative, and whether callers will access an operator during business and non-business hours.
11. Determine what calls you'll allow callers to make when they're using a specific auto attendant. For example, whether you want to allow callers to make calls to users in a single dial plan, to any extension, or whether you'll allow them to make calls outside your organization.
12. After you've planned your auto attendant settings, greetings and menu navigation, and created audio files that contain your recorded greetings, menu navigation prompts, and menu navigation responses, you're ready to create and configure your auto attendant. Here's how:
    - Create a UM auto attendant
    - Manage a UM auto attendant
13. If you've created the auto attendant structure and settings, enable the UM auto attendant so it can start accepting calls.

Create a UM auto attendant

Automatically answer and route incoming calls > UM auto attendant procedures > Set up a UM auto attendant >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-08*

After you create a Unified Messaging (UM) auto attendant, incoming calls to an external telephone number that a human operator would ordinarily answer are answered by the auto attendant. Unlike with other Unified Messaging components, such as UM dial plans and UM IP gateways, you aren't required to create UM auto attendants. However, auto attendants help internal and external callers locate users or departments that exist in an organization and transfer calls to them.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to create a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**, select the UM dial plan for which you want to add an auto attendant, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, click **New +**.
3. On the **New UM auto attendant** page, enter the following information:
   - **Name** Use this box to create the display name for the UM auto attendant. A UM auto attendant name is required and must be unique. However, it's used only for display purposes in the EAC and the Shell.

If you have to change the display name of the auto attendant after it's created, you must first delete the existing UM auto attendant and then create another auto attendant that has the appropriate name. If your organization uses multiple UM auto attendants, we recommend that you use meaningful names for your UM auto attendants. The maximum length of a UM auto attendant name is 64 characters, and it can include spaces.

Although you can name a new UM auto attendant to include spaces, if you integrate Unified Messaging with Office Communications Server 2007 R2 or Microsoft Lync Server, the name of the

auto attendant can't include spaces. Therefore, if you created an auto attendant with spaces in the display name, and you're integrating with Office Communications Server 2007 R2 or Lync Server, you must first delete that auto attendant and then create another auto attendant that doesn't include spaces in the display name.

- **Create this auto attendant as enabled** Select this check box to enable the auto attendant to answer incoming calls when you complete the New UM Auto Attendant Wizard. By default, a new auto attendant is created as disabled.

If you decide to create the UM auto attendant as disabled, you can use the EAC or the Shell to enable the auto attendant after you finish the wizard.

- **Set the auto attendant to respond to voice commands** Select this check box to speech-enable the UM auto attendant. If the auto attendant is speech-enabled, callers can respond to the system or custom prompts used by the UM auto attendant using touchtone or voice inputs. By default, the auto attendant won't be speech-enabled when it's created.

For callers to use a speech-enabled auto attendant, you must install the appropriate UM language pack that contains Automatic Speech Recognition (ASR) support and configure the properties of the auto attendant to use this language.

- **Access numbers** Use this box to enter the extension numbers or telephone numbers that callers will use to reach the auto attendant. Type an extension number or telephone number in the box, and then click **Add +** to add the number to the list. The number of digits in the extension number or telephone number that you provide doesn't have to match the number of digits for an extension number configured on the associated UM dial plan. This is because direct calls are allowed to UM auto attendants.

The number of extension numbers or telephone numbers entered is unlimited. However, you may create the new auto attendant without an extension number listed. An extension number or telephone number isn't required.

You can edit or remove an existing extension number or telephone number. To edit an existing extension number or telephone number, click **Edit +**. To remove an existing extension number or telephone number from the list, click **Remove −**.

4. Click **Save**.

Use the Shell to create a UM auto attendant

This example creates a UM auto attendant named `MyUMAutoAttendant` that can accept incoming calls but isn't speech-enabled.

```
New-UMAutoAttendant -Name MyUMAutoAttendant -UMDialPlan
MyUMDialPlan -PilotIdentifierList 55000 -Enabled $false
```

This example creates a speech-enabled UM auto attendant named `MyUMAutoAttendant`.

```
New-UMAutoAttendant -Name MyUMAutoAttendant -UMDialPlan
MyUMDialPlan -PilotIdentifierList 56000,56100 -
```

```
SpeechEnabled $true
```

Add an auto attendant extension number

Automatically answer and route incoming calls > UM auto attendant procedures > Set up a UM auto attendant >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

You can configure an extension number or multiple extension numbers on a Unified Messaging (UM) auto attendant. When you add an extension number to a UM auto attendant, that number can be used by callers to call into the auto attendant. Also, you may have to add extension numbers because there is more than one extension number that callers can use to access an auto attendant. By default, no extension numbers are configured when you create an auto attendant.

You can create a new auto attendant without setting up an extension number for the auto attendant. You can also associate more than one telephone or extension number with a single auto attendant. You can either add the extension numbers when you create the UM auto attendant or add them after you configure the auto attendant. The number of digits in the extension number you configured on the UM auto attendant must match the number of digits for an extension number that's configured on the UM dial plan associated with the UM auto attendant.

Note:

You can also add a Session Initiation Protocol (SIP) address instead of adding an extension number. A SIP address is used by some IP Private Branch eXchanges (PBXs) and Office Communications Server 2007 R2 or Microsoft Lync Server.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to add an extension or phone numbers for a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to edit and click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant you want to add extension or phone numbers to.
3. On the toolbar, click **Edit** .
4. On the **UM Auto Attendant** page > **General**, under **Access numbers**, in the text box, enter the extension or phone number that you want to use and click **Add +**.
5. Click **Save** to add the number.

Use the Shell to configure an extension number on a UM auto attendant

This example configures a UM auto attendant named `MyUMAutoAttendant` with multiple extension numbers.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
PilotIdentifierList "12345, 72000, 75000"
```

Configure business hours

Automatically answer and route incoming calls > UM auto attendant procedures > Set up a UM auto attendant >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-19*

When you configure business hours for a Unified Messaging (UM) auto attendant, you define the

hours of the day that your organization is open, and the business hours greetings and menu prompts callers will hear when they call an extension number that's configured on the auto attendant. If a caller reaches the auto attendant during hours that are outside the business hours you define, the caller will hear the non-business hours prompts and greetings.

Several default schedule options are available in the EAC. For example, most businesses are open from 8:00 A.M. to 5:00 P.M., Monday through Friday. Sometimes the default options won't fit your needs and you'll want to customize the schedule. If your business hours vary from the schedules defined by the system, you can define a customized schedule for the auto attendant.

By default, the UM auto attendant will play the business hours prompts and greetings regardless of the time of day callers dial in to the auto attendant.

Note:

When you set the schedule for business and non-business hours on a UM auto attendant, make sure the time zone is configured correctly.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to specify business hours for a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .

2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to set the business hours, and then click **Edit** .
3. On the **UM Auto Attendant** page > **Business Hours** under **Business hours**, click **Configure business hours**.
4. On the **Configure Business Hours** page, select the hours you want to use as your business hours for each day of the week.
5. Click **OK**, and then click **Save**.

Use the Shell to specify business hours for a UM auto attendant

This example sets the business hours for a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessHoursSchedule 0.10:45-0.13:15,1.09:00-
1.17:00,6.09:00-6.16:30
```

Create a holiday schedule

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-19*

You can define the dates and times your organization will be closed for holidays and other occasions. Between the start dates and the end dates you specify, callers who reach the Unified Messaging (UM) auto attendant will hear a holiday greeting you specify when you configure the holiday schedule. After the caller hears the holiday greeting you've specified, the non-business hours greeting and menu prompts will be played for the caller.

You can also create a holiday schedule within an existing holiday schedule. When you create multiple holiday schedules, Unified Messaging lets you overlap your scheduled holiday times. For example, you can define a holiday schedule from December 15th through December 31st when your organization will be closed for construction, and you can define another holiday schedule from December 24th through December 26th. When callers call in to the auto attendant from December 15th through December 23rd and from December 27th through December 31st, they'll be presented with the holiday greeting that you've specified for this schedule. For example, "We are currently closed for construction." When callers call in to the auto attendant from December 24th through December 26th, they'll be presented with another holiday greeting, such as "We are currently closed for business so that our employees can enjoy the holidays with their families."

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to specify a holiday schedule for a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then on the toolbar, click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to set the holiday schedule. On the toolbar, click **Edit** .
3. On the **UM Auto Attendant** page > **Business Hours**, under **Holiday schedule**, click **Add +**.
4. On the **New Holiday** page, configure the following:
   - **Name** Enter a name for your holiday schedule.
   - **Holiday greeting** Browse to the .wav file you want to use as your greeting. This is a required field.
   - **Start date** Use this list to select the date you want the holiday to start. The holiday schedule will start at midnight on the date specified in this list.
   - **End date** Use this list to select the date you want the holiday to end. The holiday schedule will end at 11:59 P.M. on the date specified in this list.
5. After you've configured your holiday schedule, click **OK**, and then click **Save**.

Use the Shell to specify a holiday schedule for a UM auto

attendant

This example configures a UM auto attendant named `MyUMAutoAttendant` that has business hours configured to be 10:45 to 13:15 (Sunday), 09:00 to 17:00 (Monday), and 09:00 to 16:30 (Saturday), and holiday times and their associated greetings configured to be "New Year" on January 2, 2013, and "Building Closed for Construction" from April 24, 2013 through April 28, 2013.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessHoursSchedule 0.10:45-0.13:15,1.09:00-
1.17:00,6.09:00-6.16:30 -HolidaySchedule "New
Year,newyrgrt.wav,1/2/2013","Building Closed for
Construction,construction.wav,4/24/2013,4/28/2013"
```

Enter a business name

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-19*

You can enter the name of your business in the **Business name** box on a UM auto attendant. By default, no business name is entered. If you enter a business name, a default greeting prompt with the business name will be played to callers when they call in to the Unified Messaging (UM) auto attendant.

For additional tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure a business name

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to set a business name, and then, on the toolbar, click **Edit** .
3. On the **UM Auto Attendant** page > **General**, under **Business name**, type the name of the business.
4. Click **Save**.

Use the Shell to configure a business name

This example sets the business name on a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessName "Northwind Traders"
```

Set a business location

Automatically answer and route incoming calls > UM auto attendant procedures > Set up a UM auto attendant >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-23*

You can specify the location of a business on a Unified Messaging (UM) auto attendant so that the location will be played for callers. By default, no business location is entered.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging

permissions topic.

- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure a business location

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to set the business location, and then click **Edit** .
3. On the **UM Auto Attendant** page > **General**, under **Business location**, type the location of the business.
4. Click **Save**.

Use the Shell to configure a business location

This example sets the business location on a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessLocation 'Redmond'
```

Configure the time zone

Automatically answer and route incoming calls > UM auto attendant procedures > Set up a UM auto attendant >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-17*

By default, the Unified Messaging (UM) auto attendant uses the time zone of the Mailbox server on which it's created. However, there are situations where you may have to change the time zone for a UM auto attendant to a different time zone. For example, if you have two UM dial plans and each

dial plan represents a different time zone, you must configure one UM auto attendant to have the same time zone as the Mailbox server and the other UM auto attendant to have a time zone that differs from the Mailbox server.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the time zone

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to set the time zone, and then click **Edit** .
3. On the **UM Auto Attendant** page, click **Business Hours**, and then, under **Time zone**, select the time zone from the drop-down list.
4. To save your changes, click **OK**, and then click **Save**.

Use the Shell to configure the time zone

This example sets the time zone to the Pacific time zone on a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
TimeZoneName Pacific
```

Enable a customized business hours greeting

Automatically answer and route incoming calls > UM auto attendant procedures > Set up a UM auto attendant >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-19*

You can enable a customized business hours greeting for a Unified Messaging (UM) auto attendant. The business hours greeting is the first thing callers hear when a UM auto attendant answers their call during business hours. You'll probably want to customize the greeting.

Unified Messaging includes a default system prompt for use during business hours. Although the default system prompt mustn't be replaced or changed, you may want to provide an customized greeting. You can create a customized greeting in the .wav or .wma file format to be used when callers call in to a UM auto attendant during business hours. For example, "You've reached Woodgrove Bank."

If you want to include the name of your organization or business as part of the default greeting, you can enter the name in the **Business name** box on the UM auto attendant. For details, see Enter a business name.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- Create a .wav or .wma file to be used for the greeting.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable a customized business hours greeting

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to enable a customized business hours greeting, and then click **Edit** .
3. On the **UM Auto Attendant** page, > **Greetings**, under **Business hours greeting** click **Change**, and then click **Browse** to locate the customized business hours greeting file you created before you started this procedure.

Important:
The file you use for the greeting must be a .wav or .wma file.

4. After you've located the file, click **Open**, and then click **Save**.

Use the Shell to enable a customized business hours greeting

This example enables the business hours greeting that uses a customized greeting named `GreetingFile.wav` for the UM auto attendant `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessHoursWelcomeGreetingEnabled $true -
BusinessHoursWelcomeGreetingFilename GreetingFile.wav
```

This example configures a UM auto attendant named `MyUMAutoAttendant` to have business hours configured to be 10:45 to 13:15 (Sunday), 09:00 to 17:00 (Monday), and 09:00 to 16:30 (Saturday) and holiday times and their associated greetings configured to be "`New Year`" on January 2, 2013, and "`Building Closed for Construction`" from April 24, 2013 through April 28, 2013.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessHoursSchedule 0.10:45-0.13:15,1.09:00-
1.17:00,6.09:00-6.16:30 -HolidaySchedule "New
Year,newyrgrt.wav,1/2/2013","Building Closed for
Construction,construction.wav,4/24/2013,4/28/2013"
```

This example configures a UM auto attendant named `MyAutoAttendant` and enables business hours key mappings so that when callers press 1, they're forwarded to another UM auto attendant named

SalesAutoAttendant. When they press 2, they're forwarded to extension number 12345 for Support, and when they press 3, they're sent to another auto attendant that plays an audio file.

```
Set-UMAutoAttendant -Identity MyAutoAttendant -
BusinessHoursKeyMappingEnabled $true -
BusinessHoursKeyMapping
"1,Sales,,SalesAutoAttendant","2,Support,12345","3,Directio
ns,,,directions.wav"
```

Enable a customized business hours menu prompt

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-19*

You can customize the menu prompt to be used by a Unified Messaging (UM) auto attendant during business hours. After you create a UM auto attendant, a default system prompt ("Welcome to Unified Messaging") is used as the menu prompt that callers hear after the business hours welcome greeting is played. Although the system prompt mustn't be replaced or changed, you can customize the greetings and menu prompts that are used with UM auto attendants. After you create a customized business hours menu prompt audio file, you must enable menu navigation entries on the UM auto attendant for business hours.

If you only want to include the name of your organization or business as part of the default system prompt, you can enter the name in the **Business name** box on the UM auto attendant. For details, see Enter a business name.

Important:
You must configure business hours on the auto attendant. For details, see Configure business hours.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.

- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- Create a .wav or .wma file to be used for the menu prompt.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable a customized business hours menu prompt

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan that you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to enable a customized business hours menu prompt, and then click **Edit** .
3. On the **UM Auto Attendant** page, > **Menu navigation**, under **Business hours menu navigation** click **Change**, and then click **Browse** to locate the customized business hours menu prompt file.

Important:

The file you use for the menu prompt must be a .wav or .wma file.

4. After you've located the file, click **Open**, and then click **Save**.

Use the Shell to enable a customized business hours menu prompt

This example enables a business hours main menu prompt and uses a customized prompt named `businesshoursprompts.wav` on the UM auto attendant `MyUMAutoAttendant`.

```
Command Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessHoursMainMenuCustomPromptEnabled $true -
BusinessHoursMainMenuCustomPromptFilename
BusinessHoursPrompts.wav
```

This example configures a UM auto attendant named `MyUMAutoAttendant` that has business hours configured to be 10:45 to 13:15 (Sunday), 09:00 to 17:00 (Monday), and 09:00 to 16:30 (Saturday) and holiday times and their associated greetings configured to be "`New Year`" on January 2, 2013, and "`Building Closed for Construction`" from April 24, 2013 through April 28, 2013.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessHoursSchedule 0.10:45-0.13:15,1.09:00-
1.17:00,6.09:00-6.16:30 -HolidaySchedule "New
Year,newyrgrt.wav,1/2/2013","Building Closed for
Construction,construction.wav,4/24/2013,4/28/2013"
```

This example configures a UM auto attendant named `MyAutoAttendant` and enables business hours navigation menus so that when callers press 1, they're forwarded to another UM auto attendant named `SalesAutoAttendant`. When they press 2, they're forwarded to extension number 12345 for `Support`, and when they press 3, they're sent to another auto attendant that plays an audio file.

```
Set-UMAutoAttendant -Identity MyAutoAttendant -
BusinessHoursKeyMappingEnabled $true -
BusinessHoursKeyMapping
"1,Sales,,SalesAutoAttendant","2,Support,12345","3,Directio
ns,,,directions.wav"
```

Enable a customized non-business hours greeting

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-30*

You can enable a customized non-business hours greeting for a Unified Messaging (UM) auto attendant. The non-business hours greeting is the first thing callers hear when a UM auto attendant answers their call during non-business hours. You'll probably want to customize the greeting.

Unified Messaging includes a default system prompt for use during non-business hours. Although the default system prompt mustn't be replaced or changed, you may want to provide an customized greeting. You can create a customized greeting in the .wav or .wma file format to be used when callers call in to a UM auto attendant during non-business hours. For example, "You've reached Woodgrove Bank after hours."

If you want to include the name of your organization or business as part of the default greeting, you can enter the name in the **Business name** box on the UM auto attendant. For details, see Enter a business name.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- Create a .wav or .wma file to be used for the greeting.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable a customized non-business hours greeting

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to enable a customized non-business hours greeting, and then click **Edit** .
3. On the **UM Auto Attendant** page, > **Greetings**, under **Non-business hours greeting**, click **Change**, and then click **Browse** to locate the customized non-business hours greeting file you created before you started this procedure.

Important:

The file you use for the greeting must be a .wav or .wma file.

4. After you've located the file, click **Open**, and then click **Save**.

Use the Shell to enable a customized non-business hours greeting

This example enables the non-business hours greeting that uses a customized greeting named `GreetingFile.wav` for the UM auto attendant `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
AfterHoursWelcomeGreetingEnabled $true -
AfterHoursWelcomeGreetingFilename GreetingFile.wav
```

This example configures a UM auto attendant named `MyUMAutoAttendant` that has business hours configured to be 10:45 to 13:15 (Sunday), 09:00 to 17:00 (Monday), and 09:00 to 16:30 (Saturday) and holiday times and their associated greetings configured to be "`New Year`" on January 2, 2013, and "`Building Closed for Construction`" from April 24, 2013 through April 28, 2013.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessHoursSchedule 0.10:45-0.13:15,1.09:00-
1.17:00,6.09:00-6.16:30 -HolidaySchedule "New
Year,newyrgrt.wav,1/2/2013","Building Closed for
Construction,construction.wav,4/24/2013,4/28/2013"
```

This example configures a UM auto attendant named `MyAutoAttendant` and enables non-business hours key mappings so that when callers press 1, they're forwarded to another UM auto attendant named `SalesAutoAttendant`. When they press 2, they're forwarded to extension number 12345 for `Support`, and when they press 3, they're sent to another auto attendant that plays an audio file.

```
Set-UMAutoAttendant -Identity MyAutoAttendant -
BusinessHoursKeyMappingEnabled $true -
BusinessHoursKeyMapping
"1,Sales,,SalesAutoAttendant","2,Support,12345","3,Directio
ns,,,directions.wav"
```

Enable a customized non-business hours menu prompt

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-22*

You can customize the menu prompt to be used by a Unified Messaging (UM) auto attendant outside business hours. After you create a UM auto attendant, a default system prompt (“Welcome to Unified Messaging”) is used as the menu prompt that callers hear after the non-business hours welcome greeting is played. Although the system prompt mustn't be replaced or changed, you can customize the greetings and menu prompts that are used with UM auto attendants. After you create a customized non-business hours menu prompt audio file, you must enable menu navigation entries on the UM auto attendant for non-business hours.

If you only want to include the name of your organization or business as part of the default system prompt, you can enter the name in the **Business name** box on the UM auto attendant. For details, see Enter a business name.

Important:

You must configure business hours on the auto attendant. When you configure business hours, the non-business hours are set automatically. For details, see Configure business hours.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- Create a .wav or .wma file to be used for the menu prompt.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable a customized non-business hours

menu prompt

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan that you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to enable a customized non-business hours menu prompt, and then click **Edit** .
3. On the **UM Auto Attendant** page > **Menu navigation**, under **Non-business hours menu navigation**, click **Change**, and then click **Browse** to locate the customized non-business hours menu prompt file.

Important:
The file you use for the menu prompt must be a .wav or .wma file.

4. After you've located the file, click **Open**, and then click **Save**.

Use the Shell to enable a customized non-business hours menu prompt

This example enables a UM auto attendant named `MyUMAutoAttendant` that has business hours configured to be 10:45 to 13:15 (Sunday), 09:00 to 17:00 (Monday), and 09:00 to 16:30 (Saturday) and holiday times and their associated greetings configured to be "`New Year`" on January 1, 2013, and "`Building Closed for Construction`" from April 24, 2013 through April 28, 2013.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessHoursSchedule 0.10:45-0.13:15,1.09:00-
1.17:00,6.09:00-6.16:30 -HolidaySchedule "New
Year,newyrgrt.wav,1/2/2013","Building Closed for
Construction,construction.wav,4/24/2013,4/28/2013"
```

This example configures a UM auto attendant named `MyAutoAttendant` and enables non-business hours navigation menus so that when callers press 1, they're forwarded to another UM auto attendant named `SalesAutoAttendant`. When they press 2, they're forwarded to extension number 12345 for `Support`, and when they press 3, they're sent to another UM auto attendant that plays an audio file.

```
Set-UMAutoAttendant -Identity MyAutoAttendant -
AfterHoursKeyMappingEnabled $true -
AfterHoursKeyMapping
"1,Sales,,SalesAutoAttendant","2,Support,12345","3,Directio
ns,,,directions.wav"
```

Enable an informational announcement

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-19*

You can enable an informational announcement for a Unified Messaging (UM) auto attendant. When an informational announcement is enabled, it will play immediately after the business or non-business hours greeting. By default, an informational announcement isn't configured. To enable an informational announcement, create a .wav or .wma file to be used as the informational announcement, and then configure the auto attendant to use this sound file.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- Create a .wav or .wma file to be used for the informational announcement.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable an informational announcement

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan that you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to enable an informational announcement, and then click **Edit** .

3. On the **UM Auto Attendant** page, > **Greetings**, under **Informational announcement** click **Change**, and then click **Browse** to locate the informational announcement file you created before you started this procedure.

Important:
The file you use for the greeting must be a .wav or .wma file.

4. After you've located the file, click **Open**, and then click **Save**.

Use the Shell to enable an informational announcement

This example enables an informational announcement that uses the `MyInfoAnnouncement.wav` file for the UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
InfoAnnouncementEnabled $true -InfoAnnouncementFilename
MyInfoAnnouncement.wav
```

Create menu navigation

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

You can use the **New menu navigation entry** page to create single or multiple key mappings for business or non-business hours main menu prompts for auto attendants. You can define the action that will be performed when a key on the telephone keypad is pressed, for example, transferring the call to an extension number or another auto attendant.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure UM auto attendant navigation menus

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to create menu navigation. On the toolbar, click **Edit** .
3. On the **UM Auto Attendant** page, click **Menu navigation**, select either **Enable business hours menu navigation** or **Enable non-business hours menu navigation**, and then click **Add +**.
4. On the **New menu navigation entry** page, configure the following:
   - **Prompt** Use this box to type the name of the new navigation menu. The navigation menu name is used for display purposes only. This is a required field.

Because you may want to specify multiple new navigation menus, we recommend that you use meaningful names for your key mappings. The maximum length of the name for the key mapping is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

   - **When this key is pressed** Use this list to enable key mapping. The key mapping is the number key that a caller presses to have the auto attendant perform a specific operation, for example, forwarding the caller to another auto attendant or to an operator. By default, no entries are defined.

Use the drop-down list to select the numeric key (from 1 through 9) that the caller must press. Zero (0) is reserved for the auto attendant operator.

If you select **Time Out** from the drop down list, it enables callers to be transferred to an extension number or to another auto attendant if they don't press a key on the telephone keypad. For example, "Please stay on the line and your call will be answered by the next available representative." The default setting is 5 seconds. If you enable this option, a blank key mapping will be created.

   - **Play the following audio file** Use this option to select a previously recorded audio file for callers. Click **Change**, and then click **Browse** to locate the audio file.
   - **Perform this additional action** Select one of the following options to define the action that you want the auto attendant to perform for the caller:
     - **None** If you don’t want to the auto attendant to transfer the call to an extension or to

another auto attendant, or leave a message for a user, use this option.

- **Transfer to this extension** Select this option to enable calls to be transferred to an extension number. If you enable this option, use the box to type the extension where the call will be transferred. This field allows only numeric characters. It can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
- **Transfer to this UM auto attendant** Select this option to transfer the call to an auto attendant. Click **Browse** to locate the auto attendant that you want to use. Before you enable this option, you must first create and configure the auto attendant. This option is used when you create a parent/child structure of UM auto attendants.
- **Leave a voice message for this user** Select this option to enable a caller to leave a voice mail message for a user that's on the same dial plan as the UM auto attendant that you're configuring. When a caller chooses this option from an auto attendant menu, they'll be prompted to leave a voice message for the user that was selected. Click **Browse** to locate the UM-enabled user.
- **Announce business location** Select this option to enable a caller to choose an auto attendant menu option and hear the location of the business that's configured on the UM auto attendant. To enable this to work correctly, you must first enter the business location in the **Business location** box on the **General** page on the UM auto attendant.
- **Announce business hours** Select this option to enable a caller to choose an auto attendant menu option and hear the hours of operation for the business that's configured on the UM auto attendant. To enable this to work correctly, you must first configure the business hours on the **Business hours** page on the UM auto attendant.

5. Click **OK** to create the new menu navigation.
6. On the **UM Auto Attendant** page, click **Save** to save your changes.

Use the Shell to configure UM auto attendant key mappings

This example enables business hours key mappings so that:

- When callers press 1, they will be forwarded to another UM auto attendant named `SalesAutoAttendant`.
- When they press 2, they will be forwarded to extension number 12345 for Support.
- When they press 3, they will be sent to another auto attendant that will play an audio file.

```
Set-UMAutoAttendant -id MyAutoAttendant -
BusinessHoursKeyMappingEnabled $true -
BusinessHoursKeyMapping
"1,Sales,,SalesAutoAttendant","2,Support,12345","3,Directio
ns,,,directions.wav"
```

This example sets key mappings defined in a comma-separated value (.csv) file. You must first create the .csv file with the following headings and the correct entry: <key>,<description>,

[<extension>],[<autoattendant name>],[<promptfilenamepath>],[<asrphrase1;asrphrase2>], [<leavevoicemailfor>],[<transfertomailbox>]. The values in brackets are optional. After creating the .csv file, import the .csv file using the **Import-csv** cmdlet.

```
$o = Import-csv -path "C:\UMFiles\AutoAttendants
\keymappings.csv"
Set-UMAutoAttendant MyAutoAttendant -
BusinessHoursKeyMapping $o
```

This example exports key mappings from an existing UM auto attendant into a .csv file, and then imports the same key mappings into another UM auto attendant. You could also export the key mappings to a .csv file, edit or modify the key mappings in the .csv file, and then import those key mappings into another UM auto attendant.

```
$aa = Get-UMAutoAttendant -id MyAutoAttendant
$aa1 = Get-UMAutoAttendant -id MyAutoAttendant2
$aa.BusinessHoursKeyMapping | Export-csv -path "C:\UMFiles
\AutoAttendants\keymappings.csv"
$aa1.BusinessHoursKeyMapping = (Import-csv -path "C:
\UMFiles\AutoAttendants\keymappings.csv")
```

Create business hours navigation menus

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

You can enable business hours key mappings for a Unified Messaging (UM) auto attendant. After you create a UM auto attendant, a default system prompt will be used for the business hours main menu prompt greeting that callers hear after the business hours welcome greeting is played. The default business hours main menu prompt says, "Welcome to the Microsoft Exchange auto attendant." Because no key mappings are defined by default, no menu options are available to callers, and they hear only the default main menu prompt.

When you configure key mappings, you define the options and the operations that will be performed if a caller speaks a phrase while they're using a speech-enabled auto attendant or presses a key on the telephone keypad while they're using an auto attendant that isn't speech-

enabled.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable business hours key mappings on a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to create a business hours navigation menu. On the toolbar, click **Edit** .
3. On the **UM Auto Attendant** page, click **Menu navigation**, under **Business hours menu navigation**, select **Enable business hours menu navigation**, and then click **Add +**.
4. On the **New menu navigation entry** page, use the following options to create a new navigation entry:
   - **Prompt** Use this box to type the name of the new navigation menu. The navigation menu name is used for display purposes only. This is a required field.

Because you may want to specify multiple new navigation menus, we recommend that you use meaningful names for your key mappings. The maximum length of the name for the key mapping is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

   - **When this key is pressed** Use this list to enable key mapping. The key mapping is the

number key that a caller presses to have the auto attendant perform a specific operation, for example, forwarding the caller to another auto attendant or to an operator. By default, no entries are defined.

Use the drop down list to select the numeric key (from 1 through 9) that the caller must press. Zero (0) is reserved for the auto attendant operator.

If you select **Time Out** from the drop down list, it enables callers to be transferred to an extension number or to another auto attendant if they don't press a key on the telephone keypad. For example, "Please stay on the line and your call will be answered by the next available representative." The default setting is 5 seconds. If you enable this option, a blank key mapping will be created.

- **Play the following audio file** Use this option to select a previously recorded audio file for callers. Click **Change**, and then click **Browse** to locate the audio file. If you leave the audio file as the default <None>, the Unified Messaging TTS (Text to Speech) engine will synthesize a business hours main menu prompt. Alternatively, you can create a customized audio file that can be used for the business hours main menu prompt for a speech-enabled auto attendant. For example, it might say, "To leave a voice message for sales, say 1. To leave a voice message for technical support, say 2. To leave a voice message for administration, say 3."
- **Perform this additional action** Select one of the following options to define the action that you want the auto attendant to perform for the caller:
  - **None** If you don’t want the auto attendant to transfer the call to an extension or to another auto attendant, or leave a message for a user, use this option.
  - **Transfer to this extension** Select this option to enable calls to be transferred to an extension number. If you enable this option, use the box to type the extension number where the call will be transferred. This field allows only numeric characters. It can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
  - **Transfer to this UM auto attendant** Select this option to transfer the call to an auto attendant. Click **Browse** to locate the auto attendant that you want to use. Before you enable this option, you must first create and configure the auto attendant. This option is used when you create a parent/child structure of UM auto attendants.
  - **Leave a voice message for this user** Select this option to enable a caller to leave a voice mail message for a user that's on the same dial plan as the UM auto attendant that you're configuring. When a caller chooses this option from an auto attendant menu, they'll be prompted to leave a voice message for the user that was selected. Click **Browse** to locate the UM-enabled user.
  - **Announce business location** Select this option to enable a caller to choose an auto attendant menu option and hear the location of the business that's configured on the UM auto attendant. To enable this to work correctly, you must first enter the business location in the **Business location** box on the **General** page on the UM auto attendant.
  - **Announce business hours** Select this option to enable a caller to choose an auto attendant menu option and hear the hours of operation for the business that's configured on the UM auto attendant. To enable this to work correctly, you must first configure the business hours on the **Business hours** page on the UM auto attendant.

5. Click **OK** to create the new menu navigation.
6. On the **UM Auto Attendant** page, click **Save** to save your changes.

Use the Shell to enable business hours key mappings on a UM auto attendant

This example configures a UM auto attendant named `MyAutoAttendant` and enables business hours key mappings so that when callers press 1, they're forwarded to another UM auto attendant named `SalesAutoAttendant`. When they press 2, they're forwarded to extension number 12345 for Support, and when they press 3, they're sent to another auto attendant that plays an audio file.

```
Set-UMAutoAttendant -Identity MyAutoAttendant -
BusinessHoursKeyMappingEnabled $true -
BusinessHoursKeyMapping
"1,Sales,,SalesAutoAttendant","2,Support,12345","3,Directio
ns,,,directions.wav"
```

Create non-business hours navigation menus

Automatically answer and route incoming calls > UM auto attendant procedures > Set up a UM auto attendant >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

You can enable non-business hours key mappings for a Unified Messaging (UM) auto attendant. After you create a UM auto attendant, a default system prompt will be used for the non-business hours main menu prompt greeting that callers hear after the non-business hours welcome greeting is played. The default non-business hours main menu prompt says, "Welcome to the Microsoft Exchange after hours auto attendant." Because no key mappings are defined by default, no menu options are available to callers and they hear only the default non-business hours main menu prompt.

When you configure key mappings, you define the options and the operations that will be performed if a caller speaks a phrase while they're using a speech-enabled auto attendant or presses a key on the telephone keypad while they're using an auto attendant that isn't speech-enabled.

For additional management tasks related to UM auto attendants, see UM auto attendant

procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable non-business hours key mappings on a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to create a non-business hours navigation menu. On the toolbar, click **Edit** .
3. On the **UM Auto Attendant** page, click **Menu navigation**, under **Non-business hours menu navigation**, select **Enable non-business hours menu navigation**, and then click **Add +**.
4. On the **New menu navigation entry** page, use the following options to create a new menu navigation entry:
   - **Prompt** Use this box to type the name of the new navigation menu. The navigation menu name is used for display purposes only. This is a required field.

Because you may want to specify multiple new navigation menus, we recommend that you use meaningful names for your key mappings. The maximum length of the name for the key mapping is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

   - **When this key is pressed** Use this list to enable key mapping. The key mapping is the number key that a caller presses to have the auto attendant perform a specific operation, for example, forwarding the caller to another auto attendant or to an operator. By default, no

entries are defined.

Use the drop down list to select the numeric key (from 1 through 9) that the caller must press. Zero (0) is reserved for the auto attendant operator.

If you select **Time Out** from the drop down list, it enables callers to be transferred to an extension number or to another auto attendant if they don't press a key on the telephone keypad. For example, "Please stay on the line and your call will be answered by the next available representative." The default setting is 5 seconds. If you enable this option, a blank key mapping will be created.

- **Play the following audio file** Use this option to select a previously recorded audio file for callers. Click **Change**, and then click **Browse** to locate the audio file. If you leave the audio file as the default <None>, the Unified Messaging TTS (Text to Speech) engine will synthesize a non-business hours main menu prompt. Alternatively, you can create a customized audio file that can be used for the non-business hours main menu prompt for a speech-enabled auto attendant that would say, for example, "You have reached Contoso during non-business hours. To leave a voice message for sales, say 1. To leave a voice message for technical support, say 2. To leave a voice message for administration, say 3. To reach an after hours operator, press zero."
- **Perform this additional action** Select one of the following options to define the action that you want the auto attendant to perform for the caller:
  - **None** If you don’t want the auto attendant to transfer the call to an extension or to another auto attendant, or leave a message for a user, use this option.
  - **Transfer to this extension** Select this option to enable calls to be transferred to an extension number. If you enable this option, use the box to type the extension number where the call will be transferred. This field allows only numeric characters. It can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
  - **Transfer to this UM auto attendant** Select this option to transfer the call to an existing auto attendant. Click **Browse** to locate the auto attendant that you want to use. Before you enable this option, you must first create and configure the auto attendant. This option is used when you create a parent/child structure of UM auto attendants.
  - **Leave a voice message for this user** Select this option to enable a caller to leave a voice mail message for a user that's on the same dial plan as the UM auto attendant that you're configuring. When a caller chooses this option from an auto attendant menu, they'll be prompted to leave a voice message for the user that was selected. Click **Browse** to locate the UM-enabled user.
  - **Announce business location** Select this option to enable a caller to choose an auto attendant menu option and hear the location of the business that's configured on the UM auto attendant. To enable this to work correctly, you must first enter the business location in the **Business location** box on the **General** page on the UM auto attendant.
  - **Announce business hours** Select this option to enable a caller to choose an auto attendant menu option and hear the hours of operation for the business that's configured on the UM auto attendant. To enable this to work correctly, you must first configure the business hours on the **Business hours** page on the UM auto attendant.

5. Click **OK** to create the new menu navigation.
6. On the **UM Auto Attendant** page, click **Save** to save your changes.

Use the Shell to enable non-business hours key mappings on a UM auto attendant

This example configures a UM auto attendant named `MyAutoAttendant` and enables non-business hours key mappings so that when callers say "After Hours" they will be forwarded to extension number 12345, and if they say "Directions" they will be forwarded to extension number 23456.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
AfterHoursKeyMappingEnabled $true -AfterHoursKeyMapping
"AfterhoursOperator,12345","Directions,23456"
```

Manage a UM auto attendant

Unified Messaging > Automatically answer and route incoming calls > UM auto attendant procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-30*

After you create a Unified Messaging (UM) auto attendant, you can view or configure a variety of settings. For example, you can add, remove, and edit extension numbers associated with the auto attendant. You can also enable or disable Automatic Speech Recognition (ASR) for the auto attendant and change the greetings used for business and non-business hours.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see

Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to view or configure UM auto attendant settings

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant you want to view or configure, and then on the toolbar, click **Edit** .
3. On the **UM Auto Attendant** page, click **General** to view display-only information about the UM auto attendant and to perform management tasks on the UM auto attendant, as follows:
   - **UM dial plan** This box displays the UM dial plan associated with the auto attendant. After you create an auto attendant, the dial plan associated with the auto attendant can't be changed. If you need to associate an auto attendant with a different dial plan, you must delete the dial plan and then associate the auto attendant with the correct dial plan after you re-create it.
   - **Name** This box shows the name that was assigned to the auto attendant when it was created. This is the name that will appear in the EAC.
   - **Status** This box shows whether the UM auto attendant is enabled or disabled. To enable or disable the auto attendant, close the **UM Auto Attendant** page and use the toolbar under **UM Auto Attendants** on the **UM Dial Plan** page.
   - **Access numbers** Use this box to enter an extension number or access number that leads callers to the auto attendant. By default, no extension or access numbers are configured when you create an auto attendant.

The number of digits in the extension numbers or access numbers you provide must match the number of digits for an extension number configured on the UM dial plan associated with the UM auto attendant. You can also add a Session Initiation Protocol (SIP) address to this box. A SIP address is used by some IP Private Branch eXchanges (PBXs), SIP-enabled PBXs, and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server.

You can create a new auto attendant without listing an extension number or access number. To add an extension number, type the number in this box, and then click **Add +**. You can associate more than one number with an auto attendant. You can also edit or remove an existing access number. To edit an existing number, select it and click **Edit** . To remove an existing extension number from the list, select it and click **Remove −**.

   - **Set the auto attendant to respond to voice commands** Select this check box to enable

callers to respond verbally to auto attendant prompts to navigate the menu system. By default, when an auto attendant is created, it isn't speech-enabled.

If you decide to create the UM auto attendant but not to speech-enable it, you can use the EAC or the Shell to speech-enable it after it is created.

- **Use this auto attendant when voice commands don’t work correctly** Click **Browse** to select the auto attendant that you want to use in the case that voice commands don’t work. This is also referred to as a DTMF fallback auto attendant. A DTMF fallback auto attendant can be used only if the **Set the auto attendant to respond to voice commands don’t work correctly** option is selected. You must first create a DTMF fallback auto attendant, and then click **Browse** to locate the appropriate DTMF auto attendant.

A DTMF fallback auto attendant is used when the UM speech-enabled auto attendant can't understand or recognize the speech inputs from the caller. If the DTMF auto attendant is used, the caller is required to use DTMF inputs to navigate the menu system, spell a user's name, or use a custom menu prompt. A caller won't be able to use voice commands to navigate this auto attendant.

If you don't configure a DTMF fallback auto attendant, we recommend that you configure an operator extension number on the auto attendant. If you don't configure an operator extension number, when callers use a speech-enabled auto attendant and the system doesn't recognize their voice inputs, they won't be able to navigate the system or be transferred to an operator for help.

Although not required, we recommend that you configure the DTMF fallback auto attendant to have the same configuration as the speech-enabled auto attendant. The DTMF fallback auto attendant shouldn't be speech-enabled.

- **Language for automated voice interface** Use this list to select the language that callers hear when they reach the auto attendant. The default language is determined when you install Microsoft Exchange. For on-premises and hybrid deployments, by default, U.S. English is used because the auto attendant uses the language setting on the UM dial plan. To have other language options available, you must install the UM language packs for the languages you want to include. For more information about how to install a UM language pack, see Install a UM language pack. For UM in Office 365, it’s not required that you install any additional UM language packs.

Although you can select a language other than the language selected on the UM dial plan associated with the auto attendant, we recommend that the language settings on the dial plan and the auto attendant match. If language settings don't match, when callers call an extension number defined on the dial plan, they will be presented with prompts in one language, and when they dial an extension number associated with an auto attendant, they will be presented with prompts in a different language.

- **Business name** Use this box to enter the name of the business. By default, no business name is entered. If you enter a business name in this box, a prompt with the business name will be played to callers instead of the default greeting.
- **Business location** Use this box to enter the location of the business. By default, no business location is entered. If you enter the location of the business in this box, the business location will be played for callers.

4. Use **Greetings** on the auto attendant to manage recorded greetings. You can select default greetings or previously recorded custom greetings for business hours and non-business hours. You can configure the following:
   - **Business hours greeting** This is the initial greeting that is played when a caller calls the auto attendant during your organization's business hours. By default, business hours are from 12:00 A.M. to 12:00 A.M. and no non-business hours are set. If you don't specify a custom greeting, a system prompt that says, "Welcome to the Exchange auto attendant" is played for callers. The business and non-business hours are configured on the auto attendant **Business hours**.

You may want to customize this greeting to represent your company, for example, "Thank you for calling Woodgrove Bank." You can configure a customized business hours greeting by clicking **Change** to select a previously recorded custom greeting file. The custom greeting must already have been recorded as a .wav or .wma file.

   - **Non-business hours greeting** This is the initial greeting played when a caller calls the auto attendant during your organization's non-business hours. By default, no non-business hours are configured. Therefore, there is no default non-business hours greeting. You can configure the business and non-business hours on the auto attendant **Business hours**.

You may want to customize this greeting to represent your company, for example, "Thank you for calling Woodgrove Bank but we are now closed." or "You have reached Contoso, Ltd. after business hours. Our business hours are from 8:00 A.M. until 5:00 P.M., Monday through Friday." You can configure a customized non-business hours greeting by clicking **Change** to select a previously recorded custom greeting file. The custom greeting must already have been recorded as a .wav or .wma file.

   - **Informational announcement** When enabled, this optional recording plays immediately after the business or non-business hours greeting. An informational announcement may state the organization's hours of operation, for example, "Our business hours are 8:30 A.M. to 5:30 P.M., Monday through Friday and 8:30 A.M. to 1:00 P.M. on Saturday." An informational announcement can also provide information required for compliance with company policy, for example, "Calls may be monitored for training purposes." If it's important that callers hear the whole informational announcement, it can be marked as uninterruptible, requiring the caller to listen to the whole announcement.

By default, there's no informational announcement configured on UM dial plans or auto attendants. If you enable an informational announcement and use a custom audio file specific to your organization, the **Allow announcement to be interrupted** option will be made available. The recordings must already have been recorded as .wav or .wma files. Click **Change** to locate a custom informational announcement file previously recorded.

Allow announcement to be interrupted Select this check box to enable the caller to interrupt the informational announcement. This should be enabled if you have long informational announcements. Callers may become frustrated if the informational announcement is long and they can't interrupt it to access the options provided by the auto attendant.

5. Use **Business hours** to determine the organization's open business hours. During business hours, callers hear the default business hours greeting or a customized greeting, and the business hours main menu prompt if the appropriate business hours key mappings are configured on **Menu**

navigation. You can configure the following:

- **Time zone** Use this list to select your time zone. Consider whether the dial plan associated with the auto attendant covers more than one time zone when you set your schedule.

For on-premises and hybrid deployments, by default, the time zone is configured using the local server's system time when the Mailbox server that is running the Microsoft Exchange Unified Messaging service was installed.

- **Business hours** Click **Configure business hours**, and then, on the **Configure Business Hours** page, use the grid to configure your organization's business hours.
- **Holiday schedule** Use this to define days, from 00:00 through 23:59 (12:00 A.M. through 11:59 P.M.), on which your organization will be closed for a holiday. Callers who reach the auto attendant during the times that you specify on the **New holiday** page hear a custom holiday greeting audio file that you define. When you configure the holiday schedule, you must define the holiday name, the audio file for the recorded holiday greeting, and the **Start date** and **End date**. The greetings must already have been recorded as .wav or .wma files.

6. Use **Menu navigation** to specify the menu options that are offered to callers during business and non-business hours. If you want to enable menu navigation, you must do it separately for business and non-business hours. For example, if you want to enable business hours navigation, you must add a menu prompt custom audio recording, select the **Enable business hours menu navigation** check box, click **Add +**, and then set the options on the **New menu navigation entry** page.
   - **Business hours menu navigation** This is the list of options that callers hear during the business hours that are defined on the **Business hours** page. For example, "For technical support, press or say 1. For corporate offices and administration, press or say 2. For sales, press or say 3."

To enable business hours menu navigation, you must perform the following steps:

a. **Menu prompt** Use this to specify a custom menu prompt audio file. To use a custom or previously recorded business hours menu prompt, click **Change**, and then click **Browse** to locate the menu prompt recording.

b. **Enable business hours menu navigation** Select this check box to enable options for menu navigation that will be used during business hours. When you enable business hours menu navigation, you can add new menu navigation entries for business hours.

c. Click **Add +** to create a new menu navigation entry. On the **New menu navigation entry** page, use the following options to create a new menu navigation entry:

- **Prompt** Use this box to type the name of the new navigation menu. The navigation menu name is used for display purposes only. This is a required field.

Because you may want to specify multiple new navigation menus, we recommend that you use meaningful names for your key mappings. The maximum length of the name for the key mapping is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

- **When this key is pressed** Use this list to enable key mapping. The key mapping is the number key that a caller presses to have the auto attendant perform a specific operation, for example, forwarding the caller to another auto attendant or to an operator. By

default, no entries are defined.

Use the drop down list to select the numeric key (from 1 through 9) that the caller must press. Zero (0) is reserved for the auto attendant operator.

If you select **Time Out** from the drop down list, it enables callers to be transferred to an extension number or to another auto attendant if they don't press a key on the telephone keypad. For example, "Please stay on the line and your call will be answered by the next available representative." The default setting is 5 seconds. If you enable this option, a blank key mapping will be created.

- **Play the following audio file** Use this option to select a previously recorded audio file for callers. Click **Change**, and then click **Browse** to locate the audio file. If you leave the audio file as the default <None>, the Unified Messaging TTS (Text to Speech) engine will synthesize a business hours main menu prompt. Alternatively, you can create a customized audio file that can be used for the business hours main menu prompt for a speech-enabled auto attendant. For example, it might say, "To leave a voice message for sales, say 1. To leave a voice message for technical support, say 2. To leave a voice message for administration, say 3."
- **Perform this additional action** Select one of the following options to define the action that you want the auto attendant to perform for the caller:
  - **None** If you don’t want the auto attendant to transfer the call to an extension or to another auto attendant, or leave a message for a user, use this option.
  - **Transfer to this extension** Select this option to enable calls to be transferred to an extension number. If you enable this option, use the box to type the extension where the call will be transferred. This field allows only numeric characters. It can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
  - **Transfer to this UM auto attendant** Select this option to transfer the call to an auto attendant. Click **Browse** to locate the auto attendant that you want to use. Before you enable this option, you must first create and configure the auto attendant. This option is used when you create a parent/child structure of UM auto attendants.
  - **Leave a voice message for this user** Select this option to enable a caller to leave a voice mail message for a user that's on the same dial plan as the UM auto attendant that you're configuring. When a caller chooses this option from an auto attendant menu, they'll be prompted to leave a voice message for the user that was selected. Click **Browse** to locate the UM-enabled user.
  - **Announce business location** Select this option to enable a caller to choose an auto attendant menu option and hear the location of the business that's configured on the UM auto attendant. To enable this to work correctly, you must first enter the business location in the **Business location** box on the **General** page on the UM auto attendant.
  - **Announce business hours** Select this option to enable a caller to choose an auto attendant menu option and hear the hours of operation for the business that's configured on the UM auto attendant. To enable this to work correctly, you must first configure the business hours on the **Business hours** page on the UM auto attendant.

- **Non-Business hours menu navigation** This is the list of options callers hear during the non-

business hours that are defined on the **Business hours** page. For example, "Your call is very important to us. However, you have reached Woodgrove Bank after normal business hours. If you want to leave a message, please press or say 1 and we will return your call as soon as possible."

To enable non-business hours menu navigation, you must perform the following steps:

a. **Menu prompt** Use this to specify a custom menu prompt audio file. To use a custom or previously recorded non-business hours menu prompt, click **Browse**.

b. **Enable non-business hours menu navigation** Select this check box to enable options for menu navigation that will be used during non-business hours. When you enable non-business hours menu navigation, you can add new menu navigation entries for non-business hours.

c. Click **Add +** to create a new menu navigation entry. On the **New menu navigation entry** page, use the following options to create a new menu navigation entry:

- **Prompt** Use this box to type the name of the new navigation menu. The navigation menu name is used for display purposes only. This is a required field.

Because you may want to specify multiple new navigation menus, we recommend that you use meaningful names for your key mappings. The maximum length of the name for the key mapping is 64 characters, and it can include spaces. However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

- **When this key is pressed** Use this list to enable key mapping. The key mapping is the number key that a caller presses to have the auto attendant perform a specific operation, for example, forwarding the caller to another auto attendant or to an operator. By default, no entries are defined.

Use the drop down list to select the numeric key (from 1 through 9) that the caller must press. Zero (0) is reserved for the auto attendant operator.

If you select **Time Out** from the drop down list, it enables callers to be transferred to an extension number or to another auto attendant if they don't press a key on the telephone keypad. For example, "Please stay on the line and your call will be answered by the next available representative." The default setting is 5 seconds. If you enable this option, a blank key mapping will be created.

- **Play the following audio file** Use this option to select a previously recorded audio file for callers. Click **Change**, and then click **Browse** to locate the audio file. If you leave the audio file as the default <None>, the Unified Messaging TTS (Text to Speech) engine will synthesize a non-business hours main menu prompt. Alternatively, you can create a customized audio file that can be used for the non-business hours main menu prompt for a speech-enabled auto attendant that would say, for example, "You have reached Contoso during non-business hours. To leave a voice message for sales, say 1. To leave a voice message for technical support, say 2. To leave a voice message for administration, say 3. To reach an after hours operator, press zero."
- **Perform this additional action** Select one of the following options to define the action that you want the auto attendant to perform for the caller:
  - **None** If you don’t want the auto attendant to transfer the call to an extension or to

another auto attendant, or leave a message for a user, use this option.

- **Transfer to this extension** Select this option to enable calls to be transferred to an extension number. If you enable this option, use the box to type the extension number where the call will be transferred. This field allows only numeric characters. It can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
- **Transfer to this UM auto attendant** Select this option to transfer the call to an existing auto attendant. Click **Browse** to locate the auto attendant that you want to use. Before you enable this option, you must first create and configure the auto attendant. This option is used when you create a parent/child structure of UM auto attendants.
- **Leave a voice message for this user** Select this option to enable a caller to leave a voice mail message for a user that's on the same dial plan as the UM auto attendant that you're configuring. When a caller chooses this option from an auto attendant menu, they'll be prompted to leave a voice message for the user that was selected. Click **Browse** to locate the UM-enabled user.
- **Announce business location** Select this option to enable a caller to choose an auto attendant menu option and hear the location of the business that's configured on the UM auto attendant. To enable this to work correctly, you must first enter the business location in the **Business location** box on the **General** page on the UM auto attendant.
- **Announce business hours** Select this option to enable a caller to choose an auto attendant menu option and hear the hours of operation for the business that's configured on the UM auto attendant. To enable this to work correctly, you must first configure the business hours on the **Business hours** page on the UM auto attendant.

7. Use **Address book and operator access** to define the features available to callers who dial in to the UM auto attendant. You can configure auto attendant features such as the language used when callers call in to the auto attendant and the ability for callers to transfer to an operator's extension number. You can configure the following:
   - **Options for contacting users** Use these options to determine how callers can contact users with voice mail when they call into a UM auto attendant
     a. **Allow callers to dial users** Select this check box to enable callers to transfer calls to users. By default, this option is enabled, and lets users who are associated with the dial plan transfer calls to users in the same UM dial plan. After you select this check box, you can set the group of users to whom callers can transfer by selecting the appropriate option under the **Options for searching the address book** section on this page.

If you disable this option and disable the **Allow callers to leave voice messages for users** option, the options under **Options for searching the address book** are also disabled.

     b. **Allow callers to leave voice messages for users** Select this check box to enable callers to send voice messages to users. By default, this option is enabled, and lets users who are associated with the dial plan send voice messages to users in the same UM dial plan. After you select this check box, you can set the group of users to whom callers can send voice messages by selecting the appropriate option under the **Options for searching the address book** section on this page.

If you disable this option and disable the **Allow callers to dial users** option, the options under

Options for searching the address book are also disabled.

If you disable this option, the auto attendant won't invite callers to send a voice message during a system prompt.

- **Options for searching the address book** Use these options to determine a grouping of users. By default, **Allow callers to search for user by name or alias** is selected, along with the **In this dial plan only** option. However, you can change the grouping of users to allow callers to transfer calls or send voice messages to users who are located in the global address list (GAL) for an organization. You can choose from the following:
  a. **Allow callers to search for users by name or alias** By default, this option is selected. It allows callers that call into this auto attendant to do a directory search for users by name or by their alias. An alias is assigned to a user when a mailbox is created for them. The alias is the first part of an SMTP address, for example, tonysmith@contoso.com. The SMTP address is tonysmith@contoso.com, while the alias is tonysmith. Choosing this option only affects callers that use this auto attendant and not those who use Outlook Voice Access.
     - **In this dial plan only** Select this option to allow callers who connect to the UM auto attendant to locate and contact users who are in the same dial plan that is associated with this UM auto attendant. By default, this option is enabled on the dial plan and on the auto attendant. This means that both Outlook Voice Access users and callers into the auto attendant are able to search for users within the same dial plan.
     - **In the entire organization** Select this option to allow callers who call into this UM auto attendant to search for and contact anyone listed in the GAL for the organization. This includes not only UM-enabled users but all users who are mailbox-enabled. This option allows callers to contact users in multiple dial plans. It isn't enabled by default. This setting is also available on a dial plan for Outlook Voice Access users.
  b. **Information to include for similar names** Use this drop-down list to select the option used for the UM auto attendant when users have the same or similar names. This setting is used when two or more users who have the same name exist in the directory. This is also called a matched name or disambiguation field. You can configure this setting, or you can leave the default setting on the auto attendant. By default, the auto attendant will inherit this setting from the setting on the dial plan that is linked to the auto attendant. The following is an example of a speech-enabled auto attendant:
     - System: "Welcome to Contoso. If you know the name of the person you are calling, please tell me their name at any time."
     - Caller says "Tony Smith."
     - There are multiple people with this name. Please select from one of the options: For Tony Smith, research, press 1. For Tony Smith, administration, press 2. For Tony Smith, technical support, press 3."
     - Caller presses the appropriate key on the key pad and the call is transferred to the user.

Note:
On a non-speech-enabled auto attendant, the system will tell the caller to use the key pad to input the user's name (last name first) and then search for the user. If there are multiple people in the directory with the same name, the caller is instructed to press the appropriate key to be

transferred to the user. You could optionally create a DTMF fallback auto attendant that uses only the key pad to enter a name or alias.

For these settings to be used, you must add the correct information to the user. For example, if you want the auto attendant to use a title for two users with the same name, you must add this information to the user's account. Select one of the following methods that provide more information to help the caller select the correct user in the organization:

- **Inherit From dial plan** Select this option to have the auto attendant use the default setting from the dial plan associated with the auto attendant.
- **Title** Select this option to have the auto attendant include each user's title when listing matches.
- **Department** Select this option to have the auto attendant include each user's department when listing matches.
- **Location** Select this option to have the auto attendant include each user's location when listing matches.
- **None** Select this option to have no additional information given when listing matches.
- **Prompt for alias** Select this option to have the auto attendant prompt the caller for the user's alias.

8. Under **Operator access**, you can specify auto attendant operator settings including the following:
   - **Operator extension** Use this box to type the extension number used to call an operator. This extension number can connect the caller to a human operator or a UM-enabled mailbox, or can be configured to call an external telephone number. By default, an operator extension isn't included in this box.
   - **Allow transfer to operator during business hours** Select this check box to enable callers to be transferred to a human operator during business hours by using the extension number that you configure in the **Operator extension** box. By default, this option is disabled.

It's useful to enable this option so that when a caller is unsuccessful at using the menu prompts or directory search to locate the required person during business hours, the caller can leave a voice message or connect to a human operator. After you enable this option, you can configure the operator extension number on a UM-enabled mailbox that's monitored. The caller can leave a voice message, or a human operator who has the extension number can help the caller.

   - **Allow transfer to operator during non-business hours** Select this check box to enable callers to be transferred to a human operator after business hours by using the extension number that you configure in the **Operator extension** box. By default, this option is disabled.

It's useful to enable this option so that when a caller is unsuccessful at using the menu prompts or directory search to locate the required person after business hours, the caller can leave a voice message or connect to a human operator. After you enable this option, you can configure the operator extension number configured on a UM-enabled mailbox that's monitored. The caller can leave a voice message, or a human operator who has the extension number can help the caller.

9. Use **Dialing authorization** to configure dialing rules for callers who call in to a UM auto attendant. You can use these settings to control the extension numbers that can be reached from an auto attendant or control the telephone numbers that can be dialed by callers that have

dialed into the auto attendant. You can configure the following:

- **Calls in the same UM dial plan** Select this check box to allow users who call in to an auto attendant to place or transfer calls to an extension number associated with a UM-enabled user who is associated with the same dial plan as the auto attendant. By default, this setting is enabled.

When you disable this setting, users who call in to an auto attendant can place or transfer calls to users who aren't UM-enabled or to other extension numbers not associated with a UM-enabled user. Users can't transfer calls to UM-enabled users who are associated with the same dial plan as the auto attendant. This is because the **Allow calls to any extension** setting is enabled by default.

- **Allow calls to any extension** When this setting is disabled, users who call in to an auto attendant can't place calls to users who aren't UM-enabled or to other extension numbers not associated with a UM-enabled user. However, they can place calls or transfer calls to extension numbers associated with UM-enabled users. This is because the **Calls in the same UM dial plan** setting is enabled by default. The **Allow calls to any extension** setting is enabled by default.

When this setting is enabled, users who call in to an auto attendant can place calls to users who aren't UM-enabled, to other extension numbers not associated with a UM-enabled user, and to UM-enabled users. This is because the **Calls within the same UM dial plan** setting is enabled by default.

You can enable this setting in an environment where not all users have been UM-enabled. This setting is also useful when you want to allow users who call in to a telephone number configured on an auto attendant to call extension numbers not associated with a UM-enabled user.

- **Authorized in-country/region dialing rule groups** Use this section to add or remove allowed in-country/region dialing rule groups. By default, there are no in-country/region dialing rule groups configured on UM auto attendants.

In-country/region dialing rule groups are used to allow or restrict the telephone numbers within a country or region that any user who has dialed in to the UM auto attendant can dial. This helps prevent unnecessary or unauthorized telephone calls and charges.

To add in-country/region dialing rule groups, you must first create the appropriate in-country/region dialing rule groups on the dial plan associated with the UM auto attendant, and then add the appropriate dialing rule group.

In-country/region dialing rule groups can be used by Unified Messaging to allow or restrict access to telephone numbers within a country or region. This is applied to any user who has called in to an auto attendant. For more information about outdialing, see Allow users to make calls.

- **Authorized international dialing rule groups** Use this section to add or remove allowed international dialing rule groups. By default, there are no international dialing rule groups configured on UM auto attendants.

International dialing rule groups are used to allow or restrict the telephone numbers outside a country or region that any user who has dialed in to the UM auto attendant can dial. This helps prevent unnecessary or unauthorized telephone calls and charges.

To add international dialing rule groups, you must first create the appropriate international dialing

rule groups on the dial plan associated with the UM auto attendant. After you create the required dialing rule groups on the dial plan, you must then add the dialing rule groups to the list of authorized dialing rule groups on the UM auto attendant.

International dialing rule groups can be used by Unified Messaging to allow or restrict access to telephone numbers outside a country or region. This is applied to any user who has called in to an auto attendant. For more information about outdialing, see Allow users to make calls.

10.Click **OK** to create the new menu navigation.

11.On the **UM Auto Attendant** page, click **Save** to save your changes.

Use the Shell to configure UM auto attendant properties

This example configures a UM auto attendant named `MySpeechEnabledAA` to fall back to the `MyDTMFAA` auto attendant, sets the operator's extension to 50100, and enables transfers to this extension number after business hours.

```
Set-UMAutoAttendant -Identity MySpeechEnabledAA -
DTMFFallbackAutoAttendant MyDTMFAA -OperatorExtension 50100
-AfterHoursTransferToOperatorEnabled $true
```

This example configures a UM auto attendant named `MyUMAutoAttendant` that has: Business hours configured as 10:45 to 13:15 (10:45 A.M. to 1:15 P.M.) on Sunday, 09:00 to 17:00 (9:00 A.M. to 5:00 P.M.) on Monday, and 09:00 to 16:30 (9:00 A.M. to 4:30 P.M.) on Saturday; holiday times and their associated greetings configured as "New Year" on January 2, 2013; and "Building Closed for Construction" configured from April 24 through April 28, 2013.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessHoursSchedule 0.10:45-0.13:15,1.09:00-
1.17:00,6.09:00-6.16:30 -HolidaySchedule "New
Year,newyrgrt.wav,1/2/2013","Building Closed for
Construction,construction.wav,4/24/2013,4/28/2013"
```

Use the Shell to view UM auto attendant properties

This example returns a formatted list of all UM auto attendants.

```
Get-UMAutoAttendant | Format-List
```

This example displays the properties of a UM auto attendant named MyUMAutoAttendant.

```
Get-UMAutoAttendant -Identity MyUMAutoAttendant
```

Configure a DTMF fallback auto attendant

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-30*

You can configure a speech-enabled Unified Messaging (UM) auto attendant that has a dual tone multi-frequency (DTMF) fallback auto attendant. A DTMF fallback auto attendant is used when the UM speech-enabled auto attendant can't understand or recognize the speech inputs provided by a caller. If a DTMF fallback auto attendant has been configured, the caller has to use DTMF inputs, also known as touchtone inputs, to navigate the auto attendant menu system, spell a user's name, or use a custom menu prompt. If no DTMF fallback auto attendant has been configured, and the maximum number of speech inputs is exceeded because the system didn't understand what the caller said, the system will respond with this prompt: "Sorry, I couldn't help. Please call back later."

By default, an auto attendant isn't speech-enabled when you create it. After you speech-enable the auto attendant, callers can use only voice commands to navigate the auto attendant menu system, and touchtone inputs can't be used. Although it isn't required, we recommend that you configure a DTMF fallback auto attendant for each speech-enabled auto attendant so callers can use touchtone inputs if the speech-enabled auto attendant doesn't recognize or understand the words they say. We also recommend that you don't speech-enable a DTMF fallback auto attendant.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure a speech-enabled auto attendant with a DTMF fallback auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change and click **Edit** ✎.
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to create a DTMF fallback auto attendant. On the toolbar, click **Edit** ✎.
3. On the **UM Auto Attendant** page > **General**, select the check box next to **Use this auto attendant when voice commands don't work correctly**, and then click **Browse**.
4. On the **Select a UM Auto Attendant** page, select the auto attendant you want to use as a DTMF fallback auto attendant, and then click **Save**.

Important:

You must first speech-enable the auto attendant before you can browse for a DTMF fallback auto attendant you have set up.

Use the Shell to configure a speech-enabled auto attendant with a DTMF fallback auto attendant

This example configures a UM auto attendant named `MySpeechEnabledAA` to use a DTMF fallback auto attendant named `MyDTMFAA`.

```
Set-UMAutoAttendant -Identity MySpeechEnabledAA -
DTMFFallbackAutoAttendant MyDTMFAA
```

Enable a UM auto attendant

Unified Messaging > Automatically answer and route incoming calls > UM auto attendant procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

By default, when a Unified Messaging (UM) auto attendant is created, its status is set to disabled.

After you create the UM auto attendant, you can change its status to enable it to answer incoming calls.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change and click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant you want to enable. On the toolbar, click the **Up arrow** .
3. On the **Warning** page, click **Yes**.

Use the Shell to enable a UM auto attendant

This example enables the UM auto attendant named `MyUMAutoAttendant` to answer incoming calls.

```
Enable-UMAutoAttendant -Identity MyUMAutoAttendant
```

Disable a UM auto attendant

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

By default, when a Unified Messaging (UM) auto attendant is created, its status is set to disabled. After you create the UM auto attendant, you can change its status to control whether it can answer incoming calls. For example, you might want to disable the UM auto attendant when you're recording or re-recording customized prompts and messages.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant. Also confirm that the status of the UM auto attendant is set to enabled.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to disable a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the dial plan you want to change, and on the toolbar, click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant you want to disable. On the toolbar, click **Down arrow**
3. On the **Warning** page, click **Yes**.

Use the Shell to disable a UM auto attendant

This example disables a UM auto attendant named `MyUMAutoAttendant`.

```
Disable-UMAutoAttendant -Identity MyUMAutoAttendant
```

Delete a UM auto attendant

Unified Messaging > Automatically answer and route incoming calls > UM auto attendant procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-30*

After you delete a Unified Messaging (UM) auto attendant, the incoming calls that were answered by the UM auto attendant must be answered by a human operator. A UM auto attendant can't be deleted if it's associated with a UM dial plan as the default UM auto attendant.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to delete a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to edit, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant you want to delete. On the toolbar, click **Delete** . On the **Warning** page, click **Yes**.

Use the Shell to delete a UM auto attendant

This example deletes a UM auto attendant named `MyUMAutoAttendant`.

```
Remove-UMAutoAttendant -Identity MyUMAutoAttendant
```

Enable or disable automatic speech recognition

Unified Messaging > Automatically answer and route incoming calls > UM auto attendant procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-12*

You can enable your Unified Messaging (UM) auto attendant for Automatic Speech Recognition (ASR). After you speech-enable a UM auto attendant, callers can respond verbally to auto attendant prompts and move through the menu system of the auto attendant. By default, an auto attendant isn't speech-enabled when you create it. After you speech-enable the auto attendant, callers can use only voice commands to navigate the auto attendant menu system, and touchtone inputs can't be used.

Although it isn't required, we recommend that you configure a dual tone multi-frequency (DTMF) fallback auto attendant for each speech-enabled auto attendant so callers can use touchtone inputs if the speech-enabled auto attendant doesn't recognize or understand the words they say. If a DTMF fallback auto attendant is configured, callers can use DTMF inputs, also known as touchtone inputs, to navigate the auto attendant menu system, spell a user's name, or use a custom menu prompt. We don't recommend that you speech-enable a DTMF fallback auto attendant.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For

detailed steps, see Create a UM auto attendant.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to speech-enable a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant you want to speech enable, and then click **Edit** .
3. On the **UM Auto Attendant** page > **General**, select the check box next to **Set the auto attendant to respond to voice commands** to enable speech recognition. To disable automatic speech recognition, clear this check box.
4. Click **Save**.

Use the Shell to speech-enable a UM auto attendant

This example enables ASR on a UM auto attendant named `MySpeechEnabled AA`.

```
Set-UMAutoAttendant -Identity MySpeechEnabledAA -
SpeechEnabled $true
```

Enable or prevent transferring calls from an auto attendant

Unified Messaging > Automatically answer and route incoming calls > UM auto attendant procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can enable callers to transfer calls to users through an auto attendant, or prevent them from doing so. By default this option is enabled, and lets callers transfer calls to UM-enabled users in the Unified Messaging (UM) dial plan that's associated with the UM auto attendant.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or prevent call transfers to users from a UM auto attendant

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to configure call transfer, and then click **Edit** .
3. On the **UM Auto Attendant** page > **Address book and operator access**, under **Options for contacting users**, select the check box next to **Allow callers to dial users** to enable calls to be transferred. To prevent call transfers, clear the check box.
4. Click **Save**.

Note:

If you clear this check box and also clear the **Allow callers to leave voice messages for users** check box, the **Options for searching the address book** are disabled.

Use the Shell to enable or prevent call transfers to users from a UM auto attendant

This example prevents call transfers on a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
AllowDialPlanSubscribers $false
```

This example enables call transfers on a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
AllowDialPlanSubscribers $true
```

Enable or disable sending voice messages to users

Unified Messaging > Automatically answer and route incoming calls > UM auto attendant procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-13*

You can enable callers to send voice messages to users from a Unified Messaging (UM) auto attendant, or prevent them from doing so. By default, this option is enabled and lets callers send voice messages to users in the UM dial plan that's associated with the UM auto attendant. If you disable this option, the auto attendant won't invite callers to send a voice message during a system prompt.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable callers to send voice messages or prevent them from doing so

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant you want to manage, and then click **Edit** .
3. On the **UM Auto Attendant** page > **Address book and operator access**, under **Options for contacting users**, select the check box next to **Allow callers to leave voice messages for users** to enable callers to leave voice messages. To prevent callers from leaving voice messages, clear the check box.
4. Click **Save**.

Note:

If you disable this option and also disable the **Allow callers to dial users** option, the **Options for searching the address book** are also disabled.

Use the Shell to enable callers to send voice messages or prevent them from doing so

This example prevents callers who call in to a UM auto attendant named `MyUMAutoAttendant` from sending voice messages.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
SendVoiceMsgEnabled $false
```

This example enables callers who call in to a UM auto attendant named `MyUMAutoAttendant` to send voice messages.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
SendVoiceMsgEnabled $true
```

Enable or disable directory lookups

Unified Messaging > Automatically answer and route incoming calls > UM auto attendant procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can enable directory lookups so that callers who call in to a Unified Messaging (UM) auto attendant can look up names in the directory using their telephone keypad but not be able to search the directory using voice inputs. This setting is enabled by default. If this setting is disabled, callers won't be able to search the directory for a specific person using touchtone or voice commands.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or disable directory lookups

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to enable or disable directory lookups, and then click **Edit** .
3. On the **UM Auto Attendant** page > **Address book and operator access**, under **Options for**

searching the address book, select the check box next to **Allow callers to search for users by name or alias** to enable callers to search for users. To disable callers from searching for users, clear this check box.

4. Click **Save**.

Use the Shell to enable or disable directory lookups

This example disables directory lookups on a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
NameLookupEnabled $false
```

Configure the group of users that can be contacted

Unified Messaging > Automatically answer and route incoming calls > UM auto attendant procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-09*

You can specify the group of users that callers can contact when calling into a Unified Messaging (UM) auto attendant. By default, callers can contact users within the same dial plan that's associated with the UM auto attendant. However, you can change the grouping of users to allow callers to transfer calls or send voice messages to users who are located in the organization's address book or to a specific set of users.

For additional management tasks related to UM auto attendants, see Manage a UM auto attendant.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the group of users that callers can contact

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant you want to configure, and then click **Edit** .
3. On the **UM Auto Attendant** page > **Address book and operator access**, under **Options for searching the address book**, choose from the following options:
   - **In this dial plan only** Select this option to allow callers who connect to the UM auto attendant to locate and contact users who are in the dial plan associated with the UM auto attendant.
   - **In the entire organization** Select this option to allow callers who connect to the UM auto attendant to locate and contact anyone listed in the organization's address book. This includes all users who are mailbox-enabled.
4. Click **Save**.

Use the Shell to configure the group of users that callers can contact

This example sets the scope of the users that callers can contact to all users in the organization's address book on a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
ContactScope GlobalAddressList
```

Configure an auto attendant for users who have similar names

Unified Messaging > Automatically answer and route incoming calls > UM auto attendant

procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-16*

You can configure the method to use for users with similar names on an auto attendant's **Address book and operator access** options, or you can leave the default setting on the auto attendant and configure this setting on the dial plan associated with the auto attendant. By default, an auto attendant can disambiguate between two or more users who have the same or similar names because the default setting on the auto attendant is **Inherit from dial plan**.

Note:

For the information that will be included for users with similar names to work correctly, you must provide the title, department, and location information for the recipients in your Microsoft Exchange organization.

For additional management tasks related to UM auto attendants, see UM auto attendant procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure a UM auto attendant for users with similar names

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant you want

to configure, and then click **Edit** .

3. On the **UM Auto Attendant** page, click **Address book and operator access**, and under **Information to include for users with the same name**, select one of the following:
   - **Title** The auto attendant will include each user's title when it lists matches.
   - **Department** The auto attendant will include each user's department when it lists matches.
   - **Location** The auto attendant will include each user's location when it lists matches.
   - **None** The auto attendant won't include any additional information when it lists matches.
   - **Prompt For alias** The auto attendant will prompt the caller for the user's alias.
   - **Inherit from dial plan** The auto attendant will use the default setting from the dial plan associated with the auto attendant.
4. Click **Save**.

Use the Shell to configure a UM auto attendant for users with similar names

This example sets the information to be included with users with similar names to Prompt for Alias for a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
MatchedNameSelectionMethod PromptForAlias
```

This example sets the information to be included with users with similar names to the title of the users, enables name lookups, and enables callers that dial into the auto attendant to press * to be presented with the Outlook Voice Access welcome greeting for a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
MatchedNameSelectionMethod Title -NameLookupEnabled $true -
StarOutToDialPlanEnabled $true
```

Set up voice mail for users

Exchange Server 2013 > Unified Messaging >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-13*

After you've connected your telephony network or integrated Microsoft Lync Server with Exchange Unified Messaging (UM) and created and configured the required UM components, you'll need to set up voice mail for your users.

When you're enabling users for voice mail, you'll need to link the user to a UM mailbox policy. UM mailbox policies are used to apply common settings to a group of UM- enabled users. These settings include PIN policies, outbound calling restrictions, text to send with messages, and other related settings. You can either use a default UM mailbox policy or create and customize a UM mailbox policy based on the needs of your organization.

Setting up voice mail for users

Before you enable users for UM, you must consider the type of dial plan to use, the extension numbers that will be used, and determine what PIN policies, Outlook Voice Access, and other features you'll allow users to have access to. For details, see Voice mail for users.

UM mailbox policies

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-15*

Unified Messaging (UM) mailbox policies are required when you enable users for Unified Messaging. You create UM mailbox policies to apply a common set of policies or security settings to a collection of voice mail users' mailboxes. UM mailbox policies are used to specify UM settings like the following:

- PIN policies
- Dialing restrictions
- Other general UM mailbox policy properties

For example, you can create a UM mailbox policy to increase the level of PIN security by reducing the maximum number of sign-in failures for a specific group of UM-enabled users, such as executives.

UM mailbox policies

At least one UM mailbox policy must have been created before you can enable users for Unified Messaging. You can create additional UM mailbox policies to apply a common set of settings for groups of users.

You create UM mailbox policies by using the Exchange Management Shell or the Exchange Administration Center (EAC). By default, a single UM mailbox policy is created every time you create a UM dial plan. The new UM mailbox policy is automatically associated with the UM dial plan, and part of the dial plan name is included in the display name of the UM mailbox policy. You can edit this default UM mailbox policy.

Multiple UM-enabled users can be linked to a single UM mailbox policy. However, the mailbox for each UM-enabled user must be linked to a single UM mailbox policy. This lets you control PIN security settings such as the minimum number of digits in a PIN or the maximum number of sign-in attempts for the UM-enabled users who are associated with the UM mailbox policy. You can also control message text settings or dialing restrictions for the same UM-enabled mailboxes.

UM mailbox policy procedures

Unified Messaging > Set up voice mail for users > UM mailbox policies >

Applies to: *Exchange Online*

Topic Last Modified: *2013-05-03*

Create a UM mailbox policy

Manage a UM mailbox policy

Delete a UM mailbox policy

Create a UM mailbox policy

Set up voice mail for users > UM mailbox policies > UM mailbox policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-08*

You can create a Unified Messaging (UM) mailbox policy to apply a common set of UM policy settings, such as PIN policy settings or dialing restrictions, to a collection of UM-enabled mailboxes. UM mailbox policies link a UM-enabled user with a UM dial plan and apply a common set of policies or security settings to a collection of UM-enabled mailboxes. UM mailbox policies are useful for applying and standardizing UM configuration settings for UM-enabled users.

By default, when a UM dial plan is created, a UM mailbox policy is also created. You may have to create additional UM mailbox policies or modify existing UM mailbox policies after you deploy Unified Messaging in your organization.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to create a UM mailbox policy

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, click **New +**.
3. On the **New UM mailbox policy** page, in the **Name** box, enter the name of the new UM mailbox policy.

Use this box to specify a unique name for the UM mailbox policy. This is a display name that appears in the EAC. If you need to change the display name of the UM mailbox policy after it's been created, you must first delete the existing UM mailbox policy, and then create another UM mailbox policy that has the appropriate name. You can't delete a UM mailbox policy if any UM-enabled users are associated with it.

The UM mailbox policy name is required, but it's used for display purposes only. Because your organization may use multiple UM mailbox policies, we recommend that you use meaningful names for your UM mailbox policies. The maximum length of a UM mailbox policy name is 64 characters, and it can include spaces. However, it cannot include any of the following characters: " / \ [ ] : ; | = , + * ? < >.

4. Click **Save** to save the new UM mailbox policy. When you save the UM mailbox policy, all of the default settings including PIN policies, voice mail features, and Protected Voice Mail settings are enabled. If you want to customize or change any default settings, use the **Set-UMMailbox** cmdlet to change the settings for the UM mailbox policy you just created.

Use the Shell to create a UM mailbox policy

This example creates a UM mailbox policy named `MyUMMailboxPolicy` associated with a UM dial plan named `MyUMDialPlan`.

```
New-UMMailboxPolicy -Name MyUMMailboxPolicy -UMDialPlan
```

Manage a UM mailbox policy

Set up voice mail for users > UM mailbox policies > UM mailbox policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

After you create a Unified Messaging (UM) mailbox policy, you can view and configure a variety of settings. For example, you can configure UM features like Voice Mail Preview or Play on Phone and other security-related options such as Protected Voice Mail and PIN policy settings.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to manage a UM mailbox policy

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
2. On the **UM dial plan** page, under **UM Mailbox Policies**, on the toolbar, click **Edit** .
   - Use **General** to view and configure settings for a UM mailbox policy. For example, you can view the dial plans associated with the UM mailbox policy or disable missed call notifications

for users who are associated with a specific UM mailbox policy. When you modify the settings on a UM mailbox policy, the settings are applied to all users who are associated with the UM mailbox policy. You can view or configure the following:

- **UM dial plan** Displays the name of the dial plan associated with the UM mailbox policy. This is the name of the dial plan displayed in the Shell.

When a new UM mailbox policy is created, it must be associated with a dial plan. After the UM mailbox policy is created and associated with a dial plan, the settings defined on the mailbox policy are applied to the users who are associated with the dial plan. By default, when you create a UM dial plan using the Shell, it will also create a UM mailbox policy.

- **Name** Type the name of the dial plan. A UM dial plan name is required and must be unique. However, it's used only for display in the EAC and the Shell. If you have to change the display name of the dial plan after it's been created, you must first delete the existing UM dial plan and then create another dial plan that has the appropriate name. If your organization uses multiple UM dial plans, we recommend that you use meaningful names for your UM dial plans. The maximum length of a UM dial plan name is 64 characters, and it can include spaces. (If you're integrating with Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server it's not recommended that you use spaces.) However, it can't include any of the following characters: " / \ [ ] : ; | = , + * ? < >.
- **Limit on personal greetings (minutes)** Use this text box to enter the maximum number of minutes that users who are associated with the UM mailbox policy can use when they record their voice mail greeting. You can modify this setting after the UM mailbox policy is created. Only numeric characters are allowed. The valid range for the greeting is from 1 through 10 minutes. The default setting is 5 minutes.
- **Allow voice mail preview** Select or clear this check box to enable or disable the Voice Mail Preview feature for users associated with the UM mailbox policy. Enabling this setting allows users to receive the text of a voice mail message in the message body of an email or text message. The default setting is enabled.
- **Allow users to configure call answering rules** Select this check box to allow users who are associated with the UM mailbox policy to create call answering rules. If this option is disabled on the UM dial plan, this feature won't be available to UM-enabled users associated with the UM mailbox policy. The default setting is enabled.
- **Allow message waiting indicator** Select or clear this check box to enable or disable Message Waiting Indicator for users associated with the UM mailbox policy. Message Waiting Indicator is a feature found in most legacy voice mail systems. In its most common form, it lights a lamp on the voice mail user's phone to indicate the presence of a new voice message. Message Waiting Indicator can also send a text message to the UM-enabled user's mobile phone. The default setting is enabled.
- **Allow Outlook Voice Access** Select or clear this check box to enable or disable access to Outlook Voice Access for UM-enabled users who are associated with this UM mailbox policy. Outlook Voice Access is a feature used by UM-enabled users to access their mailbox over a phone. By default, this setting is enabled.
- **Allow missed call notifications** Select or clear this check box to enable or disable missed

call notifications for users associated with the UM mailbox policy. A missed call notification is an email message sent to a user's mailbox when the user doesn't answer an incoming call. This is a different email message than the email message that contains the voice message left for a user.

Note:

When you're integrating Unified Messaging and Lync Server on-premises, missed call notifications aren't available to users who have a mailbox located on an Exchange 2007 or Exchange 2010 Mailbox server. A missed call notification is generated when a user disconnects before the call is sent to Unified Messaging.

Typically, when a user misses an incoming call, the user receives two email messages: a message that contains the voice message and a missed call notification message. By default, missed call notifications are enabled when a UM mailbox policy is created.

- **Allow Play on Phone for voice mail** Select or clear this check box to enable or disable the Play on Phone feature for users associated with the UM mailbox policy. This option is enabled by default and allows users to play their voice messages over any phone, including an office or mobile phone.
- **Allow inbound faxes** Select or clear this check box to enable or disable inbound faxes for users associated with the UM mailbox policy. By default, when you enable users for UM, their mailbox is able to receive faxes. However, if this option is disabled on the UM dial plan, UM-enabled users associated with the UM mailbox policy won't be able to receive faxes. The default setting on the UM mailbox policy is disabled.

After you have enabled the **Allow inbound faxes** setting, you will need to specify the URI for the partner fax server. If the UM mailbox policy is linked to a dial plan that can use TCP and TLS, you will need to enter URIs for both TCP and TLS.

- **Help Microsoft improve voice mail preview** These options allow Microsoft to improve the quality of Voice Mail Preview. You can enable the following settings:
  - **Allow analysis of voice messages left by callers** Use this option to help improve the quality of Voice Mail Preview in future releases of Microsoft Exchange by forwarding copies of voice messages to Microsoft for analysis. You can't set this option if all voice messages are protected.
  - **Tell callers that voice messages may be analyzed** Use this option to tell callers that the messages they leave may be analyzed by Microsoft to improve the quality of Voice Mail Preview, and allow them to opt out.

- Use **Message Text** to configure message text settings for users who are associated with a UM mailbox policy. For example, you can specify the email message text sent to users after they reset their UM PIN. You can configure the following:
  - **When a user is enabled for Unified Messaging** The text entered in this text box appears in the email message sent to users when they are enabled for UM. When a recipient's mailbox is enabled for UM and they are enabled for voice mail, an email message that welcomes the user to Unified Messaging is sent to the user. This text box is limited to 512 characters and can contain simple HTML formatting. By default, no text is defined in this text box.

This welcome message contains welcome text and the PIN information that the user will use to access the UM or voice mail system. The text entered in this text box is included at the bottom of this welcome message. You can use this text box to include information such as the voice mail technical support telephone numbers or Outlook Voice Access numbers.

If text isn't entered in this text box, the default text generated by the UM or voice mail system is included in the email message.

The text that you provide in this text box can be plain. It can also contain simple HTML formatting tags if you want to emphasize text or add hyperlinks to other content.

Example 1: If you have any questions or suggestions about voice mail service, please call the help desk at extension 4200.

Example 2: If you have any questions or suggestions about voice mail service, please call the help desk at extension 4200 or visit our website at .

- **When a user's Outlook Voice Access PIN is reset** The text entered in this text box is included in the email message sent to UM-enabled users when their UM PIN is reset.

A PIN is reset by the UM or voice mail system if the number of failed sign-in attempts exceeds 10 (by default) or if users reset their PIN using the UM features included with Microsoft Outlook, Outlook Web App, or Outlook Voice Access from a telephone. You can use this text box to include information such as security notices or other security-related information in the email message.

If text isn't entered in this text box, the default text generated by the UM system is included in the email message.

This text box is limited to 512 characters. By default, no text is defined in this text box.

The text that you provide in this text box can be plain. It can also contain simple HTML formatting tags if you want to emphasize text or add hyperlinks to other content.

- **When a user receives a voice message** The text entered in this text box is included in the email message sent to users when they receive a voice message from an incoming caller. For example, this text can include disclaimers that contain information about forwarding voice messages or system security policies that describe the correct way to handle voice messages in your organization.

If text isn't entered in this text box, the default text generated by the system is included in the email message. This text box is limited to 512 characters. By default, no text is defined in this text box.

The text that you provide in this text box can be plain. It can also contain simple HTML formatting tags if you want to emphasize text or add hyperlinks to other content.

- **When a user receives a fax message** The text entered in this text box is included in the email message sent to users when they receive an incoming fax message in their Inbox. You can use this text box to include disclaimers that contain information about forwarding fax messages or other system security policies about the correct way to handle fax messages in your organization.

If text isn't entered in this text box, the default text generated by the system is included in the email message. This text box is limited to 512 characters. By default, no text is defined in this text box.

- Use **PIN Policies** to configure PIN settings for users who are associated with a UM mailbox

policy. UM PINs enable users to access their Inboxes by using a telephone. By configuring settings on this page, you can specify the minimum number of digits for a UM PIN or the number of failed sign-in attempts before users are locked out of their UM mailbox.

Make sure that you plan carefully for the UM PIN policies that you implement in your environment. If you don't plan and implement the appropriate UM PIN policies, you may introduce security threats and mistakenly allow unauthorized access to your network. You can configure the following:

- **Minimum PIN length (digits)** Use this text box to specify the minimum number of digits that a UM user's PIN can contain. The default setting is six digits. The range is from 4 through 24 numeric digits. This setting can't be disabled.

Increasing the number of digits required for a PIN increases the level of security for your UM system. Decreasing the number of digits required for a PIN reduces the level of security for your network. The fewer the digits that are required in a PIN, the easier it is for a potential attacker to guess a user's PIN.

If this setting is set too high, users might have problems remembering their PINs. However, if the setting is too low, you risk unauthorized access to the UM system.

- **PIN recycle count** Use this setting to set the number of unique PINs that users must use before they can reuse an old PIN. For most organizations, this value should be set to the default of 5, the number of PINs that the system will remember. PIN history can't be disabled.

You can set this value from 1 through 20. Setting this value too high can frustrate users because it can be difficult to memorize many PINs. Setting it too low may introduce a security threat to your network.

- **Allow common PIN patterns** Use this setting to set PIN complexity requirements for UM. These complexity requirements are enforced on PIN changes or when new PINs are created.

If this option is disabled, sequential and repeated numbers and the suffix of the mailbox extension will be rejected. If this option is enabled, only the suffix of the mailbox extension will be rejected.

As a security best practice, we recommend that you disable this setting. If this setting is disabled, user PINs can't contain the following:

Sequential numbers, such as 123456 or 456789.

Repeated numbers, such as 111111 or 8888888.

Suffix of the mailbox extension.

- **Enforce PIN lifetime (days)** Use this text box to configure the number of days until the UM-enabled user's PIN expires. After the PIN expires, the user must create a new UM PIN. For most organizations, this value should be set to the default of 60 days.

The value of this setting can be from 0 through 999. If it's set to 0, PINs never expire. Setting this value too low can frustrate users because they are required to create and memorize new PINS too frequently.

- **Number of sign-in failures before PIN reset** Use this text box to enter the number of sequential unsuccessful or failed sign-in attempts that can occur before the UM system automatically resets a user's PIN. For most organizations, this value should be set to the default of 5 attempts.

The value of this setting can be from 0 through 999. If it's set to 0, this setting is disabled and the system won't automatically reset users' PINs. Setting this value too low can frustrate users; setting it too high gives malicious users more attempts to determine the PIN.

This setting must be set to a number lower than the number configured in the **Number of sign-in failures before lockout** setting. This setting is designed to help prevent a brute force attack on user PINs.

- **Number of sign-in failures before lockout** Use this text box to enter the maximum number of sequential unsuccessful or failed sign-in attempts before users are locked out of their mailboxes.

For example, if a user tries to sign in to the mailbox unsuccessfully five times, based on the **Number of sign-in failures before PIN reset** setting, the system will reset the user's PIN. If the user tries to use the new PIN five more times unsuccessfully, the system will again reset the PIN. If the user tries to use this new PIN five more times unsuccessfully, the user is then locked out of the mailbox. After a user is locked out, an administrator must manually reset or unlock the mailbox for the user.

This value can be set from 1 through 999. Setting this value too low can frustrate users; setting it too high gives malicious users more attempts to determine the PIN. For most organizations, this value should be set to the default of 15 attempts.

This number must be greater than the number set in the **Number of sign-in failures before PIN reset** setting. This setting is designed to help prevent a brute force attack on user PINs.

- Use **Dialing authorization** to configure dialing rules for UM-enabled users who are associated with this UM mailbox policy.

You can use these settings to control the extension numbers that can be reached or the telephone numbers that can be dialed by UM-enabled users who are associated with the UM mailbox policy. You can configure the following:

- **Calls in the same UM dial plan** Select this check box to allow UM-enabled users who call in to a subscriber access number configured on a dial plan and successfully sign in to their mailbox to place calls or transfer to UM-enabled users who have extension numbers within the same dial plan. By default, this setting is enabled.

When you disable this setting, UM-enabled users who call in to a subscriber access number configured on a dial plan and successfully sign in to their mailbox can place calls or transfer calls to users who aren't UM-enabled or to other extension numbers not associated with a UM-enabled user. However, they can't transfer to UM-enabled users who are within the same dial plan. This is because the **Calls to any extension** setting is enabled by default.

- **Calls to any extension** When this setting is enabled, users who call in to a subscriber access number configured on a dial plan and successfully sign in to their mailbox can place calls to users who aren't UM-enabled, to other extension numbers not associated with a UM-enabled user, and to UM-enabled users within the same dial plan. This is because the **Calls in the same UM dial plan** setting is enabled by default.

When this setting is disabled, users who call in to an Outlook Voice Access number configured on a dial plan and successfully sign in to their mailbox can't place calls to users who aren't UM-enabled or to other extension numbers not associated with a UM-enabled user. However, they can place

calls or transfer calls to extension numbers associated with UM-enabled users. This is because the **Calls in the same UM dial plan** setting is enabled by default. The **Calls to any extension** setting is enabled by default.

You can enable this setting in an environment where not all users have been UM-enabled. This setting is also useful when you want to allow users who call in to an Outlook Voice Access number configured on a dial plan to call extension numbers not associated with a UM-enabled user.

- **Authorized in-country/region dialing rule groups** Use this section to add or remove allowed in-country/region dialing rule groups. By default, there are no in-country/region dialing rule groups configured on UM mailbox policies.

In-country/region dialing rule groups are used to allow or restrict the telephone numbers within a country or region that Outlook Voice Access users can dial. This helps prevent unnecessary or unauthorized telephone calls and charges.

To add in-country/region dialing rule groups, you must first create the appropriate in-country/region dialing rule groups on the dial plan associated with the UM mailbox policy, and then add the appropriate dialing rule entries on the dialing rule group. After you create the required dialing rule groups on the dial plan, you must then add the dialing rule groups to the list of dialing restrictions under **Dialing authorization** on the UM mailbox policy.

In-country/region dialing rule groups can be used to enable Unified Messaging to allow or restrict access to telephone numbers within a country or region. This is applied to Outlook Voice Access users who have called in to an Outlook Voice Access number.

- **Authorized international dialing rule groups** Use this section to add or remove allowed international dialing rule groups. By default, there are no international dialing rule groups configured on UM mailbox policies.

To add international dialing rule groups, you must first create the appropriate international dialing rule groups on the dial plan associated with the UM mailbox policy, and then add the appropriate dialing rule entries on the dialing rule group. After you create the required dialing rule groups, you must add the dialing rule groups to the dialing restrictions on the UM mailbox policy.

International dialing rule groups can be used to enable Unified Messaging to allow or restrict access to telephone numbers outside a country or region. This is applied to Outlook Voice Access users who have called in to a Outlook Voice Access number.

International dialing rule groups are used to allow or restrict the telephone numbers outside a country or region that Outlook Voice Access users can dial. This helps prevent unnecessary or unauthorized telephone calls and charges.

- Use **Protected Voice Mail** to configure the following settings:
  - **Protect voice messages from unauthenticated callers** Select one of the following options from the drop-down list to determine whether an incoming call answered by Unified Messaging will protect voice messages. This setting applies to voice messages sent to UM-enabled users when they don't answer their phone. This setting also applies to voice messages sent directly to UM-enabled users when the caller uses a UM auto attendant. You can configure the following:

None Use this setting to not have protection applied to any voice messages sent to UM-enabled

users.

Private Use this setting when you want to apply protection only to voice messages that have been marked as private by the caller.

All Use this setting when you want to apply protection to all voice messages, including those not marked as private.

- **Protect voice messages from authenticated callers** Select one of the following options from the drop-down list to determine whether an incoming call answered by Unified Messaging will protect voice messages. This setting applies to voice messages sent to UM-enabled users when they don't answer their phone. This setting also applies when callers sign in to their mailbox using Outlook Voice Access, and then create and send a voice message. You can configure the following:

None Use this setting to not have protection applied to any voice messages sent to UM-enabled users.

Private Use this setting when you want to apply protection only to voice messages that have been marked as private by the caller.

All Use this setting when you want to apply protection to all voice messages, including those not marked as private.

- **Require Play on Phone for protected voice messages** Select this check box if you want to force users who receive protected voice messages to use the Play on Phone feature. Or, if the client software doesn't support rights management, users must use Outlook Voice Access. The Play on Phone feature only applies to clients using a version of Outlook that supports rights management. For Outlook 2007 and earlier versions that don't support rights management, and for Outlook Web App clients, Outlook Voice Access is the only way that users can listen to protected voice mail.

The default setting requires all users associated with the UM mailbox policy to use the Play on Phone feature to listen to voice messages that are protected. By doing this, it prevents other people from hearing the voice message from a media player over computer speakers or from a media player on a mobile phone. Even if this is enabled, a UM-enabled user can still use Outlook Voice Access to hear the protected voice mail.

This is especially useful when UM-enabled users use public computers, laptops in public places, or their mobile phone's media player to listen to protected voice mail that can contain private information.

- **Allow voice responses to email and calendar items** Use this option to allow UM-enabled users to send voice responses to protected voice mail messages. The default is enabled. If you disable this, if a UM-enabled user receives a protected voice mail message, they will not be able to use Outlook Voice Access to reply to email and calendar items.
- **Message to send to users who don’t have Windows Rights Management support** Protected voice mail can only be accessed by email clients that support Information Rights Management (IRM), or if a UM-enabled user uses Outlook Voice Access to access the protected voice mail message.

If a protected voice mail message is sent to an email client that doesn't support IRM, the text that

you include in this box will be sent to the user in an email message. This information should include instructions about what to do to be able to receive the protected voice mail message.

Use the Shell to manage a UM mailbox policy

This example sets the PIN settings for users who are associated with a UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
LogonFailuresBeforePINReset 8 -MaxLogonAttempts 12 -
MinPINLength 8 -PINHistoryCount 10 -PINLifetime 60 -
ResetPINText "The PIN that is used to allow you access to
your mailbox using Outlook Voice Access has been reset."
```

This example selects the in-country or region groups and international groups from those configured on the UM dial plan associated with the UM mailbox policy. UM-enabled users associated with this UM mailbox policy will be able to place outbound calls according to the rules defined on these groups.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowDialPlanSubscribers $true -
AllowedInCountryOrRegionGroups InCountry/
RegionGroup1,InCountry/RegionGroup2 -
AllowedInternationalGroups
InternationalGroup1,InternationalGroup2 -AllowExtensions
$true
```

This example configures the text of voice messages sent to UM-enabled users and the text included in an email message sent to a user who has been UM-enabled.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
UMEnabledText "You have been enabled for Unified
Messaging." -VoiceMailText "You have received a voice
message from Microsoft Exchange 2013 Unified Messaging."
```

Use the Shell to view UM mailbox policy properties

This example returns a formatted list of all UM mailbox policies in the Active Directory forest.

```
Get-UMMailboxPolicy | Format-List
```

This example returns the properties and values for a UM mailbox policy named `MyUMMailboxPolicy`.

```
Get-UMMailboxPolicy -Identity MyUMMailboxPolicy
```

Delete a UM mailbox policy

Set up voice mail for users > UM mailbox policies > UM mailbox policy procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

When you delete a Unified Messaging (UM) mailbox policy, the UM mailbox policy will no longer be available to be associated with recipients who are being enabled for UM. You can't delete a UM mailbox policy if it's referenced by any UM-enabled mailboxes, and you can't delete a UM dial plan if a UM mailbox policy is associated with it.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to delete a UM mailbox policy

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
2. On the **UM dial plan** page, under **UM Mailbox Policies**, on the toolbar, click **Delete** .

Use the Shell to delete a UM mailbox policy

This example deletes a UM mailbox policy named `MyUMMailboxPolicy`.

```
Remove-UMMailboxPolicy -Identity MyUMMailboxPolicy
```

Voice mail for users

Exchange Server 2013 > Unified Messaging > Set up voice mail for users >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-19*

With Unified Messaging (UM), users in an Exchange organization can receive all their email and voice messages in one mailbox. The Unified Messaging functionality and voice mail features increase user productivity and enable more flexible messaging throughout an organization.

When you're adding a user to your organization, you're given the option of creating a mailbox or connecting the user to an existing mailbox. After the mailbox is created for the user or the user is connected to an existing mailbox, you can enable the mailbox for Unified Messaging so the user can use the voice mail system and the features included with voice mail. After the user is enabled for Unified Messaging, all email, voice mail, and fax messages will be delivered to the user's mailbox. By using Microsoft Office Outlook 2007 or later versions, Outlook Web App, a mobile phone enabled for Microsoft Exchange ActiveSync, or a regular or mobile phone, users can access their email, voice messages, personal contacts, and calendaring information.

Contents

Voice mail user properties

A user must have a mailbox before they can be enabled for Unified Messaging. But, by default, a user who has a mailbox isn't enabled for Unified Messaging. After the user is UM-enabled, you can manage, modify, and configure the UM properties and voice mail features for them. You can enable a user for Unified Messaging using EAC or the Shell. For details, see Enable a user for voice mail. To enable multiple UM users, use the EAC or the **Enable-UMMailbox** cmdlet in the Exchange

Management Shell.

The relationship between a voice mail user and other UM components

When you enable a user for Unified Messaging, the user must be associated with or linked to an existing UM mailbox policy, and you must provide an extension number for them. You can associate a user with a UM mailbox policy by using the **Enable-UMMailbox** cmdlet in the Shell or by selecting the UM mailbox policy when you enable the user for Unified Messaging. By default, when you create a UM dial plan, a new UM mailbox policy is created. This policy can be modified or another policy can be created and linked to the dial plan to determine what features or settings will be applied to a user or group of users.

A UM mailbox policy contains settings such as the dialing restrictions and PIN policies for a user. When a UM mailbox policy is created, it must be associated with only one UM dial plan. Any Exchange server can answering incoming calls and provide voice mail services for any UM-enabled users who are linked with the UM dial plan. After the user is enabled for Unified Messaging, the settings from a UM mailbox policy are applied to the UM-enabled user.

Extension numbers and SIP addresses

When you enable a user for Unified Messaging, you must define at least one extension number that Unified Messaging will use when voice mail is submitted to the user's mailbox. After you enable the user for Unified Messaging, you can add secondary extension numbers to the user's mailbox, or modify or remove them by configuring the Exchange Unified Messaging proxy address (EUM proxy address) on the user's mailbox or add or remove additional or secondary extensions for the user in the EAC. You can remove the primary extension number in the EAC by removing the EUM proxy address, but it’s recommended that you don’t remove it. Removing the primary extension number won’t allow calls to be forwarded correctly to the user’s mailbox.

Note:
There's no limit to the number of secondary extension numbers that you can add for a UM-enabled user but there can only be one primary extension number per user.

The mailbox of a UM-enabled user can be associated with only one UM dial plan. The UM-enabled user can be assigned the following:

- A single primary extension number, Session Initiation Protocol (SIP) address, or E.164 address on a single dial plan.
- Multiple secondary extension numbers, SIP addresses, or E.164 addresses on a single dial plan.
- Multiple primary extension numbers, SIP addresses, or E.164 addresses on two separate dial plans.

Note:
Each extension number, SIP address, and E.164 number must be unique within a dial plan and

the number of digits in the dial plan will used for all users that are linked with the dial plan.

For example, a UM-enabled user travels frequently from New York to Tokyo. The user's mailbox is associated with the New York dial plan and a single extension number is configured on the user's mailbox. A second extension number is configured on the user's mailbox for the Tokyo dial plan. When callers dial either extension number and leave a voice message for the user, the voice message will be delivered to the same UM-enabled mailbox.

Return to top

Using the EAC to enable a user for UM and voice mail

After you create an Exchange mailbox for the user, you can configure the UM mailbox settings by using **View Details** under **Unified Messaging** in the EAC. When you enable a user, there are several settings that you need to configure:

1. **SIP address** This is the SIP address for the user. You'll see this setting if the user that you're enabling for UM is assigned to a UM mailbox policy that's linked to a SIP URI dial plan. SIP URI dial plans are used when you're integrating Office Communications Server 2007 R2 or Microsoft Lync Server. When you assign the user to a UM mailbox policy that's linked to a SIP URI or E.164 dial plan, you must still also enter an extension number for the user. The primary extension number is used by the user to access Outlook Voice Access.
2. **Extension number** You must manually enter the extension number for the user you're enabling for UM.

You must provide a valid extension number for the user and match the number of digits specified on the dial plan. You can only enter numeric characters or digits from 1 through 20. The typical extension number is 3 to 7 digits long, and is configured on the dial plan with which the UM mailbox policy is linked and assigned to the user.

3. **PIN settings for the user:**
   - **Automatically generate PIN** This setting automatically generates a PIN for the UM-enabled user to use for voice mail access via Outlook Voice Access. This is the default setting. When you click this button, a PIN is automatically generated based on the PIN policies configured on the UM mailbox policy assigned to the user. We recommend that you use this setting to help protect the user's PIN. The PIN is sent to the user in the welcome message they receive after they're enabled for UM. By default, they'll have to change this PIN when they first sign in to their mailbox to get their voice mail.
   - **Type a PIN** This setting enables you to manually specify a PIN that the user will use to access the voice mail system.

The PIN must comply with the PIN policy settings configured on the UM mailbox policy associated with this UM-enabled user. For example, if the UM mailbox policy is configured to accept only PINs that contain seven or more digits, the PIN you enter in this box must be at least seven digits long.

   - **Require the user to reset their PIN the first time they sign in** This setting forces the user to reset their voice mail PIN when they access the voice mail system from a telephone using

Outlook Voice Access for the first time. They will be prompted to enter a PIN that's more familiar to them.It's a security best practice to force UM-enabled users to change their PIN when they first sign in to help protect against unauthorized access to their data and Inbox. This check box is selected by default.

Using the Shell to enable a user for UM and voice mail

This example enables Unified Messaging and voice mail on the mailbox for tonysmith@contoso.com, sets the extension and manually sets the PIN for the user, and then assigns the user to a UM mailbox policy named `MyUMMailboxPolicy`.

```
Enable-UMMailbox -Identity tonysmith@contoso.com -
UMMailboxPolicy MyUMMailboxPolicy -Extensions 51234 -PIN
5643892 -PINExpired $true
```

This example enables Unified Messaging and voice mail on a mailbox for tonysmith@contoso.com, assigns the user to a UM mailbox policy named `MyUMMailboxPolicy`, and sets the extension number, SIP address, and manually sets the PIN for the user.

```
Enable-UMMailbox -Identity tonysmith@contoso.com -
UMMailboxPolicy MyUMMailboxPolicy -Extensions 51234 -PIN
5643892 -SIPResourceIdentifier "tonysmith@contoso.com" -
PINExpired $true
```

Disabling UM for a user

When you disable Unified Messaging for a user, the user's account may still be listed when a caller performs a directory search using a UM auto attendant menu or using Outlook Voice Access. Callers may be able to locate a user in the directory, but when they try to contact the user, they're taken back to the main menu in Unified Messaging. This may cause callers to become frustrated with the system. You can prevent callers from using a directory search to contact a user who's been disabled for Unified Messaging by connecting the user to another voice mail system, removing the user from the UM auto attendant directory search, or removing the user's account.

After a UM-enabled user account is disabled for Unified Messaging, the user may still have access to the individual UM-enabled mailbox using Outlook Voice Access or Microsoft Outlook. This can occur when all the changes aren't consistent in the directory. To lessen the risk of a user gaining access to the mailbox even though the account has been disabled for Unified Messaging, you can manually force replication to occur or remove all Unified Messaging information from the user's mailbox when the user is disabled for Unified Messaging.

Voice mail-enabled user procedures

Unified Messaging > Set up voice mail for users > Voice mail for users >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-05-03*

Enable a user for voice mail

Include text with the email message sent when a user Is enabled for voice mail

Manage voice mail settings for a user

Assign a UM mailbox policy

Change the UM dial plan

Enable calls from users who aren't UM-enabled

Disable calls from users who aren't UM-enabled

Allow callers without a caller ID to leave a voice message

Include text with the email message sent when a voice message Is received

Prevent callers without a caller ID from leaving a voice message

Disable voice mail for a user

Change a SIP address

Change an extension number

Add a SIP address

Remove a SIP address

Add an extension number

Remove an extension number

Change an E.164 number

Add an E.164 number

Remove an E.164 number

Enable a user for voice mail

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

When you enable a user for Unified Messaging (UM), a default set of properties are applied to the user, and the user will be able to use the voice mail features included with Unified Messaging. After you enable a user for voice mail, you have the option of adding a Session Initiation Protocol (SIP) address for the user if they're assigned to a UM mailbox policy that's linked to a SIP URI dial plan. Or, you can add an E.164 number for the user if they're assigned to a UM mailbox policy that's linked to an E.164 dial plan. In both cases, the user must still have an extension number configured.

An extension number is required for each user that's associated with a telephone extension, SIP Uniform Resource Identifier (URI), or E.164 dial plan. The extension number must be the correct number of digits, as specified in the UM dial plan for the UM mailbox policy.

Note:

You must add, remove, or modify extension numbers for all UM-enabled users by using the EAC or the Shell, even if they're linked to a SIP URI or E.164 dial plan. To add, remove or modify SIP address or E.164 numbers for users, you'll need to use the Shell because those options aren't available in the EAC.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable a user for voice mail

1. In the EAC, click **Recipients**.
2. In the List view, select the user whose mailbox you want to enable for Unified Messaging.
3. In the Details pane, under **Phone and Voice Features**, click **Enable**.

4. On the **Enable UM mailbox** page, click the **Browse** button next to **UM mailbox policy**, locate the UM mailbox policy to assign the user from the list, and then click **OK**.
5. On the **Enable UM mailbox** page, complete the following boxes:
    - **SIP address** or **E.164 number** In the **SIP address** or **E.164 number** text box, enter the SIP address or E.164 number for the user. These options are available if the user that you enable for Unified Messaging is assigned to a UM mailbox policy that's linked to either a SIP URI or an E.164 dial plan. You can't add a SIP address or E.164 number for a user if the user is associated with a telephone extension dial plan.

When you assign a user to a UM mailbox policy that's linked to a SIP URI or E.164 dial plan, you must enter an extension number for the user. The user will use this extension number when accessing their mailbox via Outlook Voice Access. The number of digits that you configure in this box must match the number of digits configured on the SIP URI or E.164 dial plan.

- **Extension number** Use this text box to manually enter the extension number for the user you're enabling for UM.

You must provide a valid extension number for the user and must match the number of digits specified on the dial plan. You can only enter digits from 1 through 20. The typical extension number is 3 to 7 digits long. The number of digits in the extension is set on the dial plan that's linked to the UM mailbox policy that's assigned to the user.

- Under **PIN settings**, complete the following:
    - **Automatically generate PIN** Click this button to automatically generate a PIN for the UM-enabled user to use for voice mail access via Outlook Voice Access. This is the default setting. The PIN is automatically generated based on the PIN policies configured on the UM mailbox policy assigned to the user. Using this setting will help protect the user's PIN. The PIN is sent to the user in the welcome message they receive after they're enabled for UM. By default, they'll have to change this PIN when they first sign in to their mailbox to get their voice mail.
    - **Type a PIN** Click this button to enter a PIN that the user will use to access the voice mail system. The PIN must comply with the PIN policy settings configured on the UM mailbox policy associated with this UM-enabled user. For example, if the UM mailbox policy is configured to accept only PINs that contain seven or more digits, the PIN you enter in this box must be at least seven digits long.
    - **Require the user to reset their PIN the first time they sign in** Select this check box to force the user to reset their voice mail PIN when they access the voice mail system from a telephone using Outlook Voice Access for the first time. They will be prompted to enter a PIN that's more familiar to them.It's a security best practice to force UM-enabled users to change their PIN when they first sign in to help protect against unauthorized access to their data and Inbox. This check box is selected by default.
6. On the **Enable UM mailbox** page, review your settings. Click **Finish** to enable the user for voice mail. Click **Back** to make configuration changes.

Use the Shell to enable a user for voice mail

This example enables Unified Messaging on the mailbox of tonysmith@contoso.com, sets the extension number to 51234, sets the PIN for the user to 5643892, and assigns the user to a UM mailbox policy named `MyUMMailboxPolicy`.

```
Enable-UMMailbox -Identity tonysmith@contoso.com -
UMMailboxPolicy MyUMMailboxPolicy -Extensions 51234 -PIN
5643892 -PINExpired $true
```

This example enables Unified Messaging on the mailbox of tonysmith@contoso.com, assigns the user to a UM mailbox policy named `MyUMMailboxPolicy`, and sets the extension number, SIP address, and PIN for the user.

```
Enable-UMMailbox -Identity tonysmith@contoso.com -
UMMailboxPolicy MyUMMailboxPolicy -Extensions 51234 -PIN
5643892 -SIPResourceIdentifier "tonysmith@contoso.com" -
PINExpired $true
```

Include text with the email message sent when a user Is enabled for voice mail

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-16*

When a user's mailbox is enabled for Unified Messaging (UM) voice mail, an email message is sent that welcomes the user to Unified Messaging. This message contains the PIN information the user will use to first access the voice mail system.

You can customize the text that's sent in the welcome email message by adding text in the **When a user is enabled for Unified Messaging** box on a UM mailbox policy. You can include such information as the UM technical support telephone numbers or additional Outlook Voice Access numbers. After you add the text, it will be included in the email message sent when users associated with the UM mailbox policy are enabled for Unified Messaging.

Note:
The custom text you add to the welcome message is limited to 512 characters, and it can include simple HTML text.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to customize the text sent when a mailbox is enabled for Unified Messaging

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **Message text**, in the text box for **When a user is enabled for Unified Messaging**, enter the text you want to include in the email message that's sent when users are enabled for Unified Messaging voice mail.
4. Click **Save**.

Use the Shell to customize the text sent when a mailbox is enabled for Unified Messaging

This example enables UM-enabled users who are associated with a UM mailbox policy to receive additional instructions about UM and the Outlook Voice Access number that they can use to access their mailbox over a phone.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
UMEnabledText "You've been enabled for Unified Messaging
voice mail. To access your Exchange mailbox, call your
internal telephone extension number. From outside your
office, call 425-555-1234."
```

Manage voice mail settings for a user

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can view or set the Unified Messaging (UM) and voice mail features and configuration settings for a user that's been enabled for UM and voice mail. For example, you can do the following:

- Reset their Outlook Voice Access PIN.
- Add a personal operator extension number.
- Add other extension numbers.
- Enable or disable Automatic Speech Recognition (ASR).
- Enable or disable Call Answering Rules.
- Enable or disable access to their email or calendar.

Note:

Some of the settings and features can only be configured by using the Shell.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the existing user is currently enabled for Unified Messaging. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to view or configure a UM-enabled user's properties

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox for which you want to change the UM mailbox policy.
3. In the details pane, under **Phone and Voice Features** > **Unified Messaging**, click **View details**.
4. On the **UM Mailbox** page, click **UM mailbox settings** to view or change the following UM properties for an existing UM-enabled user:
   - **PIN Status** This display-only field shows the status of the user's mailbox. By default, when a user is UM-enabled, the PIN status is listed as **Not locked out**. However, if the user has input an incorrect Outlook Voice Access PIN multiple times, the status is listed as **Locked Out**.
   - **UM mailbox policy** This box shows the name of the UM mailbox policy associated with the UM-enabled user. You can click **Browse** to locate and specify the UM mailbox policy to be associated with this UM mailbox.
   - **Personal operator extension** Use this box to specify the operator extension number for the user. By default, an extension number isn't configured. The length of the extension number can be from 1 through 20 characters. This enables incoming calls for the UM-enabled user to be forwarded to the extension number that you specify in this box.

You can configure other types of operator extension numbers on dial plans and auto attendants. However, those extensions are generally meant for company-wide receptionists or operators. The personal operator extension setting could be used when an administrative assistant or personal assistant answers incoming calls before they're answered for a particular user.

5. On the **UM Mailbox** page, under **Other extensions**, you can add, change, and view extension numbers for the user.
   - To add an extension number, click **Add +**. On the **Add another extension** page, use **Browse** to select the UM dial plan, and then enter the extension number in the **Extension number** box.
   - To remove an extension number, select the extension number you want to remove, and then click **Remove −**.
6. If you make any changes, click **Save**.

Use the Shell to configure features for a UM-enabled user

This example disables Play on Phone and missed call notifications, but enables text message (SMS) notifications.

Note:

For on-premises and hybrid deployments, when you're integrating Unified Messaging and Lync Server, missed call notifications aren't available to users who have a mailbox located on an Exchange 2007 or Exchange 2010 Mailbox server. A missed call notification is generated when a user disconnects before the call is sent to a Mailbox server.

```
Set-UMMailbox -Identity tony@contoso.com -UMEnabled $true -
UMMailboxPolicy AdminPolicy -MissedCallNotificationEnabled
$false -PlayonPhoneEnabled $false -
SMSMessageWaitingNotificationEnabled $true
```

This example prevents a user from accessing the calendar, but enables access to email when the user is using Outlook Voice Access.

```
Set-UMMailbox -Identity tony@contoso.com -UMEnabled $true -
UMMailboxPolicy AdminPolicy -Extension 523456 -FAXEnabled
$true -TUIAccessToCal $false -TUIAccessToEmail True
```

This example prevents a user from accessing the calendar and email when the user is using Outlook Voice Access.

```
Set-UMMailbox -Identity tony@contoso.com -
TUIAccessToCalendarEnabled $false -TUIAccessToEmailEnabled
$false
```

This example prevents a user from creating call answering rules, receiving incoming faxes, and using Outlook Voice Access, but enables Automatic Speech Recognition (ASR).

```
Set-UMMailbox -Identity tony@contoso.com -
AutomaticSpeechRecognitionEnabled $true -
CallAnsweringRulesEnabled $false -FaxEnabled $false -
SubscriberAccessEnabled $false
```

Use the Shell to view a UM-enabled user's properties

This example displays a list of all the UM-enabled mailboxes in the forest in a formatted list.

```
Get-UMMailbox | Format-List
```

This example displays the UM mailbox properties for tonysmith@contoso.com.

```
Get-UMMailbox -Identity tonysmith@contoso.com
```

Important:

When you're running Exchange 2007 and Exchange 2013 and the user's mailbox is located on an Exchange 2007 Mailbox server, running the **Get-UMMailbox** cmdlet won't work correctly. To resolve the issue, run the **Get-UMMailbox** cmdlet from an Exchange 2007 server or a computer running the Exchange 2007 administrative tools.

Assign a UM mailbox policy

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-30*

When you enable a user for Unified Messaging (UM) and voice mail, you must select the UM mailbox policy that will be associated with the user's mailbox. You can change the UM mailbox policy associated with the user's mailbox after the user has been enabled for UM.

You create UM mailbox policies to apply a common set of policies or security settings to a collection of mailboxes of UM-enabled users. You can use UM mailbox policies to apply settings such as the following:

- PIN policies
- Dialing restrictions
- Other general UM mailbox policy properties

Note:

A default UM mailbox policy is created every time you create a UM dial plan. You can delete the default UM mailbox policies or create additional UM mailbox policies based on the needs of your organization.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user is enabled for Unified Messaging. For detailed steps, see Enable a user for voice mail.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the UM mailbox policy assigned to a UM-enabled user

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox for which you want to change the UM mailbox policy.
3. In the details pane, under **Phone and Voice Features** > **Unified Messaging**, click **View details**.
4. On the **UM Mailbox** page, click **UM mailbox settings**, and then click **Edit** .
5. On the **UM Mailbox** page > next to **UM mailbox policy**, click **Browse** to locate the UM mailbox policy for the user.
6. Click **Save**.

Use the Shell to change the UM mailbox policy assigned to a UM-enabled user

This example associates a UM-enabled user named Tony Smith with a UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailbox -Identity tonysmith@contoso.com -
UMMailboxPolicy MyUMMailboxPolicy
```

Change the UM dial plan

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

You may need to move a user who is enabled for Unified Messaging (UM) to a different UM dial plan or change the dial plan that's associated with the user. For example, you might want to move a UM-enabled user from a Telephone Extension dial plan to a SIP URI dial plan.

To change the UM dial plan, you'll have to disable the user for Unified Messaging and then enable the user for Unified Messaging on the new UM dial plan. This is because different dial plans may have different settings and requirements, such as different extension lengths or different URI types. For example, SIP URI dial plans require a SIP Resource Identifier to be assigned to each UM-enabled mailbox, but Telephone Extension dial plans don't. Also, each UM mailbox contains references to both the UM dial plan and the UM mailbox policy. The UM mailbox policy, in turn, contains references to the UM dial plan. If you change the primary proxy address for a UM-enabled user to point to a different dial plan, the UM mailbox is in an inconsistent state.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the existing Exchange recipient is enabled for Unified Messaging. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

How do you do this?

Step 1: Create the new UM dial plan

Important:

If you're migrating UM-enabled users to Microsoft Office Communications Server 2007 R2 or to Microsoft Lync Server, you must first create a SIP URI dial plan.

For detailed instructions, see Create a UM dial plan.

Step 2: Disable the user for Unified Messaging

For detailed instructions, see Disable voice mail for a user.

Step 3: Enable the user for Unified Messaging on the new UM dial plan

Important:
If you're moving users to an environment with Office Communications Server 2007 R2 or Lync Server, you must also include a SIP Resource Identifier for the user when you enable them for UM. You must also select the UM mailbox policy that's associated with a SIP dial plan.

For detailed instructions, see Enable a user for voice mail.

Enable calls from users who aren't UM-enabled

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

You can enable or disable calls from users who aren't enabled for Unified Messaging (UM). By default, Unified Messaging allows incoming calls from unauthenticated callers through an auto attendant to be transferred to UM-enabled users. With this option enabled, users from outside an organization can transfer calls to UM-enabled users.

If this setting has been disabled for a UM-enabled user, the user's mailbox can still be located using a directory search. However, if an external caller tries to transfer to the user, the system says, "I'm sorry, I am unable to transfer the call to this user." The caller is then transferred to the operator, if an operator has been configured on the auto attendant. If no operator has been configured on the auto attendant, the call is transferred to a dial plan operator, if one has been configured. If no operator extension has been configured on the speech-enabled auto attendant, the dual tone multi-frequency (DTMF) fallback auto attendant, or the dial plan, the system responds by saying, "Sorry. Neither the operator or the touchtone service are available."

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform this procedure, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to enable calls from users who aren't UM-enabled

This example allows Tony Smith to receive voice calls from callers who aren't UM-enabled.

```
Set UMMailbox -Identity tony@contoso.com -
AllowUMCallsFromNonUsers SearchEnabled
```

Disable calls from users who aren't UM-enabled

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

You can enable or disable calls from users who aren't enabled for Unified Messaging (UM). By default, Unified Messaging allows incoming calls from unauthenticated callers through an auto attendant to be transferred to UM-enabled users. With this setting enabled, users from outside an organization can transfer calls to UM-enabled users.

If this setting has been disabled for a UM-enabled user, the user's mailbox can still be located using a directory search. However, if an external caller tries to transfer to the user, the system says, "I'm sorry, I am unable to transfer the call to this user." The caller is then transferred to the operator, if an

operator has been configured on the auto attendant. If no operator has been configured on the auto attendant, the call is transferred to a dial plan operator, if one has been configured. If no operator extension has been configured on the speech-enabled auto attendant, the dual tone multi-frequency (DTMF) fallback auto attendant, or the dial plan, the system responds by saying, "Sorry. Neither the operator nor the touchtone service are available."

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform this procedure, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to disable calls from users who aren't UM-enabled

This example prevents Tony Smith from receiving voice calls from callers who aren't UM-enabled.

```
Set UMMailbox -Identity tony@contoso.com -
AllowUMCallsFromNonUsers None
```

Allow callers without a caller ID to leave a voice message

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can allow UM-enabled users to receive voice mail messages from anonymous callers or prevent them from doing so. By default, when users are enabled for Unified Messaging (UM) and voice mail, they can receive calls that are anonymous and don't contain caller ID information.

In most cases, calls received by Unified Messaging contain a caller ID that can be used to determine the source of the incoming call. However, incoming calls may not include caller ID information for the following reasons:

- Your organization's telephony equipment is configured not to include caller ID information.
- The incoming call is from a mobile or external telephone.
- The caller has disabled caller ID on their telephone.

Because the *AnonymousCallersCanLeaveMessages* parameter is enabled by default, a UM-enabled user can receive a voice message even if caller ID information isn't included. If the *AnonymousCallersCanLeaveMessages* option is disabled, and the UM-enabled user receives a call that doesn't include a caller ID, the call will be identified as anonymous, and the UM-enabled user won't receive a voice message.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform this procedure, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to allow voice messages from anonymous

callers to be received

This example allows UM-enabled user tonysmith@contoso.com to receive voice messages from incoming calls that don't contain caller ID information.

```
Set-UMMailbox -Identity tonysmith@contoso.com -
AnonymousCallersCanLeaveMessages $true
```

Include text with the email message sent when a voice message Is received

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-16*

You can include additional text in the email message that's sent when a voice mail message is received by a user who is enabled for Unified Messaging (UM) voice mail. By default, the text that's included with a voice message indicates only that the user has received a voice message. However, you can create a custom message by adding text in the **When a user receives a voice message** box on a UM mailbox policy. For example, the text can include information about system security policies and describe the correct way to handle voice messages in your organization. After you add the text, it will be included in each email message that's sent when UM-enabled users associated with the UM mailbox policy receive a voice message.

Note:
The custom text that accompanies a voice message is limited to 512 characters, and can include simple HTML text.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.

- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the text included with a voice message

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **Message text**, in the text box for **When a user receives a voice message**, enter the text you want to include in the email message that's sent when users receive a voice message.
4. Click **Save**.

Use the Shell to change the text included with a voice message

This example includes the additional text, "Do not forward voice messages to users outside this organization", with voice messages sent to users who are associated with the UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
VoiceMailText "Do not forward voice messages to users
outside this organization."
```

Prevent callers without a caller ID from leaving a voice message

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can allow UM-enabled users to receive voice messages from anonymous callers or prevent them from doing so. By default, when users are enabled for Unified Messaging (UM) and voice mail, they can receive calls that are anonymous and don't contain caller ID information.

In most cases, calls received by Exchange servers contain a caller ID that can be used to determine the source of the incoming call. However, incoming calls may not include caller ID information for the following reasons:

- Your organization's telephony equipment is configured not to include caller ID information.
- The incoming call is from a mobile or external telephone.
- The caller has disabled caller ID on their telephone.

Because the *AnonymousCallersCanLeaveMessages* parameter is enabled by default, a UM-enabled user can receive a voice message even if caller ID information isn't included. If the *AnonymousCallersCanLeaveMessages* option is disabled, and the UM-enabled user receives a call that doesn't include a caller ID, the call will be identified as anonymous, and the UM-enabled user won't receive a voice message.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform this procedure, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to prevent voice messages from anonymous

callers from being received

This example prevents UM-enabled user tonysmith@contoso.com from receiving voice messages from calls that don't contain caller ID information.

```
Set-UMMailbox -Identity tonysmith@contoso.com -
AnonymousCallersCanLeaveMessages $false
```

Disable voice mail for a user

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

You can disable Unified Messaging (UM) for a UM-enabled user. When you do this, the user can no longer use the voice mail features found in Unified Messaging. If you prefer, when you disable UM for a user, you can keep the UM settings for the user.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the existing user is currently enabled for Unified Messaging. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to disable Unified Messaging and voice mail for a user

1. In the EAC, click **Recipients**.
2. In the list view, select the user whose mailbox you want to disable for Unified Messaging.
3. In the Details pane, under **Phone and Voice Features**, under **Unified Messaging**, click **Disable**.
4. In the **Warning** box, click **Yes** to confirm that Unified Messaging will be disabled for the user.

Use the Shell to disable Unified Messaging and voice mail for a user

This example disables Unified Messaging and voice mail for the user tonysmith@contoso.com, but keeps the UM mailbox settings.

```
Disable-UMMailbox -Identity tonysmith@contoso.com -
KeepProperties $True
```

Change a SIP address

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-14*

When you enable a user for UM and link them to a SIP URI dial plan, two EUM proxy addresses are created. One contains the user's extension number and the other contains a SIP address for the user. The extension number is used when the user calls in to an Outlook Voice Access number.

SIP URI dial plans and SIP addresses are used when you're integrating UM and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server. The SIP address is used by Communications Server or Lync Server to route incoming calls and send voice mail to the user. By default, the SIP address that's used by UM will be the SIP address that's used by Communications Server or Lync Server.

You can change the primary SIP address that was added when the user was enabled for UM or a secondary SIP address that was added later, along with the EUM proxy addresses for the user. The primary SIP address you added when the user was enabled for UM will be listed as the primary EUM

proxy address. Any additional secondary SIP addresses you added will be listed as secondary EUM proxy addresses. When secondary SIP addresses are changed, callers can leave voice mail for the user at all SIP endpoints that the user is signed in to using the new SIP addresses. All the voice messages will be delivered to the same user's mailbox.

You can use the EAC or the Shell to change a primary or a secondary SIP address. You can use the **Email Address** page on the user's mailbox in the EAC to change a primary or a secondary SIP address. You can't use the **UM Mailbox** page in the EAC to change a primary or secondary SIP address.

You can view the primary and secondary SIP addresses for a user by using the **Get-UMMailbox** cmdlet or the **Get-Mailbox** cmdlet in the Shell.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a SIP URI UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the existing user is enabled for UM and linked to a SIP URI dial plan. For detailed steps, see Enable a user for voice mail.
- Before you perform these procedures, confirm that the SIP address that will be assigned to the user is valid and formatted correctly.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the primary or a secondary SIP address

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox for which you want to change a SIP address, and then click **Edit**

.

3. On the **User Mailbox** page, under **Email address**, select the SIP address you want to change, and then click **Edit** . The primary SIP address is listed in bold letters and numbers.
4. On the **Email address** page, in the **Address/Extension** box, enter the new SIP address for the user, and then click **OK**. If you need to select a new UM dial plan, you can click **Browse**.
5. Click **Save**.

Use the Shell to change the primary or a secondary SIP address

This example changes a SIP address for Tony Smith.

Note:
Before you change a SIP address using the Shell, you need to determine the position of the EUM proxy address that you want to change. To determine the position, use the **$mbx.EmailAddresses** command. The first EUM proxy address is the default (primary) SIP address and it will be 0 in the list.

```
$mbx=Get-Mailbox tony.smith
$mbx.EmailAddresses.Item(1)="eum:tsmith@contoso.com;phone-context=MySIPDialPlan.contoso.com"
Set-Mailbox tony.smith -EmailAddresses $mbx.EmailAddresses
```

Change an extension number

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-14*

When you enable a user for UM and link them to a telephone extension dial plan, an EUM proxy address is created for the user that contains the user's extension number. You must define at least one extension number for UM to use so voice mail can be sent to the user's mailbox. The extension number is also used when the user calls in to an Outlook Voice Access number.

You can change the primary extension number that was added when the user was enabled for UM or a secondary extension number that was added later, along with the related EUM proxy addresses for the user. The primary extension number you added when the user was enabled for UM will be listed as the primary EUM proxy address. Any additional secondary extension numbers you added will be listed as secondary EUM proxy addresses. When extension numbers have been changed, callers can leave voice mail for the user at all the new extension numbers that have been set. All the

voice messages will be delivered to the same user's mailbox.

You can use the EAC or the Shell to change a primary or a secondary extension number for a user. You can use the **Email Address** page on the user's mailbox in the EAC to change a primary or secondary extension number. You can't use the **UM Mailbox** page in the EAC to change a primary extension number, but you can use it to change a secondary extension number. If you want to change a secondary extension number, you must first remove the existing secondary extension number and then add the correct secondary extension number for the user.

You can view the primary and secondary extension numbers for a user by using the **Get-UMMailbox** cmdlet or the **Get-Mailbox** cmdlet in the Shell.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a telephone extension UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user's mailbox has been enabled for UM and linked to a telephone extension dial plan. For detailed steps, see Enable a user for voice mail.
- Before you perform these procedures, confirm that the extension number that will be assigned to the user contains the correct number of digits set on the UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the primary or secondary extension number

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox for which you want to change an extension number, and then click **Edit** ✎.

3. On the **User Mailbox** page, under **Email address**, select the extension number you want to change, and then click **Edit** . The primary extension number is listed in bold letters and numbers.
4. On the **Email address** page, in the **Address/Extension** box, enter the new extension number for the user. If you need to select a new UM dial plan, you can click **Browse**.
5. Click **Save**.

Use the Shell to change the primary or secondary extension number

This example changes the extension number to 22222 for Tony Smith, a UM-enabled user.

Note:

Before you change an extension number using the Shell, you need to determine the position of the EUM proxy address that you want to change. To determine the position, use the **$mbx.EmailAddresses** command. The first EUM proxy address is the default (primary) extension number and it will be 0 in the list.

```
$mbx=Get-Mailbox tony.smith
$mbx.EmailAddresses.Item(0)="eum:22222;phone-
context=MyDialPlan.contoso.com"
Set-Mailbox tony.smith -EmailAddresses $mbx.EmailAddresses
```

Add a SIP address

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Online*

Topic Last Modified: *2012-11-14*

When you enable a user for UM and link them to a SIP URI dial plan, two EUM proxy addresses are created. One contains the user's extension number and the other contains a SIP address for the user. The extension number is used when the user calls in to an Outlook Voice Access number.

SIP URI dial plans and SIP addresses are used when you're integrating UM and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server. The SIP address is used by Communications Server or Lync Server to route incoming calls and send voice mail to the user. By default, the SIP address that's used by UM will be the SIP address that's used by Communications Server or Lync Server.

The primary SIP address you added when the user was enabled for UM will be listed as the primary EUM proxy address. If the primary SIP address was removed, the first EUM proxy address you add

that contains the user's SIP address will be listed as the primary EUM proxy address. Any additional SIP addresses you add will be listed as secondary EUM proxy addresses. When secondary SIP addresses are added, callers can leave voice mail for the user at SIP endpoints that the user is signed in to using the SIP addresses. All the voice messages will be delivered to the same user's mailbox.

You can use the EAC or the Shell to add a primary or a secondary SIP address for a user. You can use the **Email Address** page on the user's mailbox in the EAC to add a primary or secondary SIP address. You can't use the **UM Mailbox** page in the EAC to add a primary or secondary SIP address.

You can view the primary and secondary SIP addresses for a user by using the **Get-UMMailbox** cmdlet or the **Get-Mailbox** cmdlet in the Shell.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a SIP URI UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the existing user is enabled for UM and linked to a SIP URI dial plan. For detailed steps, see Enable a user for voice mail.
- Before you perform these procedures, confirm that the SIP address that will be assigned to the user is valid and formatted correctly.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to add a primary or secondary SIP address

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox for which you want to add a SIP address, and then click **Edit** ✎.
3. On the **User Mailbox** page, under **Email address**, click **Add +**.
4. On the **New email address** page, select **EUM** and, in the **Address/Extension** box, enter the new

SIP address for the user.

5. On the **New email address** page, under **Dial plan**, click **Browse** to select the SIP URI dial plan, and then click **OK**.
6. Click **Save**.

Use the Shell to add a SIP address

This example adds a SIP address for Tony Smith, a UM-enabled user.

Note:
Before you add a SIP address using the Shell, you need to determine the position of the EUM proxy address that you want to add. To determine the position, use the **$mbx.EmailAddresses** command. The first proxy address in the list will be 0.

```
$mbx=Get-Mailbox tony.smith
$mbx.EmailAddresses +="eum:tsmit@contoso.com;phone-
context=MyDialPlan.contoso.com"
Set-Mailbox tony.smith -EmailAddresses $mbx.EmailAddresses
```

Remove a SIP address

Applies to: *Exchange Online*

Topic Last Modified: *2012-11-14*

When you enable a user for UM and link them to a SIP URI dial plan, two EUM proxy addresses are created. One contains the user's extension number and the other contains a SIP address for the user. The extension number is used when the user calls in to an Outlook Voice Access number.

SIP URI dial plans and SIP addresses are used when you're integrating UM and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server. The SIP address is used by Communications Server or Lync Server to route incoming calls and send voice mail to the user. By default, the SIP address that's used by UM will be the SIP address that's used by Communications Server or Lync Server.

You can remove the primary SIP address that was added when the user was enabled for UM or a secondary SIP address that was added later, along with the EUM proxy address for the user. The primary SIP address you added when the user was enabled for UM will be listed as the primary EUM proxy address. Any additional SIP addresses you added will be listed as secondary EUM proxy addresses. When a SIP address is removed, callers can no longer leave voice mail for the user at the SIP address that was removed even if the user is signed in with the SIP address assigned to the user in Communications Server or Lync Server.

If you remove the primary SIP address, UM won't be able to send voice mail to the user's mailbox and call answering rules won't be processed. After the primary SIP address has been removed, the EUM proxy address for the user will be listed as **Null** on the user's mailbox in the EAC and when you run the **Get-Mailbox** cmdlet in the Shell. Also, when you run the **Get-UMMailbox** cmdlet, the *Extensions*, *PhoneNumber*, and *CallAnsweringRulesExtensions* parameters will be blank or null.

You can use the EAC or the Shell to remove a primary or a secondary SIP address. You can use the **Email Address** page on the user's mailbox in the EAC to remove a primary or a secondary SIP address. You can't use the **UM Mailbox** page in the EAC to remove a primary or secondary SIP address.

You can view the primary and secondary SIP addresses for a user by using the **Get-UMMailbox** cmdlet or the **Get-Mailbox** cmdlet in the Shell.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user's mailbox has been enabled for UM and linked to a SIP URI dial plan. For detailed steps, see Enable a user for voice mail.
- Before you perform these procedures, confirm that the primary and secondary SIP addresses are configured for the user.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to remove the primary or a secondary SIP address

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox from which you want to remove a SIP address, and then click **Edit** .

3. On the **User Mailbox** page, under **Email address**, select the SIP address that you want to remove from the list, and then click **Delete** 🗑. The primary EUM proxy address or SIP address is listed in bold letters and numbers.
4. Click **Save**.

Use the Shell to remove the primary or a secondary SIP address

This example removes the SIP address tsmith@contoso.com from the mailbox of Tony Smith, a UM-enabled user.

Note:

Before you remove a SIP address using the Shell, you need to determine the position of the EUM proxy address that you want to modify. To determine the position, use the **$mbx.EmailAddresses** command. The first EUM proxy address in the list will be 0.

```
$mbx = Get-Mailbox tony.smith
$mbx.EmailAddresses.Item(1) -
="eum:tsmith@contoso.com;phone-
context=MyDialPlan.contoso.com"
Set-Mailbox tony.smith -EmailAddresses $mbx.EmailAddresses
```

Add an extension number

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-14*

When you enable a user for UM and link them to a telephone extension dial plan, an EUM proxy address is created for the user that contains the user's extension number. You must define at least one extension number for UM to use so voice mail can be sent to the user's mailbox. The extension number is also used when the user calls in to an Outlook Voice Access number.

The primary extension number you added when the user was enabled for UM will be listed as the primary EUM proxy address. If the primary extension number was removed, the first EUM proxy address you add that contains the user's extension number will become the primary EUM proxy address. Any additional extension numbers you add will be listed as secondary EUM proxy addresses. When additional secondary extension numbers are added, callers can leave voice mail for the user at all extension numbers that have been set. All the voice messages will be delivered to the same user's mailbox.

You can use the EAC or the Shell to add a primary or a secondary extension number for a user. You can use the **Email Address** page on the user's mailbox in the EAC to add a primary or secondary extension number. You can't use the **UM Mailbox** page in the EAC to add a primary extension number, but you can use that page to add secondary extension numbers.

You can view the primary and secondary extension numbers for a user by using the **Get-UMMailbox** cmdlet or the **Get-Mailbox** cmdlet in the Shell.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a telephone extension UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user's mailbox has been enabled for UM and linked to a telephone extension dial plan. For detailed steps, see Enable a user for voice mail.
- Before you perform these procedures, confirm that the extension number that will be assigned to the user contains the correct number of digits set on the UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to add a secondary extension number

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox to which you want to add an extension number.
3. In the details pane, **Phone and Voice Features**, under **Unified Messaging**, click **View details**.
4. On the **UM Mailbox** page, click **Other Extensions**, and then click **Add +**.
5. On the **Other extensions** page, next to the **UM dial plan** box, click **Browse** and locate the dial plan for the user.
6. On the **Other extensions** page, in the **Extension number** box, type the extension number, and then click **OK**.

7. Click **Save**.

Use the EAC to add a primary or secondary extension number

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox to which you want to add an extension number, and then click **Edit** .
3. On the **User Mailbox** page, under **Email address**, click **Add +**.
4. On the **New email address** page, select **EUM** and, in the **Address/Extension** box, enter the extension number for the user.
5. On the **New email address** page, under **Dial plan**, click **Browse** to select the telephone extension dial plan, and then click **OK**.
6. Click **Save**.

Use the Shell to add an extension number

This example adds an extension number 22222 for Tony Smith, a UM-enabled user.

Note:
Before you add an extension number using the Shell, you need to determine the position of the EUM proxy address that you want to add. To determine the position, use the **$mbx.EmailAddresses** command. The first proxy address in the list will be 0.

```
$mbx=Get-Mailbox tony.smith
$mbx.EmailAddresses +="eum:22222;phone-
context=MyDialPlan.contoso.com"
Set-Mailbox tony.smith -EmailAddresses $mbx.EmailAddresses
```

Remove an extension number

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-14*

When you enable a user for UM and link them to a telephone extension dial plan, an EUM proxy address is created for the user that contains the user's extension number. You must define at least one extension number for UM to use so voice mail can be sent to the user's mailbox. The extension number is also used when the user calls in to an Outlook Voice Access number.

You can remove the primary extension number that was added when the user was enabled for UM

or a secondary extension number that was added later, along with the related EUM proxy addresses for the user. The primary extension number you added when the user was enabled for UM will be listed as the primary EUM proxy address. Any additional extension numbers you added will be listed as secondary EUM proxy addresses. When an extension number is removed, callers can no longer leave voice mail for the user at the extension number that was removed.

If you remove the primary extension number, UM won't be able to send voice mail to the user's mailbox and call answering rules won't be processed. After the primary extension number has been removed, the EUM proxy address for the user will be listed as **Null** on the user's mailbox in the EAC and when you run the **Get-Mailbox** cmdlet in the Shell. Also, when you run the **Get-UMMailbox** cmdlet, the *Extensions*, *PhoneNumber*, and *CallAnsweringRulesExtensions* parameters will be blank or null.

You can use the EAC or the Shell to remove a primary or a secondary extension number. You can use the **Email Address** page on the user's mailbox in the EAC to remove a primary or a secondary extension number. You can't use the **UM Mailbox** page in the EAC to remove a primary extension number, but you can use it to remove a secondary extension number.

You can view the primary and secondary extension numbers for a user by using the **Get-UMMailbox** cmdlet or the **Get-Mailbox** cmdlet in the Shell.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user's mailbox has been enabled for UM and linked to a telephone extension dial plan. For detailed steps, see Enable a user for voice mail.
- Before you perform these procedures, confirm that the primary and secondary extension numbers are configured for the user.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to remove the primary or secondary extension number

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox from which you want to remove an extension number, and then click **Edit** .
3. On the **User Mailbox** page, under **Email address**, select the extension number that you want to remove from the list, and then click **Delete** . The primary EUM proxy address or extension number is listed in bold letters and numbers.
4. Click **Save**.

Use the EAC to remove a secondary extension number

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the user whose mailbox you want to remove an extension number from.
3. In the details pane, under **Phone and Voice Features** > **Unified Messaging**, click **View details**.
4. On the **Other extensions** page, in the **Extension number** box, select the extension number you want to remove, and then click **Delete** .
5. Click **Save**.

Use the Shell to remove an extension number

This example removes the extension number 12345 from the mailbox of Tony Smith, a UM-enabled user.

Note:

Before you remove an extension number using the Shell, you need to determine the position of the EUM proxy address that you want to modify. To determine the position, use the **$mbx.EmailAddresses** command. The first EUM proxy address in the list will be 0.

```
$mbx = Get-Mailbox tony.smith
$mbx.EmailAddresses.Item(1) -="eum:12345;phone-
context=MyDialPlan.contoso.com"
Set-Mailbox tony.smith -EmailAddresses $mbx.EmailAddresses
```

Change an E.164 number

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-14*

When you enable a user for UM and link them to an E.164 dial plan, two EUM proxy addresses are created. One contains the user's extension number and the other contains the E.164 number for the user. The extension number is used when the user calls in to an Outlook Voice Access number.

You can change the primary E.164 number that was added when the user was enabled for UM or a secondary E.164 number that was added later, along with the EUM proxy addresses for the user. The primary E.164 number you added when the user was enabled for UM will be listed as the primary EUM proxy address. Any additional secondary E.164 numbers you added will be listed as secondary EUM proxy addresses. When E.164 numbers have been changed, callers can leave voice mail for the user at all the new E.164 numbers that have been set. All the voice messages will be delivered to the same user's mailbox.

You can use the EAC or the Shell to change the primary and secondary E.164 numbers for a user. You can use the **Email Address** page on the user's mailbox to change a primary or secondary E.164 number. However, you can't use the **UM Mailbox** page in the EAC to change a primary or secondary E.164 number.

You can view the primary and secondary E.164 numbers for a user by using the **Get-UMMailbox** cmdlet or the **Get-Mailbox** cmdlet in the Shell.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that an E.164 UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user's mailbox has been enabled for UM and linked to an E.164 dial plan. For detailed steps, see Enable a user for voice mail.
- Before you perform these procedures, confirm that the E.164 number that will be assigned to the UM-enabled user is valid.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server,

Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the primary or a secondary E.164 number

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox for which you want to change an E.164 number, and then click **Edit** .
3. On the **User Mailbox** page, under **Email address**, select the E.164 number you want to change, and then click **Edit** . The primary E.164 number is listed in bold letters and numbers.
4. On the **Email address** page, in the **Address/Extension** box, enter the new E.164 number for the user, and then click **OK**. If you need to select a new UM dial plan, you can click **Browse**.
5. Click **Save**.

Use the Shell to change the primary or a secondary E.164 number

This example changes an E.164 number for Tony Smith, a UM-enabled user.

Note:

Before you change an E.164 number using the Shell, you need to determine the position of the EUM proxy address that you want to change. To determine the position, use the **$mbx.EmailAddresses** command. The first EUM proxy address is the default (primary) E.164 number and it will be 0 in the list.

```
$mbx=Get-Mailbox tony.smith
$mbx.EmailAddresses.Item(1)="eum:+14255550123;phone-
context=MyE.164DialPlan.contoso.com"
Set-Mailbox tony.smith -EmailAddresses $mbx.EmailAddresses
```

Add an E.164 number

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-14*

When you enable a user for UM and link them to an E.164 dial plan, two EUM proxy addresses are created. One contains the user's extension number and the other contains the E.164 number for the user. The extension number is used when the user calls in to an Outlook Voice Access number.

The primary E.164 number you added when the user was enabled for UM will be listed as the primary EUM proxy address. If the primary E.164 number was removed, the first EUM proxy address you add that contains the user's E.164 number will be listed as the primary EUM proxy address. Any additional E.164 numbers you add will be listed as secondary EUM proxy addresses. When additional E.164 numbers are added, callers can leave voice mail for the user at all E.164 numbers that have been set. All the voice messages will be delivered to the same user's mailbox.

You can use the EAC or the Shell to add a primary or a secondary E.164 number for a user. You can use the **Email Address** page on the user's mailbox in the EAC to add a primary or secondary E.164 number. You can't use the **UM Mailbox** page in the EAC to add a primary or secondary E.164 number.

You can view the primary and secondary E.164 numbers for a user by using the **Get-UMMailbox** cmdlet or the **Get-Mailbox** cmdlet in the Shell.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that an E.164 UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user's mailbox has been enabled for UM and linked to an E.164 dial plan. For detailed steps, see Enable a user for voice mail.
- Before you perform these procedures, confirm that the E.164 number that will be assigned to the user is valid and formatted correctly.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to add a primary or secondary E.164 number

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox for which you want to add an E.164 number, and then click **Edit** .
3. On the **User Mailbox** page, under **Email address**, click **Add +**.
4. On the **New email address** page, select **EUM** and, in the **Address/Extension** box, enter the new E.164 number for the user.
5. On the **New email address** page, under **Dial plan**, click **Browse** to select the E.164 dial plan and then click **OK**.
6. Click **Save**.

Use the Shell to add an E.164 number

This example adds an E.164 number for Tony Smith, a UM-enabled user.

Note:

Before you add an E.164 number using the Shell, you need to determine the position of the EUM proxy address that you want to add. To determine the position, use the **$mbx.EmailAddresses** command. The first proxy address in the list will be 0.

```
$mbx=Get-Mailbox tony.smith
$mbx.EmailAddresses.Item(2)="eum:+14255550123;phone-context=MyDialPlan.contoso.com"
Set-Mailbox tony.smith -EmailAddresses $mbx.EmailAddresses
```

Remove an E.164 number

Set up voice mail for users > Voice mail for users > Voice mail-enabled user procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-14*

When you enable a user for UM and link them to an E.164 dial plan, two EUM proxy addresses are created. One contains the user's extension number and the other contains the E.164 number for the user. The extension number is used when the user calls in to an Outlook Voice Access number.

You can remove the primary E.164 number that was added when the user was enabled for UM or a secondary E.164 number that was added later, along with the EUM proxy addresses for the user. The primary E.164 number you added when the user was enabled for UM will be listed as the primary EUM proxy address. Any additional E.164 numbers you added will be listed as secondary EUM proxy addresses. When an E.164 number is removed, callers can no longer leave voice mail for the

user at the E.164 number that was removed.

If you remove the primary E.164 number, UM won't be able to send voice mail to the user's mailbox and call answering rules won't be processed. After you remove the primary E.164 number, the EUM proxy address for the user will be listed as **Null** on the user's mailbox in the EAC and when you run the **Get-Mailbox** cmdlet in the Shell. Also, when you run the **Get-UMMailbox** cmdlet, the *Extensions*, *PhoneNumber*, and *CallAnsweringRulesExtensions* parameters will be blank or null.

You can use the EAC or the Shell to remove a primary or a secondary E.164 number for a user. You can use the **Email Address** page on the user's mailbox in the EAC to remove a primary or a secondary E.164 number. You can't use the **UM Mailbox** page in the EAC to remove a primary or secondary E.164 number.

You can view the primary and secondary E.164 numbers for a user by using the **Get-UMMailbox** cmdlet or the **Get-Mailbox** cmdlet in the Shell.

For additional management tasks related to users who are enabled for voice mail, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that an E.164 UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user's mailbox has been enabled for UM and linked to an E.164 dial plan. For detailed steps, see Enable a user for voice mail.
- Before you perform these procedures, confirm that the primary and secondary E.164 numbers are configured for the user.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to remove the primary or a secondary E.164

number

1. In the EAC, navigate to **Recipients** > **Mailboxes**.
2. In the list view, select the mailbox from which you want to remove an E.164 number, and then click **Edit** .
3. On the **User Mailbox** page, under **Email address**, select the E.164 number that you want to remove from the list, and then click **Delete** . The primary EUM proxy address or E.164 number is listed in bold letters and numbers.
4. Click **Save**.

Use the Shell to remove the primary or a secondary E.164 number

This example removes the E.164 number +14255551010 from the mailbox of Tony Smith, a UM-enabled user.

Note:

Before you remove an E.164 number using the Shell, you need to determine the position of the EUM proxy address that you want to modify. To determine the position, use the **$mbx.EmailAddresses** command. The first EUM proxy address in the list will be 0.

```
$mbx = Get-Mailbox tony.smith
$mbx.EmailAddresses.Item(1) -="eum:+14255551010;phone-context=MyDialPlan.contoso.com"
Set-Mailbox tony.smith -EmailAddresses $mbx.EmailAddresses
```

Set up client voice mail features

Exchange Server 2013 > Unified Messaging >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-20*

This topic describes the client features that give users who are enabled for Exchange Unified Messaging (UM) access to the email and voice mail messages in their mailbox. These features let you offer your users simplified access to voice mail and email and an improved overall user experience.

Contents

Outlook Voice Access

Forwarding calls

Voice Mail Preview

Receiving faxes

Voice mail client support

Exchange ActiveSync clients The Microsoft Exchange ActiveSync protocol is used to connect mobile clients, such as those found on Internet-capable mobile devices, to an Exchange mailbox. Users can use mobile devices to access their mailbox and view email messages, view and change calendar and contact information, and listen to their voice mail messages. They can also synchronize email, voice mail, calendar items, and contact information with other devices.

Integration with Outlook Microsoft Outlook enables users to access their Exchange mailbox and view email messages in their Inbox, view and change calendar information, and listen to voice messages by using Microsoft Windows Media Player, which is embedded inside the email messages. By using a supported email client, users gain additional features, such as the Play on Phone functionality.

Integration with Outlook Web App Microsoft Outlook Web App provides users with a set of UM interfaces and tools comparable to a full-featured email client like Outlook. With Outlook Web App, users can access their Exchange mailbox by using a compliant web browser. Like Outlook, Outlook Web App provides Windows Media Player embedded in email messages so that users can listen to voice messages, and enables users to access other features such as Play on Phone.

Outlook Voice Access

In Exchange UM, a UM-enabled user can call in to an internal or external telephone number that's configured on a UM dial plan to access their mailbox and use the Outlook Voice Access menu system. Using this menu, UM-enabled users can read email, listen to voice messages, interact with their Outlook calendar, access their personal contacts, and perform tasks such as configuring their Outlook Voice Access PIN or recording their voice mail greetings. For details, see Setting up Outlook Voice Access.

Forwarding calls

A UM-enabled user can create and configure call answering rules using Outlook or Outlook Web App. Call answering rules let users control how their incoming calls should be handled. The rules are applied to incoming calls similar to the way Inbox rules are applied to incoming email messages, and are stored along with other voice settings in the user's mailbox. Up to nine call answering rules can be set up for each UM-enabled mailbox. These rules are independent of the Inbox rules and

don't take up part of the user's Inbox rules storage quota. For details, see Allow voice mail users to forward calls.

Voice Mail Preview

Voice Mail Preview is a feature that's available to users who receive their voice mail messages from the UM voice mail system. Voice Mail Preview enhances the voice mail experience by providing a text version of audio recordings. For details, see Allow users to see a voice mail transcript.

Receiving faxes

UM forwards incoming fax calls for a UM-enabled user to a dedicated fax partner solution, which establishes the fax call with the fax sender and receives the fax on behalf of the user. Before your UM-enabled users can receive fax messages in their mailbox, you must do the following:

- Enable inbound faxing on the UM dial plan linked to the users by setting the *FaxEnabled* parameter to `$true`.
- Enable inbound faxing on the UM dial plan linked to the users by setting the *Allowfax* parameter to `$true`.
- Enable inbound faxing for the users by setting the *FaxEnabled* parameter to `$true`.
- Set the partner fax server URI to allow inbound faxing.
- Configure authentication between the Mailbox server and the fax partner server.

Setting up Outlook Voice Access

Exchange Server 2013 > Unified Messaging > Set up client voice mail features >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-20*

Microsoft Outlook Voice Access lets users who are enabled for Exchange Unified Messaging (UM) access their mailboxes by using analog, digital, or cellular telephones.

An Outlook Voice Access user (also called a *subscriber*), is a user in an organization who's enabled for Unified Messaging. Subscribers use Outlook Voice Access to access their mailboxes by telephone to retrieve email, voice mail messages, personal contacts, and calendar information.

Contents

Distribution groups and contact groups

Choosing a language

Controlling Outlook Voice Access features

Outlook Voice Access overview

In Microsoft Exchange UM, a UM-enabled user can call in to an internal or external telephone number that's configured on a UM dial plan to access their mailbox and use the Outlook Voice Access menu system. Using this menu, UM-enabled users can read email, listen to voice messages, interact with their Outlook calendar, access their personal contacts, and perform tasks such as configuring their Outlook Voice Access PIN and recording their voice mail greetings.

Two types of users, authenticated and unauthenticated, can call in to an Outlook Voice Access number. When an unauthenticated user calls into an Outlook Voice Access number that is set on a UM dial plan, they are only able to do directory searches for users. Authenticated users, those that input their PIN, can perform directory searches and sign in to their mailbox to listen to email, calendar items, and voice mail, and to search personal contacts. When they are searching for a user in the directory or personal contacts, after the user is located, they can transfer calls to a user or ring the user's extension.

Outlook Voice Access interfaces

Two Unified Messaging user interfaces are available to Outlook Voice Access users: the telephone user interface (TUI) and the voice user interface (VUI) that uses Automatic Speech Recognition (ASR).

Before users can use the VUI in Outlook Voice Access, it must be enabled on the UM dial plan and on the UM mailbox policy and also be enabled for the user. By default, when you create a dial plan and a UM mailbox policy and enable voice mail for a user, the user can use ASR or the Outlook Voice Access VUI to navigate menus, messages, and other options. However, even if the user is able to use the VUI, they will have to use the telephone key pad to enter their PIN, navigate personal options, and perform a directory search. The default settings are listed in the following table.

UM component	Default setting	Exchange Management Shell example to enable VUI access
UM dial plan	Enabled	`Set-UMDialPlan -id MyUMDialPlan -AutomaticSpeechRecognitionEnabled $true`
UM mailbox policy	Enabled	`Set-UMMaiboxPolicy -id MyUMPolicy -AllowAutomaticSpeechRecognition $true`

User's mailbox	Enabled	`Set-UMMailbox -id tonysmith -AutomaticSpeechRecognitionEnabled $true`

The following section includes scenarios that describe the VUI functionality.

Return to top

Outlook Voice Access scenarios

Here are examples of how Outlook Voice Access can be used from a telephone:

- **Access email** An Outlook Voice Access user places a call to an Outlook Voice Access number from a telephone and wants to access their email. The voice prompt says, "Welcome. You're connected to Microsoft Exchange. To access your mailbox, please enter your extension. To contact someone, press the pound key." After the user enters a mailbox extension number, the voice prompt says, "Please enter your PIN and press the pound key." After the user enters a PIN, the voice prompt says, "You have two new voice mails, 10 new email messages, and your next meeting is at 10:00 A.M. Please say voice mail, email, calendar, personal contacts, directory, or personal options." When the user says "Email," the voice mail system reads the message header and then the name, subject, time, and priority for the messages that are in the user's mailbox.
- **Access calendar** An Outlook Voice Access user places a call to an Outlook Voice Access number from a telephone and wants to access their calendar. The voice prompt says, "Welcome. You're connected to Microsoft Exchange. To access your mailbox, please enter your extension. To contact someone, press the pound key." After the user enters a mailbox extension number, the voice prompt says, "Please enter your PIN and press the pound key." After the user enters a PIN, the voice prompt says, "You have two new voice mails, 10 new email messages, and your next meeting is at 10:00 A.M. Please say voice mail, email, calendar, personal contacts, directory, or personal options." When the user says "Calendar," the voice mail system says, "Sure, and which day should I open?" The user says, "Today's calendar." The voice mail system responds by saying, "Opening today's calendar." The voice mail system reads each calendar appointment for that day for the user.

Note:

If a Mailbox server running the Microsoft Exchange Unified Messaging service encounters a corrupted calendar item in a user's mailbox, it will fail to read the item, return the caller to the Outlook Voice Access main menu, and skip reading any additional meetings that may be scheduled for the rest of the day.

- **Access voice mail** An Outlook Voice Access user places a call to an Outlook Voice Access number from a telephone and wants to access voice mail. The voice prompt says, "Welcome. You're connected to Microsoft Exchange. To access your mailbox, please enter your extension. To contact someone, press the pound key." After the user enters a mailbox extension number, the voice prompt says, "Please enter your PIN and press the pound key." After the user enters a PIN, the voice prompt says, "You have two new voice mails, 10 new email messages, and your next meeting is at 10:00 A.M. Please say voice mail, email, calendar, personal contacts, directory, or

personal options." The user says "Voice mail," and the voice mail system reads the message header and then the name, subject, time, and priority for the voice messages that are in the user's mailbox.

Note:
If speech recognition is enabled, users can access their UM-enabled mailbox using speech input. Subscribers can also use touchtone, also known as dual tone multi-frequency (DTMF), by pressing 0. Speech recognition isn't enabled for PIN input.

- **Locate a user in the directory** An Outlook Voice Access user places a call to an Outlook Voice Access number from a telephone and wants to locate a person in the directory by spelling their email alias. The voice prompt says, "Welcome. You're connected to Microsoft Exchange. To contact someone, press the pound key." The user presses the pound key, and then uses touchtone inputs to spell the SMTP address of the person.

Note:
The directory search feature with an Outlook Voice Access number isn't speech-enabled. Users can spell the name of the person they want to contact only by using touchtone inputs.

Important:
In some companies (especially in East Asia), office telephones may not have letters on the keys of the telephone. This makes the spell-the-name feature that uses the touchtone interface almost impossible to use without a working knowledge of the key mappings. By default, Unified Messaging uses the E.161 key mapping. For example, 2=ABC, 3=DEF, 4=GHI, 5=JKL, 6=MNO, 7=PQRS, 8=TUV, 9=WXYZ.

When inputting a combination of letters and numbers, for example, Mike1092, the numeric digits are mapped to themselves. For an email alias of Mike1092 to be entered correctly, the user must press the numbers 64531092. Also, for characters other than A-Z and 0-9, there isn't a telephone key equivalent. Therefore, these characters shouldn't be entered. For example, the email alias jim.wilson would be entered as 546945766. Even though there are 10 characters to be input, the user enters only 9 digits because there's no digit equivalent for the period (.).

Return to top

Distribution groups and contact groups

Users can use Outlook Voice Access to send or forward a voice message, an email message, or a meeting request. They can send or forward the message or meeting request to any of the following:

- A person in their personal Contacts folder
- A person in their organization's shared address book
- A contact group they've created in their Contacts folder
- A distribution group included in their organization's shared address book

They can send messages and meeting requests by using the VUI (if ASR has been turned on) or by using touchtone inputs on their telephone keypad. They can also use Outlook Voice Access to listen to details about a group, including the members of the group.

Note:

If a user tries to send a message to a group (either a distribution group in their shared address book or a contact group in their personal Contacts folder) that doesn't include any members, the voice mail system won't give them the option to send or forward the message or meeting request. If they try to add a group with no members as one of the recipients of a message or meeting request that they're creating over the phone, the voice mail system won't add the group to the message, and will say "The message could not be sent because the contact does not appear to have a valid email address."

Choosing a language

Users can't change the language that Outlook Voice Access uses to speak to them and that they use when they reply to it. The voice mail system tries to find and use the best match for the language the user chose when they signed in to Microsoft Outlook Web App or the language they chose on the regional settings in Outlook Web App. If the language they chose isn't supported by Outlook Voice Access, the voice mail system will use the same language that callers hear when they're prompted to leave a voice message.

Return to top

Controlling Outlook Voice Access features

By default, when users dial in to Outlook Voice Access, they can use the telephone to access their calendar, email, and personal contacts, and to search the directory. You can use the Shell to prevent users from accessing one or more of these features when they use Outlook Voice Access to access their mailbox. When you modify Outlook Voice Access features on a UM mailbox policy, your changes affect all users who are associated with the UM mailbox policy. You can also disable some features on a single user's mailbox, although other features can only be disabled on a UM mailbox policy and aren't available on an individual mailbox.

Note:

You can use only the Shell to modify the Outlook Voice Access TUI settings for UM-enabled mailboxes or UM mailbox policies.

UM mailbox policy settings You can disable users' access to the following Outlook Voice Access features on a UM mailbox policy:

- Automatic Speech Recognition
- PIN-less access to voice mail
- Voice responses to other messages
- TUI access to their calendar
- TUI access to the directory
- TUI access to their email
- TUI access to their personal Contacts

UM-enabled mailbox settings You can disable a user's access to the following Outlook Voice

Access features on the user's mailbox:

- TUI access to the calendar
- TUI access to email
- Automatic Speech Recognition

You can prevent users from receiving voice mail, but let them retain the ability to access their mailbox using Outlook Voice Access. You can enable a user for UM and configure the user's mailbox with an extension number that isn't currently being used by another user in the organization.

Return to top

Outlook Voice Access commands

Topic Last Modified: *2013-04-30*

Outlook Voice Access lets Unified Messaging (UM)-enabled users access their mailbox using analog, digital, or mobile telephones. Using the menu system found in Outlook Voice Access, UM-enabled users can read email, listen to voice messages, interact with their Outlook calendar, access their personal contacts, and manage personal options such as configuring their Outlook Voice Access PIN or recording their voice mail messages. This topic contains a list of the Outlook Voice Access commands and how users can use them when they access their mailbox by calling an Outlook Voice Access number.

Outlook Voice Access user interfaces

Outlook Voice Access consists of two user interfaces: the Telephone User Interface (TUI) that uses a telephone keypad and the Voice User Interface (VUI) that uses voice commands. Users can use Outlook Voice Access to access the voice mail system from an external or internal telephone to access their personal email, voice messages, contacts, and calendaring information in their mailbox.

Email and voice mail commands reference

As an Outlook Voice Access user, when you dial in to an Outlook Voice Access number, you're presented with menu options that enable you to access your mailbox and manage your email and voice mail. The following table lists the commands that are available for managing your email and voice mail.

Email and voice mail commands

Voice command	Touchtone command	Description

"Play"		Plays the current email or voice mail message.
"Next"	#	Reads the next email or voice mail message.
"Next unread"	00 followed by ##	Reads the next unread email message. Available only for email.
"Delete"	7	Deletes the current email or voice mail message.
"Reply"	8	Replies to the user who sent the current email or voice mail message.
"Reply all"	00 followed by 88	Replies to all the users on the current email message. Not an available option for voice mail messages.
"Mark as unread"	9	Marks the email message as Unread.
"End"	33	Stops reading and goes to the end of the current email or voice mail message.
"More options"	00	Opens the More Options menu.
"Previous"	00 followed by 11	Reads the previous email or voice mail message.
"Read the header"		Reads the header of the email or voice mail message.
"Call sender"	00 followed by 2	Places a call to the user who

		sent the current email or voice mail message.
"Forward"	00 followed by 6	Forwards the current email or voice mail message to other email recipients or groups.
"Flag for follow-up"	00 followed by 44	Marks or flags the current email or voice mail message for follow-up.
"Find by name"		Uses the user's name to locate email or voice mail messages in the user's mailbox.
"Delete conversation"	00 followed by 77	Deletes all the email messages that are associated with an email conversation. Available only for email.
"Hide conversation"	00 followed by 99	Hides additional email messages that are contained within the same email conversation. Available only for email.
"Envelope information"	00 followed by 5	Reads the envelope information for the email or voice mail message.
"Select language"	00 followed by 55	Lets you select the language in which you want the email or voice mail message to be read.
"Rewind" or "Repeat"	1	Rewinds or repeats the current email or voice mail message. Available only while the

		message is being played.
"Pause"	2	Pauses the current email or voice mail message. Available only while the message is being played.
"Fast forward"	3	Fast forwards the current email or voice mail message. Available only while the message is being played.
"Slow down"	4	Plays or reads the current email or voice mail message more slowly. Available only while the message is being played.
"Faster"	6	Plays or reads the current email or voice mail message faster. Available only while the message is being played.
"Previous"	11	Reads the previous email message from the beginning. Available only for email.
"Replay"	00 followed by 1	Replays the current email or voice mail message.
"Repeat"	0	Repeats the current menu options.
"Main menu"	*	Exits to the main menu.

Important:
If you need to access an email message after you delete it using Outlook Voice Access, you can use Outlook Web App or Outlook to move the email message back into the appropriate folder

from the Deleted Items folder. You can't use Outlook Voice Access to access the Deleted Items folder.

Calendar options command reference

As an Outlook Voice Access user, when you dial in to an Outlook Voice Access number, you're presented with menu options that enable you to access your mailbox and manage your calendar. The following table lists the commands that are available for managing your calendar.

Calendar commands

Voice command	Touchtone command	Description
"Next"	#	Reads the next calendar appointment.
"Next day"	##	Opens and reads the calendar appointments for the next day.
"Repeat"	0	Repeats the menu options that are available. Or, if you're using the VUI, the system reads the calendar appointment again.
"More options"	00	Plays the more calendar options menu.
"Repeat"	1	Reads the calendar appointment again.
"Previous meeting"	00 followed by 11	Opens the previous meeting that's scheduled.
"Call location"	2	Calls the telephone number that's listed for the meeting location.
"Call organizer"	00 followed by 22	Calls the telephone number that's listed for the organizer of the meeting.

"I'll be late"	3	Sends an I'll be late message to all the meeting attendees.
"Accept" or "Tentative accept"	4	Accepts or tentatively accepts the meeting request.
"Meeting details"	5	Reads or plays back the details of the meeting that's currently being read.
"Attendance details"	00 followed by 55	Reads or plays the details of a meeting that's scheduled.
"Forward"	00 followed by 6	Forwards a meeting request for the meeting to another user.
"Decline" or "Cancel"	7	Declines or cancels the meeting request.
"Clear my calendar"	00 followed by 77	Clears your calendar for a specific time period for that day.
"Reply"	00 followed by 8	Replies to the meeting organizer.
"Reply all"	00 followed by 88	Replies to all the meeting attendees.
"Repeat menu"	5 followed by 0	Repeats the menu options that are available.
"Rewind"	5 followed by 1	Rewinds the meeting details.
	5 followed by 11	Returns to the beginning of the meeting details.
	5 followed by 2	Pauses and resumes playback of the meeting details.

"Fast forward"	5 followed by 3	Skips forward within the meeting details.
"End"	5 followed by 33	Skips to the end of the meeting details.
	5 followed by 4	Plays or reads the meeting details slower.
	5 followed by 55	Selects the language that will be used to read the meeting details.
	5 followed by 6	Plays or reads the meeting details faster.
"Main menu"	*	Exits to the main menu.

Find a contact commands reference

As an Outlook Voice Access user, when you dial in to an Outlook Voice Access number, you're presented with menu options that enable you to access your mailbox, change personal options, or call or send a message to a contact. If you choose to use your voice, which is selected by default, and select the contacts menu option, the voice mail system you to use the telephone keypad to navigate the find a contact options. You can also locate a user in the directory or a contact by using the telephone keypad. The following table lists the commands that are available for managing your contacts or searching for a user.

Contact commands

Voice command	Touchtone command	Description
"Directory"	00	Searches the directory for a user.
"Play details"	1	Plays the details of the personal contact, such as the telephone numbers that are listed for the personal contact.

"Send a message"	3	Sends a message to the personal contact that's selected.
"Find another contact"	4	Finds another personal contact.
"Call the cell"	2 followed by 1	Calls the mobile telephone number that's listed for the personal contact.
"Call the office"	2 followed by 2	Calls the business or office telephone number that's listed for the personal contact.
"Call home"	2 followed by 3	Calls the home telephone number that's listed for the personal contact.
	##	Lets you enter the email alias or name for the user in the directory if using the directory search feature.
"Main menu"	*	Exits to the main menu.

Personal options commands reference

As an Outlook Voice Access user, when you dial in to an Outlook Voice Access number, you're presented with menu options that enable you to access your mailbox and manage your personal options. When you configure personal options using Outlook Voice Access, you can only use the telephone keypad to navigate the menus. Using your voice to navigate the menus is not available for configuring personal options. The following table lists the commands that are available for managing your personal options.

Personal options commands

Voice command	Touchtone command	Description
	1	Turns on or off the telephone Out of Office greeting.

	2	Records the personal voice mail or Out of Office voice mail greeting.
	3	Changes the PIN that's used for Outlook Voice Access.
	4	Starts using the VUI or touchtone interface.
	5	Sets the local time zone to use.
	6	Chooses the 12-hour or 24-hour time format.
	*	Returns to the main menu.
	0	Repeats the menu options that are available.

For more information

Setting up Outlook Voice Access

Set up client voice mail features

Navigating menus with Outlook Voice Access

Topic Last Modified: *2013-04-23*

Outlook Voice Access is a feature in Unified Messaging (UM) that enables users to retrieve email and voice mail messages and manage their calendar and personal contacts by using an analog, digital, or mobile telephone. They can interact with their mailbox using their telephone keypad or voice commands, but must use the keypad on their telephone to search for a user in the directory for your organization.

When UM-enabled users call in to an Outlook Voice Access number, they can sign in to their mailbox using a telephone and are presented with a series of voice prompts. These voice prompts help them navigate the voice mail system menus and enable them to access their mailbox. Outlook Voice Access lets users do the following:

- Retrieve, listen to, reply to, create, and forward voice or email messages.
- Listen to or change calendar information.
- Change personal options, such as a PIN, or call or send a voice message to a personal contact.

An Outlook Voice Access number is assigned to a user when they're enabled for UM. The user can find an Outlook Voice Access number to access their mailbox in the welcome message that's sent to them when they're enabled for UM or by signing in to their mailbox using Outlook Web App, going to **Options** > **Telephone**, and locating the Outlook Voice Access number or numbers in the **Outlook Voice Access** section.

After a user enters their extension number and PIN, the voice mail system will let them know how many new voice mail and email messages they have and when their next meeting is. After the voice mail system has played this prompt, an Outlook Voice Access main menu will be read to the user and the user can say one of the following:

- Voice mail
- Email
- Calendar
- Personal options

Contents

Reading and reviewing email

Users can listen to, reply to, create, and forward unread email messages using the telephone. For example, if a user is expecting an important email message, and does not have access to the Internet, they can use a mobile telephone to dial an Outlook Voice Access number.

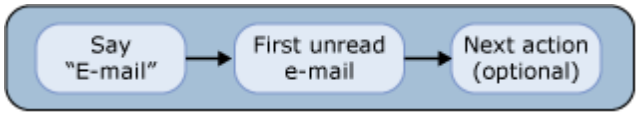
Listen to email messages

To listen to email messages using their voice, the user must dial an Outlook Voice Access number, enter their extension number and PIN, and then do the following:

1. Say "Email" to access their email.
2. The voice mail system will read the name, subject, time, and priority of the first unread email message.
3. The user can then say one of the following options:
   - "Next message" to mark the message as Read and go to the next email message.

- "Mark unread" to keep the message marked as Unread and go to the next message.
- "End" to jump to the end of the message.
- "Delete" to delete the message.

This process is shown in the following figure.



To listen to email messages using the telephone keypad, users must dial an Outlook Voice Access number, enter their extension number and PIN, and then do the following:

1. Press 2 to access their email.
2. The voice mail system will read the name, subject, time, and priority of the first unread email message.
3. The user can then press one of the following options:
   - Pound (#) key to mark the message as Read and go to the next email message.
   - 9 to keep the message marked as Unread and go to the next message.
   - 33 to jump to the end of the message.
   - 7 to delete the message.

This process is shown in the following figure.

Return to top

Reply to email messages

To listen to email messages and then reply using their voice, users must do the following:

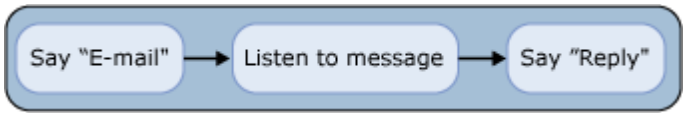
1. Say "Email."
2. Say "Next message" repeatedly until they reach the email message to which they want to reply.
3. Listen to the message or say "End" to go to the end of the message.
4. Say one of the following:
   - "Reply" to reply to the sender.
   - "Reply all" to reply to the sender and all other recipients.
   - "Forward" to forward the message to another user or group.
5. Record a reply and then hang up, remain silent, or press any key. To accept the reply message and send it, say "Send it."

This process is shown in the following figure.

To listen to email messages and then reply using the telephone keypad, users must do the following:

1. Press 2.
2. Press # repeatedly until they reach the email message to which they want to reply.
3. Listen to the message or press 33 to go to the end of the message.
4. Press one of the following:
   - 8 to reply to the sender.
   - 88 to reply to the sender and all other recipients.
   - 6 to forward the message to another user or group.
5. Record a reply, and then press #. To accept the reply message and send it, press 1.

This process is shown in the following figure.



Return to top

Listen to the next unread email message

To listen to an email message and then go to the next unread message using their voice, users must do the following:

1. Say "Email."
2. Say "Next unread." Say "Mark unread" if they want to mark the message as Unread.

This process is shown in the following figure.

To listen to an email message and then go to the next unread message using the telephone keypad, users must do the following:

1. Press 2.
2. Press ## to listen to the next unread message. Press 9 to mark the message as Unread.

This process is shown in the following figure.

Return to top

Flag an email message for follow-up

To listen to email messages and flag messages for follow-up using their voice, users must do the following:

1. Say "Email."
2. Say "Next message" repeatedly until they reach the email message that they want to flag for follow-up. Say "Mark unread" to mark the message as Unread.

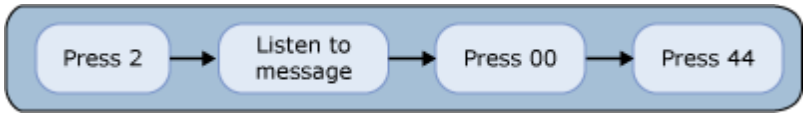
3. Listen to the message or say "End" to go to the end of the message.
4. Say "Flag" or "Flag for follow-up" to flag the message for follow-up.

This process is shown in the following figure.

To listen to email messages and flag messages for follow-up using the telephone keypad, users must do the following:

1. Press 2.
2. Press # repeatedly until they reach the email message that they want to flag for follow-up. Press 9 to mark the message as Unread.
3. Listen to the message or press 33 to go to the end of the message.
4. Press 0 (zero) twice to access more options.
5. Press 44 to flag the message for follow-up.

This process is shown in the following figure.



Return to top

Hide a conversation

To listen to email messages and hide a conversation so that the voice mail system will not continue to read other email messages that are in the same email conversation using their voice, users must do the following:

1. Say "Email."
2. Say "Next message" repeatedly until they reach the email message that they want. Say "Mark unread" to mark the message as Unread.
3. Listen to the message or say "End" to go to the end of the message.
4. Say "Hide" or "Hide conversation" to hide the conversation. The next email message from a different conversation will be read.

This process is shown in the following figure.

To listen to email messages and hide a conversation so that the voice mail system will not continue to read other email messages that are in the same email conversation using the telephone keypad, users must do the following:

1. Press 2.
2. Press # repeatedly until they reach the email message that they want to hide. Press 9 to mark the message as Unread.

3. Listen to the message or press 33 to go to the end of the message.
4. Press 99 to hide the conversation. The next email message from a different conversation will be read.

This process is shown in the following figure.

Note:

When a conversation is hidden, it is hidden only for the current session. If users sign out and then sign in to their mailbox again, the voice mail system will read email messages that are in the same conversation.

Return to top

Managing calendar meetings and appointments

Users can listen to, reply to, create, and forward meeting requests and appointments in their calendar over the telephone.

For example, a user has a meeting at 10:00 A.M. However, because of some unexpected delays, the user will be 15 minutes late. The user can inform the other meeting attendees by calling the telephone number for Outlook Voice Access, signing in to their mailbox, and then accessing the list of meetings for that day in the calendar. After the voice mail system reads the meeting request for the 10:00 A.M. meeting, the user can use the *I'll be late* feature to inform all the meeting attendees that the user will be 15 minutes late. Each attendee will receive an email message that informs them that the user will be 15 minutes late. The user also has the option to attach a voice mail message.

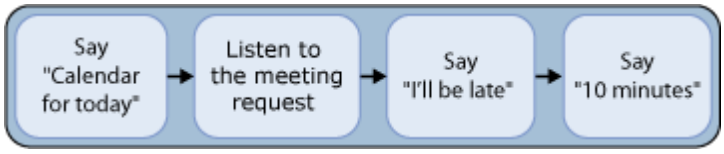
In another example, a user may have an important client who decides to schedule an all-day meeting on very short notice. The user must cancel all other meetings for that day in the simplest possible way. Using the *Clear my calendar* feature, users can quickly and easily clear their calendar for the whole day.

Send an I'll be late message

To send an I'll be late message to meeting participants using their voice, users must dial an Outlook Voice Access number, enter their extension number and PIN, and then do the following:

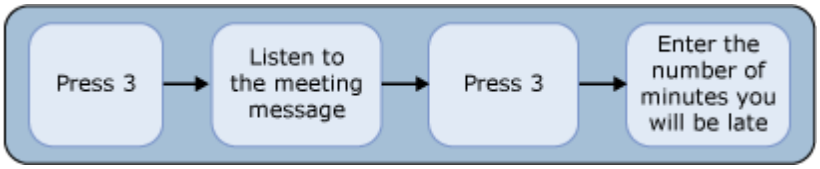
1. Say "Calendar for today" to access their calendar.
2. Listen to the meeting requests to locate the meeting for which to send an I'll be late message.
3. After the meeting request has been read, say "I'll be late."
4. The voice mail system asks, "How late?" Say "10 minutes."
5. The voice mail system asks, "Do you want to record a message?" If so, say "Yes," record the message, and then say "Send it." If not, say "No."

This process is shown in the following figure.



To send an I'll be late message to meeting participants using the telephone keypad, users must dial an Outlook Voice Access number, enter their extension number and PIN, and then do the following:

1. Press 3 to access their calendar.
2. Listen to the meeting requests to locate the meeting for which to send an I'll be late message.
3. After the meeting request has been read, press 3.
4. The voice mail system asks, "How late?" Enter 10 on the telephone key pad.



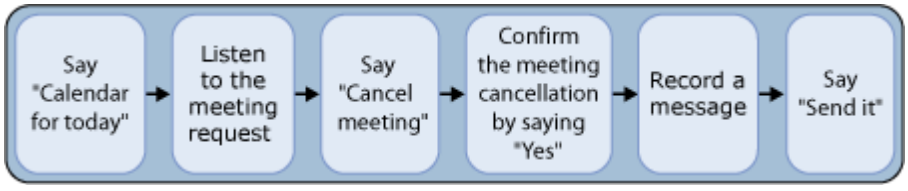
Return to top

Cancel a meeting

To cancel a meeting, the user must be the meeting organizer. To cancel the meeting using their voice, meeting organizers must do the following:

1. Say "Calendar for today."
2. Listen to the meeting requests to locate the meeting to cancel.
3. After the meeting request has been read, say "Cancel meeting."
4. Confirm the meeting cancellation by saying "Yes."
5. If the meeting organizer chooses to send a voice message, they can then say "Yes," record the message, and then say "Send it."

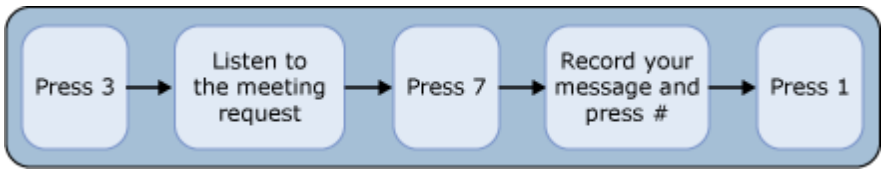
This process is shown in the following figure.



To cancel a meeting, the user must be the meeting organizer. To cancel the meeting using the telephone keypad, meeting organizers must do the following:

1. Press 3.
2. Listen to the meeting requests to locate the meeting to cancel.
3. Press 7 to cancel the meeting.
4. If the meeting organizer chooses to send a voice message, they can then press one of the following options:
   - # to stop recording the message.
   - 1 to accept the recorded message.

This process is shown in the following figure.



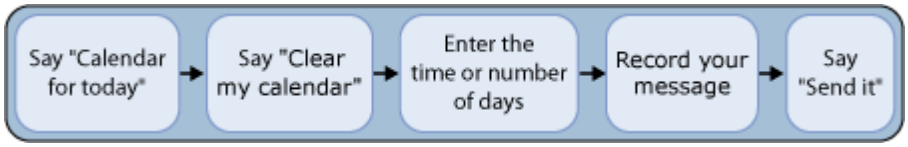
Return to top

Clear a calendar

To clear their calendar using their voice, users must do the following:

1. Say "Calendar for today."
2. Say "Clear my calendar."
3. Enter the time or the number of days to be cleared.
4. The voice mail system asks whether they want to attach a recorded voice message. If so, say "Yes," record the message, and then say "Send it." If not, say "No."

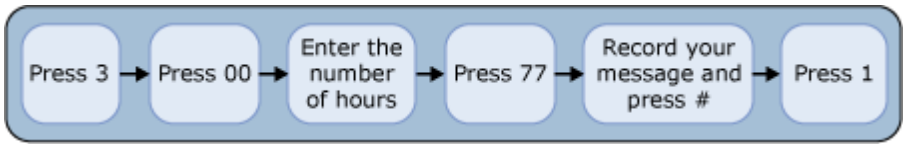
This process is shown in the following figure.



To clear their calendar using the telephone keypad, users must do the following:

1. Press 3.
2. Press 00 to go to the More Options menu.
3. Press 77 to clear their calendar.
4. Enter the number of hours to clear from the calendar.
5. If users choose to send a voice message, they can do one of the following:
   - Press # to not send a voice message.
   - Record the voice message when prompted, press # to stop recording the message, and then press 1 to accept the recorded message.

This process is shown in the following figure.



Return to top

Accept a meeting request

To accept a meeting request using their voice, users must do the following:

1. Say "Email" to access their email.
2. Listen to the email message that contains a meeting request.
3. Say "Accept" to accept the meeting request.

This process is shown in the following figure.

To accept a meeting request using the telephone keypad, users must do the following:

1. Press 2 to access their email.
2. Listen to the email message that contains a meeting request.
3. Press 4 to accept the meeting request.

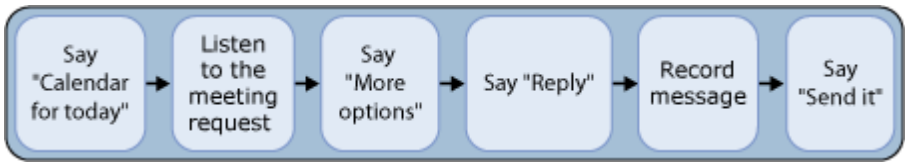
This process is shown in the following figure.

Reply to a meeting request

To reply to a meeting request using their voice, users must do the following:

1. Say "Calendar for today."
2. Listen to the meeting requests to locate the meeting request to reply to.
3. Say "More options" to open the More Options menu.
4. Say "Reply" to reply to the meeting organizer.
5. Record a message.
6. Say "Send it."

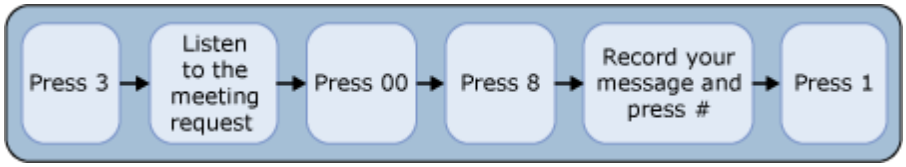
This process is shown in the following figure.



To reply to a meeting request using the telephone keypad, users must do the following:

1. Press 3.
2. Listen to the meeting requests to locate the meeting request to reply to.
3. Press 00 for more options.
4. Press 8 to reply to the meeting organizer.
5. Record a message, and then press #.
6. Press 1 to accept the recording and send the message.

This process is shown in the following figure.



Return to top

Managing personal options and contacts

Users can manage their personal options and contacts using Outlook Voice Access. They can:

- Call a personal contact.
- Locate and call a user in the directory.
- Configure personal options, such as changing their PIN over the telephone.

When users first set up their mailbox, they must create personal and Automatic Replies greetings that callers will hear when users are unable to answer their telephone. If, for example, users realize that they have forgotten to turn on an Automatic Replies voice greeting that will give callers an alternative number to call if they have an immediate issue, users can use Outlook Voice Access to access their personal options and record and turn on an Automatic Replies greeting from any telephone.

If a user has to contact an account manager with important information about a client, the user can call the number that is used for Outlook Voice Access, use the directory search feature using their telephone keypad to locate the account manager, and then place the call.

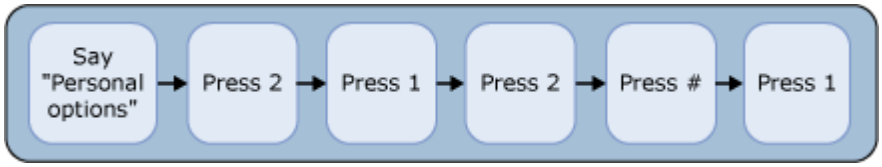
Note:
When users access the Personal Options menu, they must use the telephone keypad.

Record a personal greeting

To record a personal greeting using their voice, users must dial an Outlook Voice Access number, enter their extension number and PIN, and then do the following:

1. Say "Personal options" to access personal options.
2. Press 2 to record greetings.
3. Press 1 to record a personal greeting. Press 2 if they need to re-record the personal greeting.
4. Press # to stop recording the personal greeting.
5. Press 1 to accept the personal greeting.

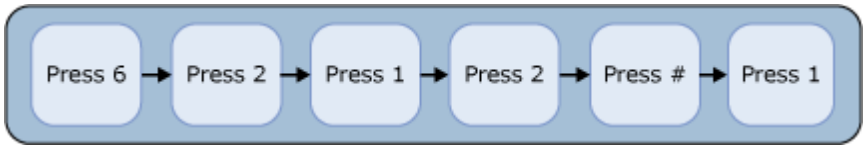
This process is shown in the following figure.



To record a personal greeting using the telephone keypad, users must dial an Outlook Voice Access number, enter their extension number and PIN, and then do the following:

1. Press 6 to access personal options.
2. Press 2 to record greetings.
3. Press 1 to record a personal greeting. Press 2 if they need to re-record the personal greeting.
4. Press # to stop recording the personal greeting.
5. Press 1 to accept the personal greeting.

This process is shown in the following figure.



Note:

When users change their telephone greeting, they are also given the option to turn on or off their email automatic reply message.

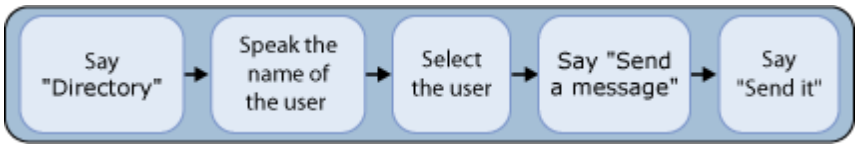
Return to top

Send a voice message to a user

To locate and send a voice message to another UM-enabled user using their voice, users must do the following:

1. Say "Directory."
2. Say the name of the person to locate.
3. Select the correct person from the list.
4. Say "Send a message," and then record the voice message.
5. Say "Send it" to send the message.

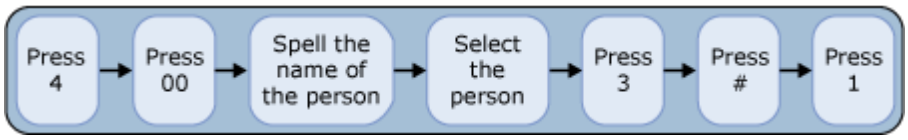
This process is shown in the following figure.



To locate and send a voice message to another UM-enabled user using the telephone keypad, users must do the following:

1. Press 4 to search for a contact.
2. Press 00 to locate the person in the directory.
3. Use the telephone keypad to spell the name of the person to locate.
4. Select the correct person from the list.
5. Press 3 to send a voice message to the person.
6. Record the voice message, and then press # to stop recording.
7. Press 1 to accept the voice message and send it.

This process is shown in the following figure.



Change a PIN

To change their PIN using their voice, users must do the following:

1. Say "Personal options."

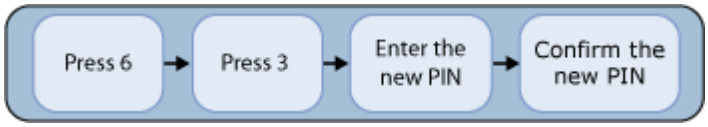
2. Press 3 to change the PIN.
3. Enter the new PIN, and then press #.
4. Press # to confirm the new PIN.

This process is shown in the following figure.

To change their PIN using the telephone keypad, users must do the following:

1. Press 6 to access personal options.
2. Press 3 to change the PIN.
3. Enter the new PIN, and then press #.
4. Press # to confirm the new PIN.

This process is shown in the following figure.



Return to top

Play on Phone

Unified Messaging > Set up client voice mail features > Setting up Outlook Voice Access >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-25*

After a voice mail message arrives, users can choose either to listen to the voice mail message through their computer speakers or headphones or to use the Play on Phone feature. The Play on Phone feature is included with Microsoft Outlook and Outlook Web App, and settings for Play on Phone are available in the **Play on phone** section under **Voice mail** options. This topic discusses how a Unified Messaging (UM)-enabled user can use the Play on Phone feature.

What is Play on Phone?

The Play on Phone feature lets UM-enabled users play voice messages over a telephone. If a UM-enabled user sits in an office cubicle, is using a public computer or a computer that's not enabled for multimedia, or is listening to a voice message that's confidential, the user might not want to—or be able to—listen to a voice message through their computer speakers. Alternatively, they can play back the voice mail message using any telephone, including home, office, or mobile phones. To review settings for Play on Phone, in Outlook, go to **File** > **Info** > **Manage voice mail**. Clicking the **Manage voice mail** button will automatically sign you in to Outlook Web App. or you can sign in to

Outlook Web App using a web browser. In Outlook Web App, go to **Options** > **Phone** > **Voice Mail** > **Play on Phone** section on the **Voice Mail** page.

When the user clicks the Play on Phone toolbar option in the voice mail form, the **Play on Phone** dialog box appears. The **Play on Phone** box provides the controls for selecting or inputting the telephone number to use to play a voice message, starting and ending the call, and a status message for monitoring the call. If the user is linked to a SIP URI dial plan, their SIP address will appear in the **Dial** box. If they are linked to an E.164 dial plan, their full E.164 number will appear in the **Dial** box.

Note:

Only one voice message can be played at a time. If the user tries to start a second Play on Phone call while a previous call is still in progress, an error message will appear.

Most recently used telephone number list

Users can see a list of telephone numbers they used most recently in the **Dial** box. The telephone number specified in the **Play on phone** section is always displayed as the top entry and is automatically selected for the user as the primary number. Users can use the drop-down menu to select other telephone numbers to dial instead of the telephone number that's configured as the primary number.

Note:

To enable users who are using the Play on Phone feature to dial an external telephone number without using an outside line access code, for example 425-555-1234 instead of 9-425-555-1234, configure in-country/region dialing rules on a UM dial plan that include the following line: group1, 9xxxxxxxxx, 91xxxxxxxxx. After you've configured the in-country/region dialing rules, add this list to the UM mailbox policy.

Play on Phone buttons

The **Play on Phone** dialog box gives users the option to **Dial** and **Hang-up**. When the **Play on Phone** dialog box is first opened, the **Dial** button is enabled and the **Hang-up** button is disabled. After a call is placed, the **Dial** button becomes disabled until the call has ended. The call can be ended either by clicking the **Hang-up** button or by physically hanging up the telephone. Closing the **Play on Phone** dialog box using the **Close** button ends the call if one is in progress. The **Play on Phone** option and other options are also available in **Reading pane** preview in Outlook. If you open the voice mail message in a separate window, the **Play on Phone** button is on the toolbar.

Subject, sent, and status section

The bottom section of the **Play on Phone** dialog box displays the subject of the voice message, the date and time sent, and a message that displays the current state of the call. Any errors specific to

the Play on Phone operation are displayed to the user in this section of the **Play on Phone** dialog box.

Phone number validation

Play on Phone performs only simple validation on input into the **Play on Phone** dialog box. Play on Phone does not validate telephone numbers. If a telephone number is not valid, Unified Messaging returns a meaningful error code to the user.

Outlook Voice Access procedures

Unified Messaging > Set up client voice mail features > Setting up Outlook Voice Access >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-15*

Enable or disable Outlook Voice Access for users

Configure an Outlook Voice Access number

Disable selected features for Outlook Voice Access users

Set mailbox features for Outlook Voice Access users

Set mailbox features for an Outlook Voice Access user

Enable or disable automatic speech recognition for an Outlook Voice Access user

Enable an informational announcement for Outlook Voice Access users

Enable a customized greeting for Outlook Voice Access users

Enable or disable Play on Phone for Outlook Voice Access users

Enable or disable sending voice messages from Outlook Voice Access

Enable or prevent transferring calls from Outlook Voice Access

Configure the group of users that Outlook Voice Access users can contact

Configure the primary way for Outlook Voice Access users to search

Configure the secondary way for Outlook Voice Access users to search

Configure the number of sign-in failures before Outlook Voice Access users are disconnected

Configure the number of input failures before Outlook Voice Access users are disconnected

Configure the limit on personal greetings for Outlook Voice Access users

Enable or disable Outlook Voice Access for users

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-12*

You can enable or disable access to Outlook Voice Access for UM-enabled users who are associated with a Unified Messaging (UM) mailbox policy. Outlook Voice Access is a feature used by UM-enabled users to access their mailbox over a phone. By default, this setting is enabled.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or disable Outlook Voice Access

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. Under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .

3. On the **UM Mailbox Policy** page, select or clear the check box next to **Allow Outlook Voice Access**.
4. Click **Save**.

Use the Shell to enable or disable Outlook Voice Access

This example allows users who are associated with the UM mailbox policy `MyUMMailboxPolicy` to use Outlook Voice Access.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowSubscriberAccess $true
```

This example prevents users who are associated with the UM mailbox policy `MyUMMailboxPolicy` from using Outlook Voice Access.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowSubscriberAccess $false
```

Configure an Outlook Voice Access number

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

An Outlook Voice Access number lets a user who is enabled for Unified Messaging (UM) and voice mail access their mailbox using Outlook Voice Access. When you configure an Outlook Voice Access or subscriber access number on a dial plan, UM-enabled users can call in to the number, sign in to their mailbox, and access their email, voice mail, calendar, and personal contact information.

By default, when you create a UM dial plan, an Outlook Voice Access number isn't configured. To configure an Outlook Voice Access number, you first need to create the dial plan, and then configure an Outlook Voice Access number under the dial plan's **Outlook Voice Access** option. Although an Outlook Voice Access number isn't required, you need to configure at least one Outlook Voice Access number to enable a UM-enabled user to use Outlook Voice Access to access their mailbox. You can configure multiple Outlook Voice Access numbers for a single dial plan.

Outlook Voice Access numbers can contain alphabetical, numeric, and special characters, separators, and spaces. For example:

- +14255551010

- +1-425-555-1010
- 4255551010
- +1 425 555 1010
- 1-800-555-CALL

For more information about the menu options available for Outlook Voice Access users, see the Quick Reference Guide for Outlook Voice Access, which is available from the Microsoft Download Center.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure an Outlook Voice Access number

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to modify and on the toolbar, click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. In **Outlook Voice Access**, under **Outlook Voice Access numbers**, use the box to enter the number you want to use, and then click **Add +**.
5. Click **Save**.

Use the Shell to configure an Outlook Voice Access number

This example sets the Outlook Voice Access number to 4255550100 for a UM dial plan named

MyUMDialPlan.

```
Set-UMDialPlan -identity MyUMDialPlan -
AccessTelephoneNumbers 4255550100
```

Disable selected features for Outlook Voice Access users

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

Outlook Voice Access contains two interfaces: the telephone user interface (TUI) and the voice user interface (VUI). By default, when users dial in to Outlook Voice Access, they can access their calendar, email, and personal contacts, and search the directory. You can use the Shell to prevent users from accessing one or more of these features when they use Outlook Voice Access to access their mailbox. When you modify Outlook Voice Access features on a Unified Messaging (UM) mailbox policy, your changes affect all users who are associated with the UM mailbox policy.

You can disable users' access to the following Outlook Voice Access features on a UM mailbox policy:

- Calendar
- Directory
- Email
- Personal contacts

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

You can also use the Shell to disable Outlook Voice Access features on the mailbox of a single UM-enabled user. When you do this, the features will be disabled only for that user. Although you can't disable all the Outlook Voice Access features that are found on a UM mailbox policy for a single user, you can disable access to their calendar and to their email.

For additional management tasks related to UM mailboxes, see Voice mail for users.

Note:
You can use only the Shell to modify the Outlook Voice Access features for UM-enabled users on a UM mailbox policy or on the mailbox of a single UM-enabled user.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that a user has been enabled for UM. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to disable selected Outlook Voice Access features for UM-enabled users on a UM mailbox policy

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.

This example prevents users associated with a UM mailbox policy named `MyUMMailboxPolicy` from accessing their calendar when they dial in to Outlook Voice Access.

```
Set-UMMailboxPolicy -id MyUMMailboxPolicy -
AllowTUIAccessToCalendar $false
```

This example prevents users associated with the UM mailbox policy named `MyUMMailboxPolicy` from accessing the directory when they dial in to Outlook Voice Access.

```
Set-UMMailboxPolicy -id MyUMMailboxPolicy -
AllowTUIAccessToDirectory $false
```

This example prevents users associated with the UM mailbox policy named `MyUMMailboxPolicy` from accessing their email when they dial in to Outlook Voice Access.

```
Set-UMMailboxPolicy -id MyUMMailboxPolicy -
AllowTUIAccessToEmail -$false
```

This example prevents users associated with the UM mailbox policy named `MyUMMailboxPolicy` from accessing personal contacts when they dial in to Outlook Voice Access.

```
Set-UMMailboxPolicy -id MyUMMailboxPolicy -
AllowTUIAccessToPersonalContacts $false
```

Use the Shell to disable selected Outlook Voice Access features on the mailbox of a single UM-enabled user

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.

This example disables access to the calendar on a UM mailbox named tony@contoso.com when the user dials in to Outlook Voice Access.

```
Set-UMMailbox -id tony@contoso.com -
TUIAccessToCalendarEnabled $false
```

This example disables access to email on a UM mailbox named tony@contoso.com when the user dials in to Outlook Voice Access.

```
Set-UMMailbox -id tony@contoso.com -TUIAccessToEmailEnabled
$false
```

Set mailbox features for Outlook Voice Access users

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

Outlook Voice Access contains two interfaces: a telephone user interface (TUI) and a voice user interface (VUI). You can configure a UM-enabled user's TUI settings when the user accesses a mailbox using the Unified Messaging (UM) system in Exchange 2013. When you modify a UM-

enabled user's TUI settings on a UM mailbox policy, the changes affect all users who are associated with the UM mailbox policy. You can modify the following TUI settings on a UM mailbox policy:

- PIN-less access to voice mail
- Voice responses to other messages
- TUI access to their calendar
- TUI access to the directory
- TUI access to their email
- TUI access to their personal contacts

Note:

You can use only the Shell to modify the Outlook Voice Access TUI settings for UM-enabled users.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to modify TUI settings on a UM mailbox policy

This example sets TUI-related settings on a UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailbox -identity MyUMMailboxPolicy -
AllowSubscriberAccess $true -AllowTUIAccessToCalendar
$false -AllowTUIAccessToDirectory $false -
AllowTUIAccessToEmail -$true -
AllowTUIAccessToPersonalContacts $true
```

Set mailbox features for an Outlook Voice Access user

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

Telephone user interface (TUI) settings are used when a user accesses the Unified Messaging (UM) system by using Outlook Voice Access. When you modify a UM-enabled user's TUI configuration settings, you modify properties and their values on the UM-enabled user's mailbox.

You can change the following TUI settings for a UM-enabled user:

- Allow subscriber access
- Allow TUI access to the calendar
- Allow TUI access to email
- Allow Automatic Speech Recognition

For additional management tasks related to UM users, see Set mailbox features for an Outlook Voice Access user.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that the existing Exchange recipient is enabled for Unified Messaging and voice mail. For detailed steps, see Enable a user for voice mail.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to modify a single UM-enabled user's TUI

settings

This example enables calendar and email access using the TUI for a UM-enabled user named Tony Smith.

```
Set-UMMailbox -Identity tony@contoso.com TUIAccessToCal
True -TUIAccessToEmail True -OperatorNumber 111111 -
DisableMissedCallNotification False -AnonCallBlock True
```

Note:

TUI settings for users are also available on UM mailbox policies. Modifying TUI settings on a UM mailbox policy affects all users who are associated with the UM mailbox policy. For more information about how to modify TUI settings on a UM mailbox policy, see Set mailbox features for Outlook Voice Access users.

Enable or disable automatic speech recognition for an Outlook Voice Access user

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can configure Automatic Speech Recognition (ASR) for a user who's enabled for Unified Messaging (UM) and voice mail. When ASR is enabled on the mailbox of an Outlook Voice Access user, the user can move through the mailbox menus using voice commands. ASR is enabled by default. If ASR is disabled, the user must use dual tone multi-frequency (DTMF), also known as touchtone, inputs to move through the menus.

Note:

You can't use the EAC to configure this feature. You must use the Shell to enable or disable ASR for a voice mail user.

For additional management tasks related to UM or voice mail users, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to enable or disable ASR for a UM-enabled user

This example enables ASR for a UM-enabled user named `tonysmith`.

```
Set-UMMailbox -Identity tonysmith@contoso.com -
AutomaticSpeechRecognitionEnabled $true
```

This example disables ASR for a UM-enabled user named `tonysmith`.

```
Set-UMMailbox -Identity tonysmith@contoso.com -
AutomaticSpeechRecognitionEnabled $false
```

Enable an informational announcement for Outlook Voice Access users

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-30*

You can enable an informational announcement on a Unified Messaging (UM) dial plan.

Informational announcements are used for general announcements that change more frequently than the welcome greeting does, or for announcements that are required by corporate compliance policies.

By default, callers, including Outlook Voice Access users who dial in to an Outlook Voice Access number that's been configured, don't hear an informational announcement. If you want one to be played, you must create a .wav or .wma file to use for the informational announcement after you create a UM dial plan, and then enable the informational announcement on the dial plan.

When it's important that the whole informational announcement is heard, you can configure the announcement to be uninterruptible. This prevents a caller from pressing a key or speaking a command to interrupt and stop the announcement.

For more information about the menu options that are available for Outlook Voice Access users, see the Quick Reference Guide for Outlook Voice Access, which is available from the Microsoft Download Center.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable an informational announcement

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan that you want to modify, and then click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. In **Outlook Voice Access**, under **Informational announcement**, click **Change**, and then click **Browse** to locate the announcement file.

Important:

The file you use for the informational announcement must be a .wav or .wma file.

5. After you've located the file, click **Open**, and then click **Save**.

Use the Shell to enable an informational announcement

This example enables an informational announcement that uses the informational.wav informational announcement file on a UM dial plan named `MyUMDialPlan`.

```
Set-UMDialPlan -Identity MyUMDialPlan -
InfoAnnouncementEnabled $true-InfoAnnouncementFilename c:
\UMGreetings\informational.wav
```

Enable a customized greeting for Outlook Voice Access users

Set up client voice mail features > Setting up Outlook Voice Access > Outlook Voice Access procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-03-15*

By default, each Unified Messaging (UM) dial plan uses a standard .wav file for the welcome greeting that's played to callers, including Outlook Voice Access users who dial in to an Outlook Voice Access number that's been configured. However, you can create a .wav or .wma file for the welcome greeting, and then enable it on the UM dial plan.

For example, you might want to change the default welcome greeting and instead provide a welcome greeting that's specific to your company, such as "Welcome to Outlook Voice Access for Woodgrove Bank." To do this, you record the customized welcome greeting and save it as a .wav or .wma file. Then you configure the dial plan to use the customized welcome greeting.

For more information about the menu options available for Outlook Voice Access users, see the Quick Reference Guide for Outlook Voice Access, which is available from the Microsoft Download Center.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.

- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable a customized welcome greeting

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan that you want to modify, and then click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. In **Outlook Voice Access**, under **Welcome greeting**, click **Change**, and then click **Browse** to locate the greeting file.

Important:
The file you use for the welcome greeting must be a .wav or .wma file.

5. After you've located the file, click **Open**, and then click **Save**.

Use the Shell to enable a customized welcome greeting

This example enables a welcome greeting that uses the C:\UMPrompts\welcome.wav file on a UM dial plan named `MyUMDialPlan`.

```
Set-UMDialPlan -Identity MyUMDialPlan -
WelcomeGreetingEnabled $true -WelcomeGreetingFilename c:
\UMPrompts\welcome.wav
```

Enable or disable Play on Phone for Outlook Voice Access users

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-12*

You can enable or disable the Play on Phone feature for users associated with a Unified Messaging (UM) mailbox policy. This option is enabled by default and allows users to play their voice mail messages over any phone. This option isn't available to UM-enabled users who have a mailbox on a Microsoft Exchange Server 2007 server.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or disable Play on Phone

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page, select or clear the check box next to **Allow Play on Phone for voice mail**.
4. Click **Save**.

Use the Shell to enable or disable Play on Phone

This example enables the Play on Phone feature for users who are associated with the UM mailbox policy `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
```

```
AllowPlayOnPhone $true
```

This example disables the Play on Phone feature for users who are associated with the UM mailbox policy `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowPlayOnPhone $false
```

Enable or disable sending voice messages from Outlook Voice Access

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-13*

You can enable Outlook Voice Access users to send voice mail messages to other UM-enabled users who are associated with the same dial plan, or prevent them from doing so.

By default, this setting is enabled. If you disable this setting, Outlook Voice Access users that call into an Outlook Voice Access number won't be able to send voice messages to users within the same dial plan.

For additional tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or prevent Outlook Voice Access users sending voice messages to users in the same dial plan

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to change, and then click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. In **Transfer & search**, under **Allow callers to**, select **Leave voice messages without ringing a user's phone** to allow sending voice messages. If you want to prevent sending voice messages for users, clear this setting.
5. Click **Save**.

Use the Shell to enable or prevent Outlook Voice Access users sending voice messages to users in the same dial plan

This example enables Outlook Voice Access users associated with the UM dial plan named `MyUMDialPlan` to send voice messages to users associated with the same dial plan.

```
Set-UMDialPlan -identity MyUMDialPlan -SendVoiceMsgEnabled $true
```

This example prevents Outlook Voice Access users associated with the UM dial plan named `MyUMDialPlan` from sending voice messages to users associated with the same dial plan.

```
Set-UMDialPlan -identity MyUMDialPlan -SendVoiceMsgEnabled $false
```

Enable or prevent transferring calls from Outlook Voice Access

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can enable Outlook Voice Access users to transfer calls to a user who's associated with a Unified Messaging (UM) dial plan, or prevent them from doing so. By default, both this option and the **Leave voice messages without ringing a user's phone** option are enabled, so that Outlook Voice Access users can transfer calls to users in the same UM dial plan and leave voice messages for them. This setting only applies to Outlook Voice Access users who have entered their PIN and are authenticated.

For additional tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or prevent Outlook Voice Access users from transferring calls

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan that you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, click **Configure**.
3. In **transfer & search**, under **Allow callers to**, select the check box next to **transfer to users** to enable callers to transfer calls to other users within the dial plan. If you want to prevent Outlook Voice Access users from transferring calls to users, clear this check box.
4. Click **Save**.

Use the Shell to enable or prevent Outlook Voice Access users from transferring calls

This example enables Outlook Voice Access users to transfer calls to users in the same dial plan on a UM dial plan named `MyUMDialPlan`.

```
Set-UMDialPlan -identity MyUMDialPlan -
AllowDialPlanSubscribers $true
```

This example prevents Outlook Voice Access users from transferring calls to users in the same dial plan on a UM dial plan named `MyUMDialPlan`.

```
Set-UMDialPlan -identity MyUMDialPlan -
AllowDialPlanSubscribers $false
```

Configure the group of users that Outlook Voice Access users can contact

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-09*

You can specify which users can receive transferred calls or voice mail messages from Outlook Voice Access users. By default, the **In this dial plan only** option is selected. You can change this setting to allow Outlook Voice Access users to transfer calls or send voice messages to users located in the entire organization, to an existing UM auto attendant, or to a specific extension number.

For additional tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the group of users that Outlook Voice Access users can contact

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to modify, and then click **Edit** ✎.
3. On the **UM dial plan** page, click **Configure**.
4. In **Transfer & search**, under **Allow callers to search for users by name or alias**, select one of the following options:
   - **In this dial plan only** Use this option to allow Outlook Voice Access users who call in to an Outlook Voice Access number to locate and contact users who are within the same dial plan.
   - **In the entire organization** Use this option to allow Outlook Voice Access users who call in to an Outlook Voice Access number to locate and contact anyone in the entire organization. This includes all users who are mailbox-enabled.
   - **Only on this auto attendant** Use this option to allow Outlook Voice Access users who call in to an Outlook Voice Access number to connect to a specific auto attendant. You must create the auto attendant before you specify it here. This allows Outlook Voice Access users to be transferred to another auto attendant. The auto attendant you choose here can be a speech-enabled or non-speech-enabled auto attendant.
   - **Only for this extension** Use this option to allow Outlook Voice Access users to connect to an extension number that you specify. You can use only numeric digits for the extension. The number of digits that you define in this field must match the number of digits in the extension numbers that are configured on the UM dial plan.
5. Click **Save**.

Use the Shell to configure the group of users that Outlook Voice Access users can contact

This example sets the group of users that Outlook Voice Access users can contact for a UM dial plan named `MyUMDialPlan` to the entire organization.

```
Set-UMDialPlan -Identity MyUMDialPlan -ContactScope
'GlobalAddressList' -UMAutoAttendant $null -
AllowDialPlanSubscribers $false -AllowExtensions $false
```

This example sets the group of users that Outlook Voice Access users can contact for a UM dial plan

named `MyUMDialPlan` to the `DialPlan`.

```
Set-UMDialPlan -Identity MyUMDialPlan -ContactScope
DialPlan -AllowDialPlanSubscribers $false -AllowExtensions
$false
```

Configure the primary way for Outlook Voice Access users to search

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

When you create a Unified Messaging (UM) dial plan, you can configure the primary and secondary ways that callers can search for names to locate a user when they call an Outlook Voice Access number or a UM auto attendant that's associated with the dial plan. Callers can use touchtone inputs to locate a UM-enabled user.

Note:

None isn't an available option for the primary way callers can search for names. When **None** is selected for the secondary way they can search for names, only the primary way will be available to callers. If you configure both the primary and secondary ways that callers can search for names, they will be prompted for both ways.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the primary dial by name method

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to change, and then click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. In **Settings**, under **Primary way to search for names**, use the drop-down list to select the option you want:
   - **Last first** (default)
   - **First last**
   - **SMTP address**
5. Click **Save**.

Use the Shell to change the primary dial by name method

This example sets the primary dial by name method to `FirstLast`. This enables callers who call the Outlook Voice Access number or a UM auto attendant associated with the dial plan to search for a UM-enabled user by their first and then last name.

```
Set-UMDialPlan -Identity MyUMDialPlan -DialByNamePrimary
FirstLast
```

This example sets the primary dial by name method to `LastFirst`. This enables callers who call the Outlook Voice Access number or a UM auto attendant associated with the dial plan to search for a UM-enabled user by their last and then first name.

```
Set-UMDialPlan -Identity MyUMDialPlan -DialByNamePrimary
LastFirst
```

This example sets the primary dial by name method to `SMTP address`. This enables callers who call the Outlook Voice Access number or a UM auto attendant associated with the dial plan to search for a UM-enabled user by their SMTP address.

```
Set-UMDialPlan -Identity MyUMDialPlan -DialByNamePrimary
SMTPAddress
```

Configure the secondary way for

Outlook Voice Access users to search

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

When you create a dial plan, you can configure the primary and secondary *dial by name methods* or ways that callers can search for names. Callers use these dial by name methods to look up names to locate and contact a user when they call in to an Outlook Voice Access number or when they call in to a UM auto attendant that's associated with the dial plan. Callers can use touchtone inputs to locate a UM-enabled user.

Note:

If **None** is selected as the secondary way for callers to search for names, only the primary way of searching for names will be available to callers who want to locate users. If you configure both the primary and secondary ways that callers can search for names, callers will be prompted for both ways.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the secondary dial by name method

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.

2. In the list view, select the UM dial plan you want to change, and then click **Edit** ✎.
3. On the **UM Dial Plan** page, click **Configure**.
4. In **Settings**, under **Secondary way to search for names**, use the drop-down list to select the option you want:
   - **Last first** (default)
   - **First last**
   - **SMTP address**
   - **None**
5. Click **Save**.

Use the Shell to change the secondary dial by name method

This example sets the secondary dial by name method to `FirstLast`. This enables callers who call the Outlook Voice Access number or a UM auto attendant associated with the dial plan to search for a UM-enabled user by their first and then last name.

```
Set-UMDialPlan -Identity MyUMDialPlan -DialByNameSecondary FirstLast
```

This example sets the secondary dial by name method to `LastFirst`. This enables callers who call the Outlook Voice Access number or a UM auto attendant associated with the dial plan to search for a UM-enabled user by their last and then first name.

```
Set-UMDialPlan -Identity MyUMDialPlan -DialByNameSecondary LastFirst
```

This example sets the secondary dial by name method to `SMTP address`. This enables callers who call the Outlook Voice Access number or a UM auto attendant associated with the dial plan to search for a UM-enabled user by their SMTP address.

```
Set-UMDialPlan -Identity MyUMDialPlan -DialByNameSecondary SMTPAddress
```

This example sets the secondary dial by name method to `None` and the primary dial by name method to `SMTP address`. This enables callers who call the Outlook Voice Access number or a UM auto attendant associated with the dial plan to search for a UM-enabled user by their SMTP address only.

```
Set-UMDialPlan -Identity MyUMDialPlan -DialByNamePrimary SMTPAddress -DialByNameSecondary None
```

Configure the number of sign-in failures before Outlook Voice Access users are disconnected

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-09*

You can specify the number of sequential unsuccessful sign-in attempts that are allowed before a caller is disconnected. The value of this setting can be from 1 through 20. Setting this value too low can frustrate users. For most organizations, this value should be set to the default of three attempts.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the number of sign-in failures before users are disconnected

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
3. On the **UM dial plan** page, click **Configure**.

4. In **Settings**, under **Number of sign-in failures before disconnecting**, enter the number of sign-in failures.
5. Click **Save**.

Use the Shell to configure the number of sign-in failures before users are disconnected

This example sets the number of sign-in failures before users are disconnected to 5 for a UM dial plan named `MyUMDialPlan`.

```
Set-UMDialPlan -identity MyUMDialPlan -
LogonFailuresBeforeDisconnect 5
```

Configure the number of input failures before Outlook Voice Access users are disconnected

Set up client voice mail features > Setting up Outlook Voice Access > Outlook Voice Access procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-09*

You can configure the number of times that users who call in to an Outlook Voice Access number can enter incorrect data before they're disconnected. This setting applies to both Outlook Voice Access users and unauthenticated callers who use directory search.

The following are examples of types of data that are considered incorrect:

- A caller requests an extension number that isn't found in the system.
- The system can't locate the user's extension number to transfer the call.
- A caller presses a menu option that isn't valid.

The value of this setting can be from 1 through 20. For most organizations, this value should be set to the default of three attempts. Setting this value too low may prematurely disconnect callers.

For additional management tasks related to UM dial plans, see UM dial plan procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the input failures before disconnect

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
3. On the **UM dial plan** page, click **Configure**.
4. In **Settings**, under **Number of input failures before disconnecting**, enter the number of input failures.
5. Click **Save**.

Use the Shell to configure the input failures before disconnect

This example sets the input failures before disconnect to 5 on a UM dial plan named `MyUMDialPlan`.

```
Set-UMDialPlan -identity MyUMDialPlan -
InputFailuresBeforeDisconnect 5
```

Configure the limit on personal greetings for Outlook Voice Access users

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-05*

The **Limit on personal greetings (minutes)** setting enables you to enter the maximum number of minutes that users associated with the Unified Messaging (UM) mailbox policy can use to record their voice mail greetings. This setting applies to both their standard voice mail and their Out of Office voice mail greetings. By default, the maximum greeting duration is set to 5 minutes. However, you can configure the maximum greeting duration to any setting between 1 and 10 minutes.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the maximum greeting duration

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then on the toolbar, click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then on the toolbar, click **Edit** .
3. On the **UM mailbox policy** page > **General**, under **Limit on personal greetings (minutes)**, enter the length of time, in minutes, allowed for personal greetings for voice mail users.
4. Click **Save**.

Use the Shell to change the maximum greeting duration

This example configures the maximum greeting duration on the UM mailbox policy `MyUMMailboxPolicy` to 3 minutes.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy
MaxGreetingDuration 3
```

Allow voice mail users to forward calls

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

The Call Answering Rules feature was first introduced in Exchange 2010. Using this feature, users who are enabled for voice mail can control how their incoming calls should be handled. Call answering rules are applied to incoming calls similar to the way Inbox rules are applied to incoming email messages.

Call answering rules are created and configured by a voice mail-enabled user using Outlook or Outlook Web App. The rules are stored along with other voice settings in the user's mailbox. A total of nine call answering rules can be set up for each UM-enabled mailbox. These rules are independent of the Inbox rules that are set up by users, and don't take up part of the Inbox rules storage quota for the user.

By default, when a user is enabled for Unified Messaging (UM) and voice mail, no call answering rules are configured. If an incoming call is answered by the voice mail system, the caller is prompted to leave a voice message or if the caller doesn't get prompted, the caller will also be able to leave a voice message for the user.

If your users want to have the voice mail system just answer their incoming calls and record a voice message, you don't have to create any call answering rules. However, if you decide that you want to set up conditions or actions, you can set them up by using the **Call Answering Rules** section on the **Voice Mail** page in Outlook Web App. Use the **Call Answering Rules** section to create, edit, and delete call answering rules.

Anatomy of call answering rules

A call answering rule consists of two parts: conditions and actions. You can associate one or more conditions with a single call answering rule. The call answering rule will only be processed if all the conditions for the rule are met. You can also associate one or more actions with a single call

answering rule. These actions determine what options will be offered to the caller when the call answering rule is processed.

Call Answering Rules supports the following conditions:

- Who the incoming call is from
- The time of day
- Calendar free/busy status
- Whether automatic replies are turned on for email

The following actions are supported:

- Find me
- Transfer the caller to someone else
- Leave a voice message

If a user records a custom greeting for a call answering rule, they must include the menu option as part of the custom greeting when they configure the call answering rule. If they don't, Unified Messaging won't generate a menu prompt that lets the caller know what his or her choices are. After the custom greeting is played, the server will wait for the caller's input. If a menu option isn't included in the greeting, the caller won't input anything and the server will prompt them, asking "Are you still there?"

Conditions

Conditions are rules that you can apply to call answering rules. By using a combination of conditions, you can create multiple call answering rules that will trigger when the conditions are met. To create a default rule that will be applied to every call, you create a rule that doesn't contain any conditions.

There are three conditions that can be used when you set up call answering rules, including:

- Caller ID
- Time-of-the-day
- Free/busy status

Actions

Actions are used to define what you want to happen when a condition is met. The two kinds of actions are:

- Find Me
- Call Transfer

Adding a Find Me action

When a caller selects Find Me, the voice mail system will attempt to locate you at up to two different phone numbers, and then connect the caller to you if you're available at one of the phone numbers.

- You can specify text that will be read to the caller. For example, if you enter “Urgent Matters” to

inform your callers that they should only select this action if they have important things to discuss with you, the voice mail system will say "For Urgent Matters, press the 1 key."

- You have to associate the Find Me action with the number on the telephone keypad that the caller will press to select this action. In the example above, the **1** telephone key is the number callers will press to reach you at one of the phone number or numbers you specify.
- Next, you have to specify the one or two phone numbers that the voice mail system will dial. If you specify two telephone numbers, the second number will be dialed if you're not available at the first. Each phone number that you specify has an associated duration. The duration is the time period during which the voice mail system will try to dial the phone number before it moves on to the next number. Or, if you can't be contacted, the voice mail system will go back to the options menu.
- After you've entered this information, click **Apply** to save the Find Me settings.

Adding Call Transfer actions

By setting a Call Transfer action, you provide callers with the option to be transferred to another person's phone number. There are several options that are available when you want to transfer an incoming call to another phone or contact.

- You can specify text that will be read to the caller. For example, you can enter “Important Matters” to inform your callers that they should choose this option if they have an important matter to discuss and need to speak to someone.
- You have to associate the **Call Transfer** action with the number on the telephone keypad that the caller will press to select this action.
- When you choose the Call Transfer action, you have to specify a person or phone number for the caller to be transferred to. You can choose a phone number or select a contact to be called when the caller presses the correct key on the telephone keypad. If you specify a contact who's within your company directory, the voice mail system will try to transfer the call to the extension number of that contact.
- In addition to specifying a person or number for the caller to be transferred to, you also need to specify the number on the telephone keypad that the caller will press to select the Call Transfer action.
- After you've entered this information, click **Apply** to save the Call Transfer settings.

Selecting a call answering rule for each incoming call

After you create and configure Call Answering Rules, Unified Messaging will:

1. Determine whether the user has created any call answering rules. If not, UM will offer the caller the option of leaving a voice message.
2. If one or more call answering rules have been configured, UM will evaluate each of these rules. The first rule whose conditions are met will be processed.
3. After evaluating all the rules, if UM doesn't find a rule whose conditions are met, UM will ask the caller to leave a voice message.

Dialing rules

Depending on how a call answering rule is configured, an incoming call may result in a call transfer. When this happens, the transfer target phone number will be subject to the dialing rules and restrictions on the UM mailbox policy that the called party is associated with. For more information about outdialing and dialing rules and restrictions, see Allow users to make calls.

Enabling/disabling Call Answering Rules

By default, Call Answering Rules is automatically enabled for UM-enabled users. However, you can disable call answering rules for users by disabling the feature on a UM mailbox policy or the user's mailbox. For details about how to enable or disable Call Answering Rules, see the following topics:

- Allow or prevent users in the same UM mailbox policy from creating call answering rules
- Allow or prevent a user from creating call answering rules

Forwarding calls procedures

Unified Messaging > Set up client voice mail features > Allow voice mail users to forward calls >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-29*

Allow or prevent a user from creating call answering rules

Allow or prevent users in the same UM mailbox policy from creating call answering rules

Create a call answering rule

View and manage a call answering rule

Enable or disable a call answering rule for a user

Remove a call answering rule for a user

Allow or prevent a user from creating call answering rules

Set up client voice mail features > Allow voice mail users to forward calls > Forwarding calls procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can specify whether you want individual users to be able to create and manage their own call answering rules by configuring their mailbox properties. By default, they can create call answering rules.

You can enable or disable Call Answering Rules for multiple users that are enabled for Unified Messaging (UM) by configuring Call Answering Rules on a UM dial plan or UM mailbox policy.

Note:

You can't use the EAC to configure this feature. You must use the Shell to enable or disable Call Answering Rules for a voice mail user.

For additional management tasks related to allowing users to forward calls, see Forwarding calls procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform this procedure, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to enable or disable call answering rules for a UM-enabled user

This example enables Call Answering Rules for the user tony@contoso.com.

```
Set-UMMailbox -Identity tony@contoso.com -
CallAnsweringRulesEnabled $true
```

This example disables Call Answering Rules for the user tony@contoso.com.

```
Set-UMMailbox -Identity tony@contoso.com -
CallAnsweringRulesEnabled $false
```

Allow or prevent users in the same UM mailbox policy from creating call answering rules

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can allow users who are associated with a Unified Messaging (UM) mailbox policy to configure call answering rules, or prevent them from doing so. If the option to configure call answering rules is disabled on a UM dial plan, the Call Answering Rules feature won't be available to UM-enabled users associated with the UM mailbox policy. The default setting is enabled.

For additional management tasks related to allowing users to forward calls, see Forwarding calls procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or disable call answering rules on a UM mailbox policy

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. Under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page, select or clear the check box next to **Allow users to configure call answering rules**.
4. Click **Save**.

Use the Shell to enable or disable call answering rules on a UM mailbox policy

This example allows users who are associated with the UM mailbox policy `MyUMMailboxPolicy` to create call answering rules.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowCallAnsweringRules $true
```

This example prevents users who are associated with the UM mailbox policy `MyUMMailboxPolicy` from creating call answering rules.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowCallAnsweringRules $false
```

Create a call answering rule

Set up client voice mail features > Allow voice mail users to forward calls > Forwarding calls procedures >

Topic Last Modified: *2013-05-07*

You can use the Shell to create one or more call answering rules for a user. You can also use the **New-UMCallAnsweringRule** cmdlet in an Exchange Management Shell script to create call answering rules for multiple users.

Call answering rules are applied to incoming calls similar to the way Inbox rules are applied to incoming email messages. By default, when a user is enabled for Unified Messaging (UM), no call answering rules are configured. Even so, incoming calls are answered by the mail system and callers are prompted to leave a voice message.

Note:
Users that are UM-enabled can sign in to Outlook Web App to create, manage, and remove call answering rules.

For additional management tasks related to Call Answering Rules, see Forwarding calls procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM call answering rules" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform this procedure, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- You can only use the Shell to perform this procedure.To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell.To learn how to use Windows PowerShell to connect to Exchange Online, see **Connect to Exchange Online using remote PowerShell**.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to create a call answering rule

This example creates the call answering rule `MyCallAnsweringRule` in the mailbox for Tony Smith with the priority of 2.

```
New-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith
```

This example creates the call answering rule `MyCallAnsweringRule` in the mailbox for Tony Smith and performs the following actions:

- Sets the call answering rule to two caller IDs.
- Sets the priority of the call answering rule to 2.

- Sets the call answering rule to allow callers to interrupt the greeting.

```
New-UMCallAnsweringRule -Name MyCallAnsweringRule -
CallerIds "1,4255550100,,","1,4255550123,," -Priority 2 -
CallersCanInterruptGreeting $true -Mailbox tonysmith
```

This example creates the call answering rule `MyCallAnsweringRule` in the mailbox for Tony Smith and performs the following actions:

Sets the priority of the call answering rule to 2.

Creates key mappings for the call answering rule.

If the caller reaches the voice mail for the user and the status of the user is set to Busy, the caller can:

- Press the 1 key and be transferred to a receptionist at extension 45678.
- Press the 2 key so the Find Me feature will be used for urgent issues, ring extension 23456 first, and then ring extension 45671.

```
New-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith -ScheduleStatus 0x4 - -KeyMappings
"1,1,Receptionist,,,,,45678,","5,2,Urgent
Issues,23456,23,45671,50,,"
```

View and manage a call answering rule

Set up client voice mail features > Allow voice mail users to forward calls > Forwarding calls procedures >

Topic Last Modified: *2013-05-07*

You can use the Shell to view or configure one or more call answering rules for a user. You can also use the **Get-UMCallAnsweringRule** or **Set-UMCallAnsweringRule** cmdlets in an Exchange Management Shell script to view or manage call answering rules for multiple users.

Call answering rules are applied to incoming calls similar to the way Inbox rules are applied to incoming email messages. By default, when a user is enabled for Unified Messaging (UM), no call answering rules are configured. Even so, incoming calls are answered by the mail system and callers are prompted to leave a voice message.

Important:
Users that are UM-enabled can sign in to Outlook Web App to create, manage, and remove call

answering rules.

For additional management tasks related to Call Answering Rules, see Forwarding calls procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM call answering rules" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform this procedure, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- You can only use the Shell to perform this procedure.To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell.To learn how to use Windows PowerShell to connect to Exchange Online, see **Connect to Exchange Online using remote PowerShell**.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to view a call answering rule

You can retrieve the properties for a single call answering rule or a list of call answering rules in a UM-enabled user's mailbox.

This example returns a formatted list of call answering rules in a user's UM-enabled mailbox.

```
Get-UMCallAnsweringRule-Mailbox tonysmith | Format-List
```

This example displays the properties of the call answering rule `MyUMCallAnsweringRule`.

```
Get-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
```

Use the Shell to configure a call answering rule

You can configure or change a call answering rule that's stored in a user's mailbox. You can specify

the following conditions:

- Who the incoming call is from
- Time of day
- Calendar free/busy status
- Whether automatic replies are turned on for email

You can also specify the following actions:

- Find me
- Transfer the caller to someone else
- Leave a voice message

This example sets the priority to 2 on the call answering rule `MyCallAnsweringRule` that exists in the mailbox for Tony Smith.

```
Set-UMCallAnsweringRule -Mailbox tonysmith -Name
MyCallAnsweringRule -Priority 2
```

This example performs the following actions on the call answering rule `MyCallAnsweringRule` in the mailbox for Tony Smith:

- Sets the call answering rule to two caller IDs.
- Sets the priority of the call answering rule to 2.
- Sets the call answering rule to allow callers to interrupt the greeting.

```
Set-UMCallAnsweringRule -Name MyCallAnsweringRule -
CallerIds "1,4255550100,,","1,4255550123,," -Priority 2 -
CallersCanInterruptGreeting $true -Mailbox tonysmith
```

This example changes the free/busy status to Away on the call answering rule `MyCallAnsweringRule` in the mailbox for Tony Smith and sets the priority to 2.

```
Set-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith@contoso.com -ScheduleStatus 0x8
```

Enable or disable a call answering rule for a user

Set up client voice mail features > Allow voice mail users to forward calls > Forwarding calls procedures >

Topic Last Modified: *2013-05-07*

You can use the Shell to enable or disable one or more call answering rules for a user. You can also

use the **Enable-UMCallAnsweringRule** or **Disable-UMCallAnsweringRule** cmdlets in an Exchange Management Shell script to enable or disable one or more call answering rules for multiple users.

Call answering rules are applied to incoming calls similar to the way Inbox rules are applied to incoming email messages. By default, when a user is enabled for Unified Messaging (UM), no call answering rules are configured. Even so, incoming calls are answered by the mail system and callers are prompted to leave a voice message.

For additional management tasks related to call answering rules, see Forwarding calls procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM call answering rules" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform this procedure, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- You can only use the Shell to perform this procedure.To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell.To learn how to use Windows PowerShell to connect to Exchange Online, see **Connect to Exchange Online using remote PowerShell**.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to enable a call answering rule

When a call answering rule is created, it's enabled. You can use the Shell to enable a call answering rule that was previously disabled. Enabling a call answering rule enables the **Enable-UMCallAnsweringRule** cmdlet to retrieve the call answering rule, including the conditions and actions for a specified call answering rule.

This example enables the call answering rule `MyUMCallAnsweringRule` in the mailbox for Tony Smith.

```
Enable-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
```

```
-Mailbox tonysmith
```

The example uses the *WhatIf* switch to test whether the call answering rule `MyUMCallAnsweringRule` in the mailbox for Tony Smith is ready to be enabled and if there are any errors within the command.

```
Enable-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
-Mailbox tonysmith -WhatIf
```

This example enables the call answering rule `MyUMCallAnsweringRule` in the mailbox for Tony Smith and prompts the signed-in user to confirm that the call answering rule is to be enabled.

```
Enable-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
-Mailbox tonysmith -Confirm
```

Use the Shell to disable a call answering rule

Disabling a call answering rule prevents it from being retrieved and processed when an incoming call is received. When you create a call answering rule, you should disable it while you're setting up conditions and actions. This prevents the call answering rule from being processed when an incoming call is received before you've correctly configured the call answering rule.

This example disables the call answering rule `MyUMCallAnsweringRule` in the mailbox for Tony Smith.

```
Disable -UMCallAnsweringRule -Identity
MyUMCallAnsweringRule -Mailbox tonysmith
```

This example uses the *WhatIf* switch to test whether the call answering rule `MyUMCallAnsweringRule` in the mailbox for Tony Smith is ready to be disabled and if there are any errors within the command.

```
Disable -UMCallAnsweringRule -Identity
MyUMCallAnsweringRule -Mailbox tonysmith -WhatIf
```

This example disables the call answering rule `MyUMCallAnsweringRule` in the mailbox for Tony Smith and prompts the signed-in user to confirm that they're disabling the call answering rule.

```
Disable-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
-Mailbox tonysmith -Confirm
```

Remove a call answering rule for a user

Set up client voice mail features > Allow voice mail users to forward calls > Forwarding calls procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-05-07*

You can use the Shell to remove one or more call answering rules for a user. You can also use the **Remove-UMCallAnsweringRule** cmdlet in an Exchange Management Shell script to remove one or more call answering rules for multiple users.

Call answering rules are applied to incoming calls similar to the way Inbox rules are applied to incoming email messages. By default, when a user is enabled for Unified Messaging (UM), no call answering rules are configured. Even so, incoming calls are answered by the mail system and callers are prompted to leave a voice message.

Note:

Users that are UM-enabled can sign in to Outlook Web App to create, manage, and remove call answering rules.

For additional management tasks related to Call Answering Rules, see Forwarding calls procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM call answering rules" entry in the Unified Messaging permissions topic.
- Before you perform this procedure, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform this procedure, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- You can only use the Shell to perform this procedure.To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell.To learn how to use Windows PowerShell to connect to Exchange Online, see **Connect to Exchange Online using remote PowerShell**.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to remove a call answering rule

This example removes the call answering rule `MyUMCallAnsweringRule` from a user's mailbox. The user's mailbox is the mailbox of the user running the cmdlet.

```
Remove-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
```

This example removes the call answering rule `MyUMCallAnsweringRule` from the mailbox of Tony Smith.

```
Remove-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
-Mailbox tonysmith
```

Allow users to make calls

Exchange Server 2013 > Unified Messaging > Set up client voice mail features >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-29*

Outdialing is the process by which users call in to a UM dial plan using an Outlook Voice Access number and place or transfer a call to an internal or external telephone number. Unified Messaging uses many outdialing settings to dial calls for users. To configure outdialing, you must configure dialing rules, dialing rule groups, and dialing authorizations on Unified Messaging (UM) dial plans and then authorize outdialing on UM dial plans, UM mailbox policies, and auto attendants. You can also configure UM dial plans to have dialing or access codes, a national number prefix, and in-country/region or international number formats that enable you to control outdialing in your organization. This topic discusses dialing rules, dialing rule groups, and dialing authorizations and how they are used to authorize and control outdialing for your organization.

Contents

Overview

Outdialing happens when:

- A call is placed to an external telephone number.
- A call is transferred to an auto attendant.
- A call is transferred to a user in your organization.

- A UM-enabled user uses the Play on Phone feature.

For outdialing to work correctly, the following settings must be configured correctly:

- **Dialing rules** Dialing rules define the number that is dialed by the UM-enabled user and the number that will be dialed by the Private Branch eXchange (PBX) or IP PBX.
- **Dialing rule groups** Dialing rule groups determine the types of calls that users within a dialing group can make.
- **Dialing authorizations** Dialing authorizations determine the restrictions that will be applied to prevent users from incurring unnecessary telephone charges or from dialing long-distance calls.

To enable outdialing for users who call in to a dial plan or an auto attendant, you must:

- Make sure the VoIP gateways represented by a UM IP gateway that is linked with a dial plan will allow outgoing calls.
- Create dialing rule groups by creating dialing rules on the UM dial plan.
- Add dialing authorizations for in-country/region and international dialing rule groups on the UM dial plan, UM mailbox policy, or auto attendant associated with the same dial plan as the UM IP gateway.

Types of users

Two types of users can use the outdialing feature in Unified Messaging: authenticated and unauthenticated. All users who call in to a UM auto attendant are unauthenticated. When users call in to an Outlook Voice Access number, they're considered unauthenticated because they haven't provided their extension number and PIN and signed in to their mailbox. Users are authenticated after they provide their extension number and PIN and successfully sign in to their mailbox.

When users call in to an Outlook Voice Access number configured on a UM dial plan and try to place or transfer a call without signing in to their mailbox, only the UM dial plan outdialing settings are applied to the call. When anonymous or unauthenticated users call in to a UM auto attendant, both the outdialing settings configured on the auto attendant and the outdialing settings configured on the dial plan associated with the auto attendant are applied to the call.

When users call in to the Outlook Voice Access number configured on a dial plan and successfully sign in to their mailbox, they become authenticated users. When they're authenticated, the outdialing call settings use the dialing rules and dialing authorization settings on the UM mailbox policy that's linked to those users.

Return to top

Outdialing settings

You need to configure several settings to apply outdialing rules for your organization. In addition to configuring the UM dial plans, UM auto attendants, and UM mailbox policies that you've created with the correct dialing rules and dialing authorizations, you need to configure access codes, number prefixes, and number formats on the UM dial plans. The following outdialing settings are

configured on dial plans, auto attendants, and UM mailbox policies:

- Outside line, country/region, and international access codes
- National number prefixes
- In-country/region and international number formats
- Configured in-country/region and international dialing rule groups
- Allowed in-country/region and international dialing rule groups
- Dialing rule entries
- Dialing authorizations

For you to successfully configure outdialing for your organization, you first need to understand how each component can be used with outdialing and how the component must be configured. The following table introduces each component that needs to be configured on UM dial plans, UM auto attendants, and UM mailbox policies before outdialing will work correctly.

Outdialing components

Component	Description
Dial codes, number prefixes, and number formats	UM uses dial codes, number prefixes, and number formats to determine the correct number to dial when placing an outgoing call. You can configure dial codes, number prefixes, and number formats to restrict outgoing calls for users who dial in to a UM auto attendant associated with a UM dial plan or for users who dial in to an Outlook Voice Access number configured on the dial plan.
Dialing rule groups	Dialing rule groups are created to enable telephone numbers to be modified before they're sent to the PBX for outgoing calls. Dialing rule groups remove numbers from or add numbers to telephone numbers being called by UM. For example, you can create a dialing rule group that automatically adds a 9 as a prefix to a 7-digit telephone number to provide access to an outside line. In this example, users who place outgoing calls don't have to dial the 9 before the telephone number to reach someone external to the organization.

	Each dialing rule group contains dialing rules that determine the types of in-country/region and international calls that users within a dialing rule group can make. Dialing rule groups apply to the users who are associated with a UM dial plan or to UM auto attendants and UM mailbox policies associated with the UM dial plan. Each dialing rule group must contain at least one dialing rule.
Dialing rule entries	A dialing rule is used to determine the types of calls that users within a dialing rule group can make. When you create a dialing rule group, you configure one or more dialing rules. When you configure each dialing rule, you must enter the dialing rule name, number pattern to transform (*number mask*), and dialed number. You can also enter a comment. Comments can be used to describe how the dialing rule will be used or to describe a group of users to whom the dialing rule will apply. When you add a number mask and the dialed number to a dialing rule, you can substitute the letter x for a digit in a telephone number, for example, 91425xxxxxxx. You can also use an asterisk (*) symbol as a wildcard character, for example, 91425*.
Dialing authorizations	A dialing authorization uses dialing rule groups to apply dialing restrictions for users who are associated with a specific UM mailbox policy, dial plan, or auto attendant. They can also be used when you want to let users place calls to in-country/region or international telephone

	numbers. After you create dialing rules on a UM dial plan, you add the dialing rule group to a UM mailbox policy, dial plan, or auto attendant. After the dialing rule group is added to a UM mailbox policy, all settings or rules defined will apply to UM-enabled users who are linked with the UM mailbox policy.

Return to top

Configuring outdialing

A dialing rule group is a collection of one or more dialing rules configured on a UM dial plan. Two types of dialing rule groups can be configured on a UM dial plan: in-country/region and international. In-country/region dialing rule groups apply to telephone numbers dialed within the same country or region. International dialing rule groups apply to international telephone numbers dialed from one country or region to another country or region.

Each UM dial plan can contain one or more dialing rule groups. To apply a dialing rule group to a set of users, after you create the dialing rule group, you must add it to the list of allowed dialing rule groups on the UM dial plan and on the UM auto attendants and UM mailbox policies associated with the UM dial plan.

Dialing rule groups enable you to specify dialing rules that you want to apply to a group of UM-enabled users who fall into a specific category. For example, you can use dialing rule groups to specify which group of users can place international calls and which group can make only in-state or local calls. You can create a dialing rule group using the Exchange Administration Center (EAC) or the **Set-UMDialPlan** cmdlet in the Exchange Management Shell. When you create a dialing rule group, you must define at least one dialing rule for the group.

When a user dials a telephone number, UM takes the number and looks for a match in the dialing rules. If a match is found, UM uses the dialing rule to determine the number to dial by looking at the telephone number or digits listed in the **Dialed Number** section of the dialing rule. The number listed in the **Dialed Number** box of the dialing rule will be dialed.

The following table shows an example of dialing rule groups and dialing rules. In this example, Local-Calls-Only and Low-Rate are the dialing rule groups that have been created. The dialing rule group Local-Calls-Only has two dialing rules: 91425* and 91206*, and the dialing rule group Low-Rate also has two dialing rules: 91509* and 91360*.

Dialing rule groups and dialing rules

Name	NumberMask	DialedNumber	Comment
Local-Calls-Only	91425*	91*	Local calls
Local-Calls-Only	91206*	91*	Local calls
Low-Rate	91509*	9*	In-state calls
Low-Rate	91360*	9*	In-state calls

For example, when a user dials 9-1-425-555-1234, UM dials 4255551234. UM removes any nonnumeric characters (in this example, the hyphens) and applies the number mask from the dialing rule. In this example, UM applies the number mask 91*. This tells UM not to dial the 9 or the 1, but to dial all the other numbers in the telephone number that appear to the right of the number 1. This includes all the numbers represented by the asterisk (*).

You can use the EAC or the Shell to create and configure single or multiple in-country/region and international dialing rule groups and dialing rules. However, if you're creating many or complex dialing rule groups and dialing rules, you can use a comma-separated value (.csv) file in the Shell. You can import or export a list of dialing rule groups and dialing rules.

To import a list of dialing rule groups and dialing rules that you've defined in a .csv file, run the **Set-UMDialPlan** cmdlet, as follows.

```
Set-UMDialPlan "MyUMDialPlan" -
ConfiguredInCountryOrRegionGroups $(IMPORT-CSV c:\dialrules
\InCountryRegion.csv)
```

To retrieve a list of the dialing rule groups configured on a UM dial plan, run the **Get-UMDialPlan** cmdlet, as follows.

```
(Get-UMDialPlan -id
"MyUMDialPlan").ConfiguredInCountryOrRegionGroups | EXPORT-
CSV C:\incountryorregion.csv
```

The .csv file must be created and saved in the correct format. Each line in the .csv file represents one dialing rule. However, each dialing rule is configured on the same dialing rule group. Each rule in the file will have four sections separated by commas. These sections are name, number mask, dialed number, and comment. Each section is required, and you must enter the correct information in each section except for the comment section. There should be no spaces between the text entry and the comma for the next section, nor should there be any blank lines between the rules or at the end. The following is an example of a .csv file that can be used to create in-country/region dialing rule groups and dialing rules.

Name,NumberMask,DialedNumber,Comment

Low-rate,91425xxxxxxx,9xxxxxxx,Local call

Low-rate,9425xxxxxxx,9xxxxxxx,Local call

Low-rate,9xxxxxxx,9xxxxxxx,Local call

Any,91*,91*,Open access to in-country/region numbers

Long-distance,91408*,91408*,long distance

The following is an example of a .csv file that can be used to create international dialing rule groups and dialing rule entries.

Name,NumberMask,DialedNumber,Comment

International, 901144*, 901144*, international call

International, 901133*, 901133*, international call

Return to top

Applying configured dialing rule groups

Dialing rule groups are created on a UM dial plan. You can create in-country/region or international dialing rule groups using the EAC or the **Set-UMDialPlan** cmdlet in the Shell. After you create the appropriate dialing rule groups on a UM dial plan and define the dialing rules, you can apply the dialing rule groups that you created to a UM dial plan, a UM auto attendant, or to users who are associated with a UM mailbox policy, and authorize outdialing depending on how the user accesses the voice mail system.

You can apply the dialing rule groups that you created on a UM dial plan to the following:

- **Same dial plan** The settings will apply to all users who call in to an Outlook Voice Access number but don't sign in to their mailbox. To apply an in-country/region dialing rule group named `MyAllowedDialRuleGroup` to the same dial plan, use the Shell **Set-UMDialPlan** cmdlet, as follows.

```
Set-UMDialPlan -Identity MyUMDialPlan -
AllowedInCountryOrRegionGroups MyAllowedDialRuleGroup
```

- **Single or multiple UM mailbox policies** The settings that are configured on a UM mailbox policy will apply to all users who are linked with that UM mailbox policy. The settings configured on a UM mailbox policy apply to users who call in to an Outlook Voice Access number and sign in to their mailbox. To apply an in-country/region dialing rule group named `MyAllowedDialRuleGroup` to a single UM mailbox policy, use the **Dialing authorization** page on the UM mailbox policy in the EAC or use the **Set-UMMailboxPolicy** cmdlet in the Shell, as follows.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
AllowedInCountryOrRegionGroups MyAllowedDialRuleGroup
```

- **Single or multiple auto attendants associated with the UM dial plan** This will apply to all users who call in to a UM auto attendant. To apply the in-country/region dialing rule group

named `MyAllowedDialRuleGroup` to a single UM auto attendant, use the **Dialing authorization** page on the auto attendant in the EAC or the **Set-UMAutoAttendant** cmdlet in the Shell, as follows.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
AllowedInCountryOrRegionGroups MyAllowedDialRuleGroup
```

The following table summarizes the way that dialing rule groups are applied in Unified Messaging.

Applying outdialing rules

Caller type	Scope	Outdialing settings applied
Outlook Voice Access number	User calls a dial plan Outlook Voice Access number and signs in to the mailbox	UM mailbox policy
Anonymous caller	User calls a dial plan Outlook Voice Access number	UM dial plan
Anonymous caller	User calls an auto attendant pilot or extension number	UM auto attendant
Caller from inside the organization	User calls the Play on Phone number	UM mailbox policy

Return to top

Applying dialing rules

The outdialing process happens when:

- Unified Messaging places a call to an external telephone number for a caller.
- Unified Messaging transfers a call to an auto attendant.
- Unified Messaging transfers a call to a user in your organization.
- A UM-enabled user uses the Play on Phone feature.

In each outdialing scenario, UM will apply the dialing rules that have been configured, and then place the call for the user. However, depending on the scenario and how the call is initiated by the user, UM may apply only some of the dialing rules to the telephone number being dialed. In other outdialing scenarios, UM may apply all the outdialing rules configured to the telephone number being dialed.

Return to top

Dial codes, number prefixes, and number formats

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-11-18*

You can configure several dialing codes that Unified Messaging (UM) uses to dial internal and external calls for UM-enabled users. Frequently, you want to configure a dial plan together with the dialing or access codes, a national number prefix, or in-country/region or international number formats so that you can control outdialing for users in your organization. This topic discusses dial codes, number prefixes, and number formats and how you can use them to control outdialing for your organization.

Contents

Overview

Outdialing is the process in which users call in to a UM dial plan or UM auto attendant and then place a call to an internal or external telephone number. When a user calls in to a UM dial plan or a UM auto attendant and then places a call, Unified Messaging uses the settings configured on the dial plan, auto attendant, and UM mailbox policies to place the call. UM places an outgoing call in the following situations:

- When it places a call to an external telephone number for a caller
- When it transfers a call to an auto attendant
- When it transfers a call to a user (either UM-enabled or not) in your organization
- When a UM-enabled user uses the Play on Phone feature

Two types of users use outdialing: authenticated users and unauthenticated users. Unauthenticated users call in to an Outlook Voice Access number configured on a UM dial plan but don't sign in to their mailbox. Unauthenticated users also call in to a number configured on a UM auto attendant. Authenticated users call in to an Outlook Voice Access number and successfully sign in to their

mailbox. When users call in to an Outlook Voice Access number, they are initially considered unauthenticated because they haven't provided their extension number and PIN and signed in to their mailbox. They are authenticated after they provide their extension number and PIN and successfully sign in to their mailbox.

When an unauthenticated user calls in to a UM auto attendant and places a call using outdialing, the outdialing settings configured on the UM dial plan and the auto attendant are used. When an unauthenticated user calls in to an Outlook Voice Access number configured on a dial plan, only the settings configured on the dial plan are used. When a user has successfully signed in to their mailbox, configuration settings from the dial plan and the UM mailbox policy associated with the authenticated user are applied to the authenticated user.

You need to configure several settings to control outdialing for your organization. To control outdialing, you need to configure the UM dial plans, auto attendants, and UM mailbox policies in Unified Messaging. The following settings can be configured on UM dial plans, auto attendants, and UM mailbox policies to control outdialing:

- Outside line, in-country/region, and international access codes
- National number prefixes
- In-country/region and international number formats
- In-country/region and international dialing rule groups
- Allowed in-country/region and international dialing rule groups
- Dialing rule entries

You configure access codes, number prefixes, and number formats on a UM dial plan on the **Dial Codes** page in the Exchange admin center (EAC). You can also configure the settings using the **Set-UMDialPlan** cmdlet in the Exchange Management Shell. You can choose to configure all the settings, none of the settings, or only some of the settings. Each setting controls a specific part of the outdialing process.

UM uses access codes, number prefixes, and number formats to determine the correct number to dial. They can be configured to restrict outgoing calls for users who dial in to a UM auto attendant associated with a UM dial plan or who dial in to the Outlook Voice Access number configured on the dial plan.

For more information about outdialing in Unified Messaging, see Dial codes, number prefixes, and number formats.

Return to top

Outside line access code

You can configure an outside line access code, also known as a *trunk access code*, on each dial plan that you create. This is the number used to gain access to an outside telephone line. This number is also configured on the Private Branch eXchanges (PBXs) or IP PBXs in your organization. In most telephony networks, users dial the number 9 to gain access to an outside line and place a call to an external telephone number.

You should configure an outside line access code on each dial plan that you create. This dialing code will apply to all users who are linked with a UM mailbox policy that's linked with the UM dial plan. When a caller who's linked with the dial plan places a call and the dial plan dials the outgoing call, UM adds the outside line access code (usually 9) in front of the dialed number string so that the PBX or IP PBX can dial the number correctly. If you don't configure the outside line access code, the PBX or IP PBX may not recognize the number that's sent.. For example, as stated earlier, in many organizations, the access code that users dial to gain access to an outside line is 9, and this is configured on a PBX or IP PBX. Unified Messaging must add the outside line access code (9) before the telephone number string for the PBX or IP PBX to correctly dial the outgoing number. If you configure the dialing code so that Unified Messaging will add the outside line access code, Unified Messaging will be able to use the outside line access code to access an outside line before it dials the external telephone number string. The dialing code that you configure will apply to all users who are linked with a UM mailbox policy linked with the UM dial plan.

National number prefix

The national number prefix and the country/region code can also be configured on a UM dial plan. Unified Messaging uses the number you enter to dial the correct national number prefix or country/region code when a user dials an outgoing call destined within the same country/region or an international call. For example, when a user from North America places an outgoing international call to Europe, UM will add the national number prefix before the number string that it sends to the PBX or IP PBX to place the outgoing call. The number 1 is used as the national number prefix for North America.

In-Country/region access code

A country/region code can be configured on a UM dial plan. The country/region access code consists of the digits associated with a specific country or region. Unified Messaging uses the country/region access code to dial the correct telephone number when a call is placed to a telephone number from inside the same country or region. UM will add this number before the number string that it sends to the PBX or IP PBX when it places the outgoing call. For example, UM will add the number 1 to a call placed from the United States and destined for the United States. For the United Kingdom, the country/region code is 44.

International access code

An international access code can be configured on a UM dial plan. The international access code consists of the digits used to access international telephone numbers. Unified Messaging uses the international access code to dial the correct international access code when a call is placed from a telephone number within a country/region and the number being dialed is located in another country/region. UM will add this number before the number string that it sends to the PBX or IP PBX

when it places the outgoing call. For example, UM will use 011 as the international access code for the United States. For Europe, the international access code is 00.

Return to top

In-Country/region and international number formats

You can configure the incoming call configuration for country/region and international number formats on a UM dial plan. After you configure these settings, Unified Messaging will be able to recognize incoming calls from inside a country/region and internationally between UM dial plans within the same organization. You can also add number formats for incoming calls that are placed within a single dial plan. Configuring these options enables your organization to save money by preventing outgoing calls that shouldn't be made by users from inside your organization, and helps to prevent toll fraud. UM will use the information that you configure to examine the number format of the incoming call and verify that the number pattern matches before it accepts the call. For example, you may have multiple dial plans inside an organization. If you have one dial plan for the United States and another for the United Kingdom, you may want to let users in the United States dial plan have UM place calls to users who are located in the United Kingdom dial plan, but not let the users in the United States dial plan place calls directly to other countries/regions or internationally.

Allowing users to make calls procedures

Applies to: *Exchange Online*

Topic Last Modified: *2013-05-03*

Enable outgoing calls on UM IP gateways

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-13*

You can enable outgoing calls for a Unified Messaging (UM) IP gateway if outgoing calls have been disabled. When you select the **Allow outgoing calls through this UM IP gateway** option on the properties for the UM IP gateway, you configure the UM IP gateway to accept and send outgoing calls to a Voice over IP (VoIP) gateway, Private Branch eXchange (PBX) enabled for Session Initiation Protocol (SIP), IP PBX, or session border controller (SBC). Although the **Allow outgoing calls through this UM IP gateway** setting controls whether the UM IP gateway is able to initiate outgoing calls for users, it doesn't affect call transfers or incoming calls from a VoIP gateway, PBX enabled for SIP, IP PBX, or SBC.

Outdialing is the term used to describe a situation in which a user in one UM dial plan initiates a call to a UM-enabled user in another dial plan or to an external telephone number.

To allow outdialing for UM-enabled users, you must:

- Verify that the UM IP gateway allows outgoing calls.
- Create dialing rule groups by creating dialing rule entries on the UM dial plan associated with the UM IP gateway.
- Add the correct dialing rule groups to the list of dialing restrictions in **Dialing authorization** on the UM dial plan, auto attendant, or UM mailbox policy.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable outgoing calls for a UM IP gateway

1. In the EAC, navigate to **Unified Messaging** > **UM IP Gateways**, select the UM IP gateway you want to change, and then click **Edit** .
2. On the **UM IP Gateway** page, select the check box next to **Allow outgoing calls through this UM IP gateway**.
3. Click **Save**.

Use the Shell to enable outgoing calls for a UM IP gateway

This example enables outgoing calls on a UM IP gateway named `MyUMIPGateway`.

```
Set-UMIPGateway -Identity MyUMIPGateway -OutcallsAllowed
$true
```

Disable outgoing calls on UM IP gateways

Set up client voice mail features > Allow users to make calls > Allowing users to make calls procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-13*

You can enable or disable outgoing calls for a Unified Messaging (UM) IP gateway. When you clear the **Allow outgoing calls through this UM IP gateway** option on the properties for the UM IP gateway, you configure the UM IP gateway to not accept and send outgoing calls to a Voice over IP (VoIP) gateway, IP PBX, or session border controller (SBC). Although the **Allow outgoing calls through this UM IP gateway** setting controls whether the UM IP gateway is able to initiate outgoing calls for users, it doesn't affect call transfers or incoming calls from a VoIP gateway, IP PBX, or SBC.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to disable outgoing calls for a UM IP gateway

1. In the EAC, navigate to **Unified Messaging** > **UM IP Gateways**, select the UM IP gateway you want to change, and then click **Edit** .
2. On the **UM IP Gateway** page, clear the check box next to **Allow outgoing calls through this UM IP gateway**.
3. Click **Save**.

Use the Shell to disable outgoing calls for a UM IP gateway

This example disables outgoing calls on a UM IP gateway named `MyUMIPGateway`.

```
Set-UMIPGateway -Identity MyUMIPGateway -OutcallsAllowed
$false
```

Configure dial codes

Set up client voice mail features > Allow users to make calls > Allowing users to make calls procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can configure dial codes, number prefixes, and number formats that are used by Unified Messaging to dial incoming and outgoing calls for users who are enabled for UM. In most cases, you'll configure a dial plan with the dial codes, prefixes, and number formats currently configured on your telephony network.

Dial codes and number prefixes are used to determine the correct number to dial for an outgoing call that's placed by a UM-enabled user. *Outdialing* is the term used to describe the process by which a user in a UM dial plan initiates an outgoing call. Number formats are used for incoming calls within a country or region, international calls, or calls that are placed within a dial plan. You can configure a dial plan to match the incoming call number format for both in-country/region and international numbers. When you configure the in-country/region and international number formats, you can restrict incoming calls for users linked with a dial plan.

For additional management tasks related to outdialing, see Allowing users to make calls procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure dial codes, prefixes, and number formats

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. Select the UM dial plan you want to manage, and then click **Edit** .
3. On the **UM Dial Plan** page, click **Configure**.
4. On the **UM dial plan** page > **Dial codes**, configure the following options:
   - **Outside line access code**
   - **International access code**

- **National number prefix**
- **Country/Region code**

5. Under **Number formats for dialing between dial plans**, configure the following:
   - **Country/Region number format**
   - **International number format**
   - **Number formats for incoming calls within the same dial plan** To add a number format, click **Add +**.
6. Click **Save** to save your changes.

Use the Shell to configure dial codes, prefixes, and number formats

This example configures a UM dial plan named `MyUMDialPlan` with an in-country or region number format, an international number format, and the following dial codes:

- 9 for the outside line access code
- 011 for the international access code
- 1 for the national number prefix
- 1 for the country or region code

```
Set-UMDialPlan -Identity MyUMDialPlan -
OutsideLineAccessCode 9 -InternationalAccessCode 011 -
NationalNumberPrefix 1 CountryorRegionCode 1 -
InCountryOrRegionNumberFormat 1425xxxxxxx -
InternationalNumberFormat 441425xxxxxxx
```

Create dialing rules for users

Set up client voice mail features > Allow users to make calls > Allowing users to make calls procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

Dialing rule groups consist of dialing rule entries. Dialing rules are used to modify a phone number before sending it to an on-premises telephone system (PBX) or IP PBX for outgoing calls. Dialing rules serve two purposes:

- They specify the numbers that can be dialed for outgoing calls. When you create a dialing rule, you specify the number formats that can be dialed. Any number that doesn't match one of the formats you specified is rejected. If you don't set any dialing rules, callers can place calls within your organization but can't make any outgoing calls.

- They transform the numbers dialed before sending them out to your on-premises telephone system. Dialing rules can strip numbers from or add numbers to the number dialed. For example, you can use dialing rules to add the outside line access code for your telephone system or to add or remove the in-country/region code for long-distance or local numbers.

To specify the types of outgoing calls you want to allow for a UM dial plan, you create a dialing rule group with dialing rules and then use them to authorize outgoing calls for Outlook Voice Access users and callers that dial into a UM auto attendant. You create separate dialing rule groups for in-country/region and for international calls.

Note:

If you are integrating UM with Microsoft Lync Server, we recommend that you create at least one dialing rule group and authorize that dialing rule group on the SIP URI dial plans, UM mailbox policies, and UM auto attendants to allow all outgoing calls to be forwarded to Lync Servers.

For other management tasks for outdialing, see Allowing users to make calls procedures.

Examples of commonly used dialing rules

Number pattern	Dialed number	When would you use this dialing rule?
*	*	Allow all outgoing calls.
1425xxxxxxx	91425xxxxxxx	Prevent users from getting an internal extension or an error when they forget to dial the outside access line number.
1xxxxxxxxxx	1xxxxxxxxxx	Allow all numbers that start with 1.
xxxxxxx	1425xxxxxxx	Add 1 and the local area code 425 to 7-digit numbers.

What do you need to know before you begin?

- Estimated time to complete: Less than 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed

steps, see Create a UM dial plan.

- If you will be applying dialing rule groups to UM mailbox policies, you will need to confirm that a UM mailbox policy is created. For detailed steps, see Create a UM mailbox policy.
- If you will be applying dialing rule groups to UM auto attendants, you will need to confirm that a UM auto attendant is created. For detailed steps, see Create a UM auto attendant.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the EAC to create a dialing rule

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, click **Configure**.
3. On the **UM Dial Plan** page > **Dialing rules**, click **Add +** under **In-country/region dialing rules** or **International dialing rules**.
4. On the **New Dialing Rule** page, enter the following information:
   - **Dialing rule name** Enter the name of the dialing rule group you want this rule to be a part of. To combine it with other rules, use the same group name. To create a new dialing rule group, enter a new unique name.
   - **Number pattern to transform (number mask)** Enter the number pattern to transform before dialing, for example, 91425xxxxxxx. If a caller dials a number that matches, UM transforms it to the dialed number before placing the call. Enter only numbers and the wildcard (x). The number pattern is also called a *number mask*.
   - **Dialed number** Enter the number to dial. Use only numbers and the wildcard (x), as in the number pattern 9xxxxxxx. Wildcards (x) are substituted with the digits from the original number dialed by the user. Make sure the number of wildcards in the dialed number is the same as the number of wildcards in the number pattern.
   - **Comment** Enter a comment or description for this dialing rule. You can use the comment to describe what the rule does, for example, "Add a 9 to outgoing calls."
5. Click **OK** to save the dialing rule. You can continue to enter rules, using the same dialing rule group name for rules that you want to authorize together.

Authorize calls using dialing rules

Set up client voice mail features > Allow users to make calls > Allowing users to make calls procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

By default, users aren't able to place outgoing calls. To specify the kinds of calls users can make, you first create dialing rules, then authorize groups of these dialing rules on UM dial plans, UM mailbox policies, or UM auto attendants. Before you can authorize dialing rule groups, you have to define dialing rules on a UM dial plan. For details, see Create dialing rules for users.

Each dialing rule that you create will contain the types of calls or number patterns that you want to give users access to. You can allow different types of users to make different types of calls. The calls you allow can be within a country or region, or they can be international.

To authorize or restrict dialing, the following settings must be configured correctly:

- **Dialing rules** Dialing rules define the number that UM-enabled users dial and the number that will be sent from Unified Messaging and dialed by the Private Branch eXchange (PBX) or IP PBX. You create a dialing rule group by adding a dialing rule. After you create a dialing rule group, you add it to the list of authorized calls for an in-country/region or international dialing rule group.
- **Dialing rule groups** Dialing rule groups determine the types of calls that users within the dialing group can make.
- **Dialing authorizations** Dialing authorizations are used to determine the restrictions that will be applied to prevent users from incurring unnecessary telephone charges or from dialing long-distance calls.

How do I authorize a dialing rule group?

Where you authorize dialing rule groups depends on the types of callers that you want to allow to make outgoing calls. For example, if you want only Outlook Voice Access users to place outgoing calls, you would create your dialing rules and then authorize those dialing rule groups to the UM mailbox policy that the Outlook Voice Access users are linked to. The following table shows how to authorize calls for different types of callers.

Type of caller	Authorize dialing rule groups here
Unauthenticated callers who call in to an Outlook Voice Access number and don't enter a PIN	UM dial plan. For details, see Authorize calls for users in a dial plan.
Authenticated callers who call in to an Outlook Voice Access number and enter a PIN	UM mailbox policy for the caller. For details, see Authorize calls for a group of users.
Unauthenticated callers who call in to a telephone number	UM auto attendant. For details, see Authorize calls for auto

that's configured on a UM auto attendant	attendant callers.	

Depending on which users you're authorizing to make outbound calls, you'll use the **Dialing authorization** page in the Exchange admin center (EAC) for the dial plan, the auto attendant, or the UM mailbox policy.

Authorize calls for auto attendant callers

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

You can enable dialing authorizations on a Unified Messaging (UM) auto attendant. Dialing authorizations on an auto attendant are used to prohibit users who call in to the auto attendant from making in-country/region or international telephone calls, or *outdialing*. Outdialing happens when Unified Messaging makes an outgoing call for a user after they've called into a phone number that is configured on a UM auto attendant.

For additional management tasks related to outdialing, see Allowing users to make calls procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM auto attendant has been created. For detailed steps, see Create a UM auto attendant.
- Before you perform these procedures, confirm that in-country/region and international dialing rules have been created on a UM dial plan. For detailed steps, see Create dialing rules for users.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable dialing authorizations on a UM auto attendant for in-country/region rule groups

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** ✎.
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to create a dialing authorization, and then click **Edit** ✎.
3. On the **UM Auto Attendant** page > **Dialing authorization**, click **Add +** under **Authorized in-country/region dialing rule groups**.
4. On the **Select Dialing Rule Groups to Allow** page, select the dialing rule group, click **OK**, and then click **Save**.

Use the EAC to enable dialing authorizations on a UM auto attendant for international rule groups

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** ✎.
2. On the **UM Dial Plan** page, under **UM Auto Attendants**, select the UM auto attendant for which you want to create a dialing authorization, and then click **Edit** ✎.
3. On the **UM Auto Attendant** page > **Dialing authorization**, click **Add +** under **Authorized international dialing rule groups**.
4. On the **Select Dialing Rule Groups to Allow** page, select the dialing rule group, click **OK**, and then click **Save**.

Use the Shell to enable in-country/region and international dialing authorizations on a UM auto attendant

This example enables the InCountry/RegionGroup1, InCountry/RegionGroup2. InternationalGroup1, and InternationalGroup2 dialing authorizations on a UM auto attendant named `MyUMAutoAttendant`.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
AllowedInCountryOrRegionGroups InCountry/
RegionGroup1,InCountry/RegionGroup2 -
AllowedInternationalGroups
```

InternationalGroup1,InternationalGroup2

Authorize calls for users in a dial plan

Allow users to make calls > Allowing users to make calls procedures > Authorize calls using dialing rules >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

You can enable dialing authorizations on a Unified Messaging (UM) dial plan. Dialing authorizations on a dial plan are used to prohibit unauthenticated Outlook Voice Access users from making in-country/region or international telephone calls, or *outdialing*. Outdialing happens when Unified Messaging places an outgoing call for a user after they've called in to an Outlook Voice Access phone number that is configured on a UM dial plan. When you configure a setting on a UM dial plan, that setting applies to all unauthenticated users that call in to an Outlook Voice Access number.

For additional management tasks related to outdialing, see Allowing users to make calls procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform this procedure, confirm that in-country/region and international dialing rules have been created on a UM dial plan. For detailed steps, see Create dialing rules for users.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable dialing authorizations on a UM dial

plan for in-country/region dialing rule groups

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, click **Configure**.
3. On the **UM Dial Plan** page > **Dialing authorization**, click **Add +** under **Authorized in-country/region dialing rule groups**.
4. On the **Select Dialing Rule Groups to Allow** page, select the dialing rule group, click **OK**, and then click **Save**.

Use the EAC to enable dialing authorizations on a UM dial plan for international dialing rule groups

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, click **Configure**.
3. On the **UM Dial Plan** page > **Dialing authorization**, click **Add +** under **Authorized international dialing rule groups**.
4. On the **Select Dialing Rule Groups to Allow** page, select the dialing rule group, click **OK**, and then click **Save**.

Use the Shell to enable in-country/region and international dialing authorizations on a UM dial plan

This example enables the InCountry/RegionGroup1, InCountry/RegionGroup2, InternationalGroup1, and InternationalGroup2 dialing authorizations on a UM dial plan named `MyUMDialPlan`.

```
Set-UMDialPlan -Identity MyUMDialPlan -
AllowedInCountryOrRegionGroups InCountry/
RegionGroup1,InCountry/RegionGroup2 -
AllowedInternationalGroups
InternationalGroup1,InternationalGroup2
```

Authorize calls for a group of users

Allow users to make calls > Allowing users to make calls procedures > Authorize calls using dialing rules >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

You can enable dialing authorizations on a Unified Messaging (UM) mailbox policy. You can use dialing authorizations on a mailbox policy to prohibit authenticated Outlook Voice Access users that are linked to the UM mailbox policy from making in-country/region or international telephone calls, or *outdialing*. Outdialing happens when Unified Messaging places an outgoing call for a user after they've called in to an Outlook Voice Access phone number that is configured on a UM dial plan. When you configure a setting on a UM mailbox policy, that setting applies to all UM-enabled users linked with the UM mailbox policy.

For additional management tasks related to outdialing, see Allowing users to make calls procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that in-country/region and international dialing rules have been created on a UM dial plan. For detailed steps, see Create dialing rules for users.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable dialing authorizations on a UM mailbox policy for in-country/region dialing rule groups

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy for which you want to create a dialing authorization, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **Dialing authorization**, click **Add +** under **Authorized in-country/region dialing rule groups**.
4. On the **Select Dialing Rule Groups to Allow** page, select the dialing rule group, click **OK**, and then click **Save**.

Use the EAC to enable dialing authorizations on a UM mailbox policy for international dialing rule groups

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy for which you want to create a dialing authorization, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **Dialing authorization**, click **Add +** under **Authorized international dialing rule groups**.
4. On the **Select Dialing Rule Groups to Allow** page, select the dialing rule group, click **OK**, and then click **Save**.

Use the Shell to enable in-country/region and international dialing authorizations on a UM mailbox policy

This example enables the InCountry/RegionGroup1, InCountry/RegionGroup2, InternationalGroup1, and InternationalGroup2 dialing authorizations on a UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
AllowedInCountryOrRegionGroups InCountry/
RegionGroup1,InCountry/RegionGroup2 -
AllowedInternationalGroups
InternationalGroup1,InternationalGroup2
```

Allow users to see a voice mail transcript

Exchange Server 2013 > Unified Messaging > Set up client voice mail features >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-23*

Voice Mail Preview is a feature that's available to users who receive their voice mail messages from Unified Messaging (UM). Voice Mail Preview enhances the existing UM voice mail functionality by providing a text version of audio recordings. The voice mail text is displayed in email messages within Microsoft Outlook Web App, Outlook 2010 and later versions, and in other supported email programs. For more information, see Microsoft Speech Technologies.

Do users need to use a specific email program?

No. Voice Mail Preview is included in the message body text of any email program, including mobile programs. Although users can use other email programs to receive voice messages, Outlook and Outlook Web App provide a better experience. For example, in Outlook 2010 and later versions, when a specific word is clicked in the Voice Mail Preview text, the audio playback of the voice message will start to play at that word. This is useful for listening to a specific part of a voice message.

Can users search for specific voice mail messages?

Yes. Words and phrases in the Voice Mail Preview text are automatically indexed, so voice messages will appear in search results. In Outlook 2010 and later versions or in Outlook Web App, users can also use the **Audio Notes** box to add text about a voice message. These notes are also included in searches, to make it easier to locate a message.

Why is this feature called “Voice Mail Preview”?

It’s important to set users’ expectations correctly. Voice Mail Preview doesn't necessarily produce text that’s the same as what callers say in their voice messages. In fact, it's usually inaccurate in some way. To call it transcription would suggest a more perfect result than can generally be achieved. Preview suggests that the reader should be able to understand the gist of the voice content, which is closer to the real capability of the feature.

What makes the Voice Mail Preview text more or less accurate?

The accuracy of the Voice Mail Preview text depends by many factors and sometimes those factors can't be controlled. However, Voice Mail Preview text is likely to be more accurate when:

- The caller leaves a simple voice message that doesn't include slang terms, technical jargon, or unusual words or phrases.
- The caller uses a language that's easily recognized and translated by the voice mail system. Generally, voice messages left by callers who don't speak too quickly or too softly and who don't have strong accents will produce more accurate sentences and phrases.
- The voice message is free of background noise, echo, and the audio doesn't drop out.

Which languages can be used with Voice Mail Preview?

Voice Mail Preview text is available in the following languages:

- English (US) (en-US)
- English (Canada) (en-CA)
- French (France) (fr-FR)
- Italian (it-IT)
- Polish (pl-PL)
- Portuguese (Portugal) (pt-PT)
- Spanish (Spain) (es-ES)

If you have an on-premises or hybrid deployment of UM, you can download the UM language packs from the Microsoft Download Center.

If you have an on-premises or hybrid deployment, after you install a UM language pack, the dial plans and auto attendants can be configured to use the language you've chosen. For online customers, you don't have to install any UM language packs. Many companies have only one UM dial plan. UM will try to create a voice mail preview in the default dial plan language, but will only be successful if the default language supports Voice Mail Preview. A UM dial plan can only be configured to create voice mail previews in one language at a time.

To configure UM to provide voice mail previews in a language other than en-US, follow these steps:

1. Verify that Voice Mail Preview is supported in the language you want to use.
2. If you have an on-premises or hybrid deployment, download and install the appropriate UM language pack. Downloading and installing the language pack doesn't configure the dial plan default language.
3. Configure the dial plan with the language that will be used for Voice Mail Preview. For more information, see Set the default language on a dial plan.

How Voice Mail Preview displays text in the supported languages depends on the type of voice message that's sent. There are two types:

- **Voice messages that are recorded when a user doesn't answer their phone**

For these messages, the language used for Voice Mail Preview is determined by the caller's spoken language and whether the language is supported. For example, if a caller leaves a voice message in Italian, the Voice Mail Preview text will appear in Italian if Italian has been configured on the dial plan. However, if a caller leaves a message in Japanese, no Voice Mail Preview text will be included with the message because Japanese isn't available.

- **Voice messages that are sent to by an Outlook Voice Access user**

For messages sent by an Outlook Voice Access user, the language that's used for Voice Mail Preview is controlled by the voice mail administrator. Thus, the Voice Mail Preview text will be in the same language as the voice mail system. However, if a caller speaking a language that's not supported for Voice Mail Preview uses Outlook Voice Access to leave a message, no Voice Mail Preview text will be included with the message. To learn more about Outlook Voice Access, see Setting up Outlook Voice Access.

Does UM know when a voice mail preview is inaccurate?

The confidence level is determined for each voice mail preview included with a voice message. The

voice mail system measures how well the sounds in the recording match the words, numbers, and phrases. If matches are found easily, the confidence level is high. A higher level of confidence is generally associated with a higher accuracy.

If the confidence level is determined to be lower than a certain value, the phrase **Voice Mail Preview (confidence is low)** is included above the Voice Mail Preview text. If the confidence level is low, it's likely that the Voice Mail Preview text will be inaccurate.

Unified Messaging uses Automatic Speech Recognition (ASR) to calculate its confidence in the preview, but it has no way to determine which words are wrong and which are correct.

However, UM does try to learn to improve accuracy of its voice mail previews. For example, it tries to match the caller's telephone number (if provided) with the user's personal Contacts and your organization's address book or contacts from social networks. If UM finds a match, it will include the name of the caller, along with its standard lists of names and words, when running ASR on the voice recording.

Can Voice Mail Preview be used if it isn't completely accurate?

Users may have a better experience with Voice Mail Preview if they don't try to read the preview too carefully, word by word. Instead, they should look for names, phone numbers, and phrases such as "Call me back" or "I need to talk" that may provide clues about the purpose of the call.

Voice Mail Preview isn't expected to dictate messages exactly, but it can help users answer questions such as the following:

- Is this voice message related to my work?
- Is this voice message important to me?
- Did the caller leave a number? Is it different from any numbers that I may have listed for them?
- Does the caller consider this voice message urgent?
- Should I step out of a meeting to call this person back?
- I was expecting a call to confirm my request. Is this the confirmation call?

Can Voice Mail Preview be turned on or off?

Yes. If you've enabled Voice Mail Preview, users can turn it on or off using Outlook 2010 or a later version or Outlook Web App. However, the dial plan language must support Voice Mail Preview and the UM language pack for that language must be installed.

Although Voice Mail Preview settings are the same whether a user is using Outlook 2010 or a later version or Outlook Web App, they'll access them differently:

Outlook Web App

To access the Voice Mail Preview settings in Outlook Web App, users click **Settings** > **phone** > **Voice mail**. On the **Voice mail** page, the settings are available under **voice mail preview**.

By default, both Voice Mail Preview options are available when a user is enabled for Unified Messaging. If the UM dial plan is configured to use a UM language pack that supports Voice Mail Preview, Unified Messaging will create voice mail previews for users when:

- A caller leaves a voice mail message because the user doesn't answer their phone.
- A UM-enabled user signs in to Outlook Voice Access and records a voice message for one or more recipients.

When a caller leaves a voice message, and **Include preview text with voice messages I receive** is selected, Unified Messaging will create a voice mail preview in the email message, attach the audio file, and send it to the recipient's mailbox. You may want to disable this option if the language that's configured on the dial plan doesn't include Voice Mail Preview support and you don't want voice mail previews included in voice mail messages.

When users sign in to Outlook Voice Access and they send a voice message to another user, they may want to clear the **Include preview text with voice messages I send through Outlook Voice Access** check box. For example, they might want to do this if they're sending voice messages in a language that Voice Mail Preview doesn't support or if they don't want to include the voice mail preview with the voice message because it's too long.

Voice Mail Preview advisor

Unified Messaging > Set up client voice mail features > Allow users to see a voice mail transcript >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

Microsoft Exchange Unified Messaging (UM) includes a feature called Voice Mail Preview, which uses automatic speech recognition (ASR) to add a text version of the voice mail audio file to voice mail messages. ASR isn't entirely accurate, especially when it's used to record audio over a phone that contains unknown voices and noises. Some organizations require consistently error-free (or near-error-free) transcripts of voice messages. The Voice Mail Preview Partner program can help such organizations meet those requirements.

Voice Mail Preview uses Microsoft speech technologies to provide a text version of audio recordings. The voice mail text is displayed in email messages within Microsoft Outlook Web App, Outlook 2010 or later versions, and other email programs.

By default, when you enable a user for UM in an on-premises or hybrid deployment, voice mail previews will be sent if a supported UM language pack is installed. When you enable a user for UM in Exchange Online, all the UM language packs are installed. However, Voice Mail Preview isn't supported in all languages that are installed.

There are Voice Mail Preview partners that offer enhanced transcription support and services for the

Voice Mail Preview feature. These partners employ people to correct voice mail transcriptions that were created using ASR. Each Voice Mail Preview partner must meet a set of requirements to be certified to interoperate with Exchange UM.

If you determine that the voice mail previews sent to your users aren't accurate enough, you can contact one of the certified Voice Mail Preview partners listed at Microsoft Pinpoint and sign up with them at an additional cost.

Contents

Overview

When Unified Messaging records the audio for a voice message, it uses ASR to create voice mail preview text from the audio file, and then submits the whole voice message for delivery to the user. For each voice message that's created, Unified Messaging determines a confidence level for the voice mail preview included with the message. It measures how well the sounds in the recording match the words, numbers, and phrases in the message. If the system finds matches easily, the confidence level will be high. A higher level of confidence is generally associated with a higher accuracy.

The accuracy of voice mail preview text depends on many factors, and sometimes those factors can't be controlled. However, the text is likely to be more accurate when:

- A simple voice message is left, and the caller doesn't use slang terms, technical jargon, or unusual words or phrases.
- The caller uses a language that's easily recognized and translated by the voice mail system. Generally, voice messages left by callers who don't speak too quickly or too softly and who don't have strong accents will produce more accurate sentences and phrases.
- The voice message is free of background noise and echoes, and the audio doesn't drop out.

Most customers who use Unified Messaging find that the voice mail previews are accurate enough for their users. However, when ASR is applied to recordings made over the phone by unknown voices and background noises, the voice mail preview text usually isn't completely accurate. If the level of confidence is consistently low or the voice mail previews that are received aren't very accurate, you can increase the accuracy of the voice mail previews that users receive as follows:

- Sign up for a voice transcription service from a Voice Mail Preview partner.
- After you've signed up with a Voice Mail Preview partner, set the partner up to work with UM. For more information about how to configure UM for a Voice Mail Preview partner, see Configure Voice Mail Preview partner services for users.

When you've signed up with a Voice Mail Preview partner, the Exchange servers in your organization redirect voice messages with the audio file attached to the Voice Mail Preview partner instead of generating voice mail preview text for voice messages and submitting the voice messages to the user's mailbox. The email message with the voice mail preview text produced by the Voice Mail Preview partner is then submitted to the Exchange servers in your organization for delivery to the recipient's mailbox.

Important:
We recommend that all customers who plan to deploy Unified Messaging obtain the assistance of a UM specialist. A UM specialist helps you ensure that there's a smooth transition to UM from a legacy voice mail system. Performing a new deployment or upgrading a legacy voice mail system requires significant knowledge about VoIP gateways, IP PBXs, PBXs, session border controllers (SBCs), and Unified Messaging. For more information about how to contact a UM specialist, see the Microsoft Exchange Server 2013 Unified Messaging (UM) Specialists or Microsoft Pinpoint for Unified Messaging.

Return to top

Exchange Unified Messaging Voice Mail Partner program

To become certified as a Voice Mail Preview partner that interoperates with Exchange UM, the partner must implement the requirements contained in the Voice Mail Preview Interoperability Specification, and the partner solution must be certified by an independent certification vendor. If you're interested in certifying your transcription service to work with Exchange UM, submit a request to Voice Mail Preview Partners for Exchange Unified Messaging.

Voice Mail Preview partners certified for Exchange Unified Messaging

If you've already deployed Unified Messaging in your organization and you're looking for a certified Voice Mail Preview partner to provide transcription support services, see Microsoft PinPoint. These software vendors have been certified as interoperable with Exchange UM.

Configuring Voice Mail Preview partners

After UM has been configured, it forwards voice messages with the audio to a dedicated Voice Mail Preview partner, which then takes the audio file and creates the voice mail preview text. However, to allow users to receive the voice mail preview with their voice message in their mailbox, you must configure a UM mailbox policy, associate users with the UM mailbox policy, and then have the users verify that they can receive voice mail previews in their voice messages in Outlook 2010 or a later version or Outlook Web App. For more information about how to configure UM for a Voice Mail

Preview partner, see Configure Voice Mail Preview partner services for users.

VoIP or media gateways and IP PBX support

Configuring VoIP gateways and IP PBXs for your organization is a difficult deployment task that must be completed correctly to successfully deploy Unified Messaging with a Voice Mail Preview partner. For information that can help you configure your VoIP gateways and IP PBXs, and for the most up-to-date information about how to configure them, see Telephony advisor for Exchange 2013 or Configuration notes for supported VoIP gateways, IP PBXs, and PBXs.

Testing interoperability of Exchange UM with VoIP gateways has been integrated with the Microsoft Unified Communications Open Interoperability Program. For more information, see Microsoft Unified Communications Open Interoperability Program.

Return to top

Voice Mail Preview procedures

Unified Messaging > Set up client voice mail features > Allow users to see a voice mail transcript >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-01-30*

Configure Voice Mail Preview partner services for users

Set the Voice Mail Preview partner address

Set the Voice Mail Preview partner ID

Set the maximum message duration for a Voice Mail Preview partner

Set the maximum delivery delay for a Voice Mail Preview partner

Enable Voice Mail Preview for users

Disable Voice Mail Preview for users

Configure Voice Mail Preview partner services for users

Set up client voice mail features > Allow users to see a voice mail transcript > Voice Mail

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-13*

You can configure a Voice Mail Preview partner on a Unified Messaging (UM) mailbox policy. After you've configured Voice Mail Preview partner settings, such as the Voice Mail Preview partner ID and Voice Mail Preview partner address, on a UM mailbox policy, the settings you configure will apply to all UM-enabled users who are linked with that mailbox policy.

Note:
You must use the Shell to configure a Voice Mail Preview partner.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

How do you do this?

Step 1: Sign up with a partner service

To find the list of certified partners and detailed instructions for how to sign up, see Voice Mail Preview advisor or see the Microsoft PinPoint website. After you've signed up, the Voice Mail Preview partner will provide you a partner ID and the SMTP address to use to forward the voice messages.

In Step 2, you'll apply the Partner ID and SMTP address you acquired in Step 1 to the required UM mailbox policies.

Step 2: Set the Voice Mail Preview partner address and ID

This example sets the Voice Mail Preview partner address to exumvmp@fabrikam.com and the Voice Mail Preview partner ID to CON123-2010 on a UM mailbox policy named *MyUMMailboxPolicy*.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
VoiceMailPreviewPartnerAddress exumvmp@fabrikam.com
-VoiceMailPreviewPartnerAssignedID CON123-2010
```

Step 3: Configure advanced Voice Mail Preview partner settings

If the partner requires custom settings, you may want to set two additional parameters for a Voice Mail Preview partner as follows:

- *VoiceMailPreviewPartnerMaxMessageDuration*
- *VoiceMailPreviewPartnerMaxDeliveryDelay*

This example sets the maximum message duration to 300 seconds (5 minutes) and the maximum delivery delay to 600 seconds (10 minutes) on a UM mailbox policy named *MyUMMailboxPolicy*.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
VoiceMailPreviewPartnerMaxMessageDuration 300 -
VoiceMailPreviewPartnerMaxDeliveryDelay 600
```

Step 4: Assign a UM-enabled user to the UM mailbox policy for a Voice Mail Preview partner

If you want to configure the Voice Mail Preview partner service for some, but not all, UM-enabled users in a UM dial plan, you must create a new UM mailbox policy and configure the partner settings. When you've finished, you can apply the new policy to selected UM-enabled users. For more information about how to assign a UM-enabled user to a UM mailbox policy, see the following topics:

- Assign a UM mailbox policy
- Set-UMMailbox

For more information about the Voice Mail Preview partner program, see Voice Mail Preview advisor.

Set the Voice Mail Preview partner address

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-13*

You can set a Voice Mail Preview partner address on a Unified Messaging (UM) mailbox policy. After you've set the Voice Mail Preview partner address on a UM mailbox policy, the setting will apply to all UM-enabled users who are linked with that mailbox policy.

Note:

You must use the Shell to set a Voice Mail Preview partner address.

For more information about the Voice Mail Preview partner program, see Voice Mail Preview advisor.

For additional management tasks related to Voice Mail Preview, see Voice Mail Preview procedures.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to set the Voice Mail Preview partner address on a UM mailbox policy

This example sets the Voice Mail Preview partner address to exumvmp@fabrikam.com on a UM mailbox policy named *MyUMMailboxPolicy*.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
VoiceMailPreviewPartnerAddress exumvmp@fabrikam.com
```

Set the Voice Mail Preview partner ID

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-13*

You can set a Voice Mail Preview partner ID on a Unified Messaging (UM) mailbox policy. After you've set the Voice Mail Preview partner ID on a UM mailbox policy, the setting will apply to all UM-enabled users who are linked with that mailbox policy.

Note:
You must use the Shell to set the Voice Mail Preview partner ID.

For more information about the Voice Mail Preview partner program, see Voice Mail Preview advisor.

For additional management tasks related to voice mail preview, see Voice Mail Preview procedures.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to set the Voice Mail Preview partner ID on a UM mailbox policy

This example sets the Voice Mail Preview partner ID to CON123-2010 on a UM mailbox policy named *MyUMMailboxPolicy*.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy
-VoiceMailPreviewPartnerAssignedID CON123-2010
```

Set the maximum message duration for a Voice Mail Preview partner

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-13*

You can set the maximum message duration for a Voice Mail Preview partner on a Unified Messaging (UM) mailbox policy. After you've set the maximum message duration, the setting will apply to all UM-enabled users who are linked with that mailbox policy.

Note:

You must use the Shell to set the maximum message duration for a Voice Mail Preview partner.

For more information about the Voice Mail Preview partner program, see Voice Mail Preview advisor.

For additional management tasks related to Voice Mail Preview, see Voice Mail Preview procedures.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server,

Exchange Online, or Exchange Online Protection..

Use the Shell to set the maximum message duration for a Voice Mail Preview partner

This example sets the maximum message duration for a Voice Mail Preview partner to 300 seconds (5 minutes) on a UM mailbox policy named *MyUMMailboxPolicy*.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
VoiceMailPreviewPartnerMaxMessageDuration 300
```

Set the maximum delivery delay for a Voice Mail Preview partner

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-13*

You can set the maximum delivery delay for a Voice Mail Preview partner on a Unified Messaging (UM) mailbox policy. After you've set the maximum delivery delay, the setting will apply to all UM-enabled users who are linked with that UM mailbox policy.

Note:
You must use the Shell to set the maximum delivery delay for a Voice Mail Preview partner.

For more information about the Voice Mail Preview partner program, see Voice Mail Preview advisor.

For additional management tasks related to voice mail preview, see Voice Mail Preview procedures.

What do you need to know before you begin?

- Estimated time to complete: 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed

steps, see Create a UM mailbox policy.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to set the maximum delivery delay for a Voice Mail Preview partner

This example sets the maximum delivery delay to 600 seconds (10 minutes) on a UM mailbox policy named *MyUMMailboxPolicy*.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
VoiceMailPreviewPartnerMaxDeliveryDelay 600
```

Enable Voice Mail Preview for users

Set up client voice mail features > Allow users to see a voice mail transcript > Voice Mail Preview procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

You can enable the Voice Mail Preview feature for users associated with a Unified Messaging (UM) mailbox policy if it has been disabled. Enabling this setting allows users to receive the text of a voice mail message in the message body of an email or text message. The default setting is enabled.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.

- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable Voice Mail Preview

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **General**, select the check box next to **Allow voice mail preview**.
4. Click **Save**.

Use the Shell to enable Voice Mail Preview

This example allows users who are associated with the UM mailbox policy `MyUMMailboxPolicy` to use the Voice Mail Preview feature.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowVoiceMailPreview $true
```

Disable Voice Mail Preview for users

Set up client voice mail features > Allow users to see a voice mail transcript > Voice Mail Preview procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

You can disable the Voice Mail Preview feature for users associated with a Unified Messaging (UM) mailbox policy. Disabling this setting prevents users from receiving the text of a voice mail message in the message body of an email or text message. The default setting is enabled.

For additional management tasks related to UM mailbox policies, see UM mailbox policy

procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to disable Voice Mail Preview

1. In the EAC, navigate to **Unified Messaging** > **UM Dial plans**, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **General**, clear the check box next to **Allow voice mail preview**.
4. Click **Save**.

Use the Shell to disable Voice Mail Preview

This example prevents users who are associated with the UM mailbox policy `MyUMMailboxPolicy` from using the Voice Mail Preview feature.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowVoiceMailPreview $false
```

Enable voice mail users to receive faxes

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

Microsoft Exchange Unified Messaging (UM) enables voice mail messages to be delivered to a user's mailbox and also lets users receive faxes in their mailbox. In UM, a fax is sent to the user's mailbox as an email message that has an image file with a .tif extension attached. The user can open the attached file by using a software application that can open and view image files that have a .tif extension. This topic discusses faxing and how it works in UM.

Note:

Although Unified Messaging doesn't let users send outgoing faxes, many third-party solutions, such as an Internet fax service, an email faxing service, or a third-party fax server application, can be used to send outgoing faxes.

Contents

Overview of faxing

Fax is an abbreviation for the word facsimile. It's a technology that's used to electronically transfer documents. Generally, faxes are sent and received by fax machines or computer fax modems by using the Public Switched Telephone Network (PSTN), which is a telephony or circuit-based network. However, other options can be used to send and receive faxes.

Most organizations today want their users to be able to send and receive faxes. Organizations use one or more of the methods described in the following list to send or receive faxes over the PSTN or over the Internet. There are advantages and disadvantages to each of these methods.

- Traditional fax machines and computer-based faxing
- Faxing by using fax servers or fax gateways
- Faxing by using a Voice over IP (VoIP) network
- Faxing by using an email client application

To send a fax message, users in an organization may have to do the following:

- Print a hard copy of the document to be faxed and use a physical fax machine to send it.
- Save the document on their computer and use a fax modem to send the fax, but this requires a phone line connected to the computer.
- Use an Internet fax service that lets them fax a document from a vendor-specific software application.
- Send an outgoing fax to a fax server by using a software application that's configured to use the fax server.

To receive a fax, users in an organization may have to do the following:

- Receive a fax on a physical fax machine within the organization.
- Receive a fax by using a fax modem that's installed on their computer.
- Receive a fax from an Internet faxing service.
- Receive a fax from a fax server that's configured on a network.
- Receive a fax from a fax partner's server that uses Unified Messaging to deliver the fax using a VoIP network.

Return to top

Faxing methods

There are several options for sending and receiving faxes, including the following:

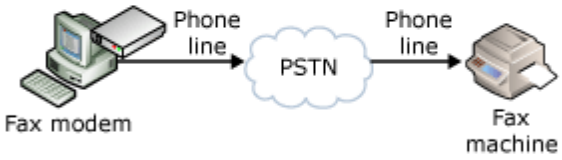
Traditional fax machines and computer-based faxing A scanner, a fax modem in a computer, a printer with built-in faxing capabilities, or a dedicated fax machine can be used to send and receive faxes. All of these can be used to transmit data in the form of pulses by using a telephone line to another fax device, usually another fax machine or computer that has a fax modem. The pulses are then transformed into images or used to print the image on paper.

The traditional fax method requires at least a single telephone line on the sending and receiving device, and only one fax can be sent or received at a time. A disadvantage of sending and receiving faxes by using a fax modem is that the computer must be turned on and running fax software or a fax service. This kind of computer-based faxing doesn't use the Internet to send or receive faxes.

Traditional and computer-based faxing

Traditional fax machine

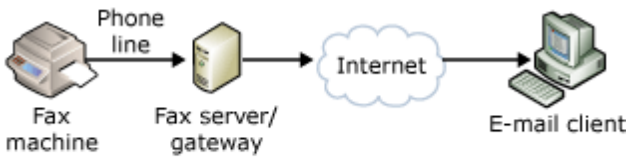
Fax modem



Fax servers or gateways and Internet fax services There are several ways to send and receive faxes over the Internet. These include using a software application on a computer or using an email client to receive faxes. In most cases, this kind of faxing involves using a fax server or fax gateway to convert between faxes and email. This method has become increasingly popular because it enables organizations to remove or avoid purchasing additional fax machines. It also eliminates the need to install additional telephone lines. This kind of faxing involves creating a document, including a fax cover page with the correct identifying information, and then sending the document to a traditional fax machine. For example, the user uses a software application such as Microsoft Word or Microsoft Outlook to create and send the fax to the fax server or gateway. The fax server or gateway receives the fax and then sends it by using a traditional telephone line to a fax machine or a fax modem that's installed on a computer.

Faxing by using fax servers or gateways

Fax Receiving



Sending a fax

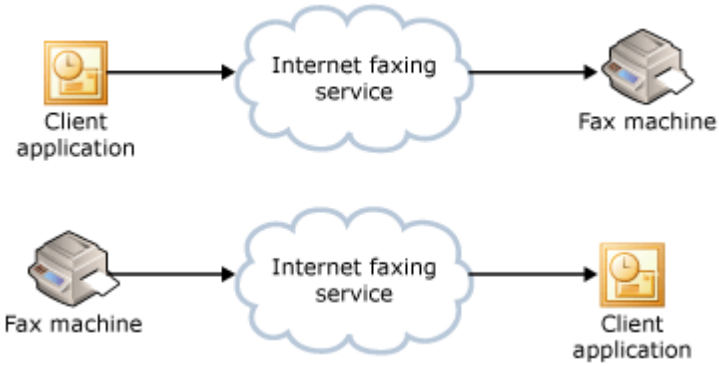
Internet fax services let a user send faxes from a computer by using the Internet. A software application such as Word or Outlook can be used to create and send the fax to an Internet fax service. Many companies offer Internet fax services on a subscription basis or by charging for each fax message that's sent. Internet fax services offer the following advantages:

- No fax machine is required.
- No software or hardware must be installed.

- No dedicated telephone lines are required.
- Confidentiality.
- Multiple faxes can be sent at the same time.
- Faxes can be received when the computer is turned off.

The following figure shows how Internet fax services can be used to send and receive faxes.

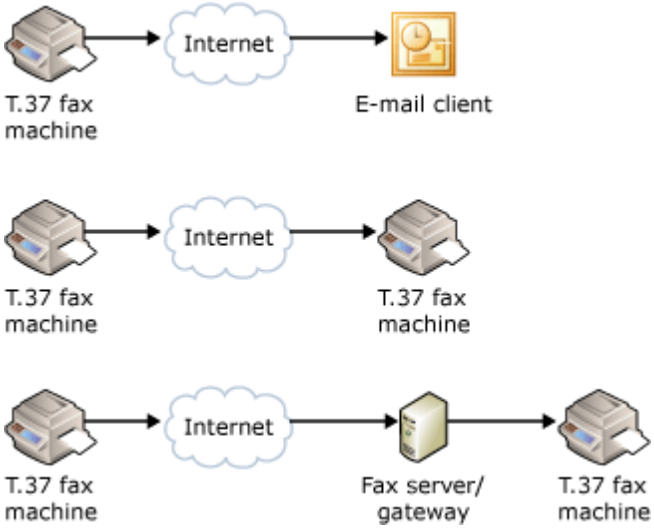
Internet fax services



Faxing by using an email client application Faxes can be sent and received by a fax machine over the Internet and then received by an email client such as Outlook.

The T.37 protocol was designed to enable a fax machine to send fax messages over the Internet to an email client. The faxes are sent over the Internet as email attachments, typically as .tif or .pdf files. In this kind of faxing, a fax machine that supports iFax or T.37 is required, in addition to an email address for the sending and receiving fax machines. To work with traditional fax machines and fax modems, all T.37 fax machines support standard faxing by using a telephone line. However, in some cases, T.37 fax machines can be used when a fax gateway is also being used. The following figure shows how T.37-based fax machines and email clients can be used to send and receive faxes.

Faxing with email



Faxing by using a VoIP network VoIP is a technology that provides hardware and software that enables people to use an IP-based network as the transmission medium for telephone calls. On a VoIP network, voice and fax data is sent in packets by using IP instead of traditional circuit transmissions or the circuit-switched telephone lines of the PSTN. A VoIP gateway that you connect to your IP network uses VoIP to send voice data packets between a Client Access server running the

Microsoft Exchange Unified Messaging Call Router service and a Mailbox server running the Microsoft Exchange Unified Messaging service and a Private Branch eXchange (PBX) system. You can also use an IP PBX to perform the functions of both a VoIP gateway and a PBX.

There are two basic types of networks: circuit-switched and packet-switched. A circuit-switched network is a network in which there exists a dedicated connection. A dedicated connection is a circuit or channel that's set up between two nodes so that they can communicate. After a call is established between two nodes, the connection can be used only by these two nodes. When the call is ended by one of the nodes, the connection is canceled. In circuit-switched networks, such as the PSTN, multiple calls are transmitted across the same transmission medium. Frequently, the medium that's used in the PSTN is copper. However, fiber optic cable might also be used.

In packet-switched networks such as the Internet or a local area network (LAN), packets are routed to their destination through the most expedient route, but not all packets traveling between two hosts travel the same route, even those from a single message. This almost guarantees that the packets will arrive at different times and out of order. In a packet-switched network, packets (messages or fragments of messages) are individually routed between nodes over data links that may be shared by other nodes. With packet switching, unlike circuit switching, multiple connections to nodes on the network share the available bandwidth. Packet-switched networking has made it possible for the Internet to exist and, at the same time, has made data networks—especially LAN-based IP and VoIP networks—more available and widespread.

Return to top

T.38

T.38 is a faxing standard and protocol that enables faxing over an IP-based network. An IP-based network that uses the T.38 protocol uses Simple Mail Transfer Protocol (SMTP) and Multipurpose Internet Mail Extension (MIME) to send a message to a recipient's mailbox. T.38 allows for IP fax transmissions for IP-enabled fax devices and fax gateways. The devices can include IP network-based hosts such as client computers and printers. In Exchange Unified Messaging, the fax images are separate documents encoded as .tif files and attached to an email message. Both the email message and the .tif file attachment are sent to the user’s UM-enabled mailbox.

Unified Messaging relies on the gateway's abilities to translate or convert Time Division Multiplex (TDM) or telephony circuit-switched based protocols like Integrated Services Digital Network (ISDN) and QSIG from a PBX to IP-based or VoIP-based protocols like Session Initiation Protocol (SIP), Real-Time Transport Protocol (RTP), or T.38 for receiving fax messages. The VoIP gateway is integral to the functionality and operation of Unified Messaging. The VoIP gateway is responsible for sensing fax tones. Client Access and Mailbox servers rely on the VoIP gateway to send a notification that a fax has been detected. Then the Client Access and Mailbox servers will renegotiate the media session and use the T.38 protocol.

Return to top

Overview of faxing with Unified Messaging

In Unified Messaging, the user receives fax images as separate documents encoded as .tif image files that are attached to an email message. Both the email message and the .tif attachment are sent to the user's UM-enabled mailbox.

There are several advantages to sending a fax message to the user's mailbox. These advantages include the following:

- You can reduce the number of physical or traditional fax machines.
- The number of telephone lines used for faxing in an organization can be reduced, because the Mailbox server can queue many faxes and send each fax when one of the telephone lines becomes available.
- Faxes that are received as a .tif image file are better quality than a traditional fax. Incoming faxes can be printed by a local or shared printer.
- Faxes sent to the user's mailbox are more secure because they're less likely than hard-copy faxes to be picked up by someone other than the recipient.
- Users can receive faxes without leaving their desk.
- Fax messages that are received can be monitored to make sure that they comply with an organization's security policies.

A single fax message can be sent only to a single UM-enabled user. Unified Messaging can't forward fax messages to a distribution list. If you need to have this functionality, you must follow these steps:

1. Create a mailbox to answer the fax call. This will be the mailbox for the distribution list.
2. UM-enable the distribution list mailbox.
3. Create a rule for this UM-enabled mailbox. The rule will be configured to forward all messages to the selected distribution list.

Return to top

Receiving incoming faxes

Receiving a fax on a VoIP network differs from receiving a fax on a standard fax machine or by using a fax server that's located on an IP-based network. To enable faxes to be sent and received over a VoIP network, you must have a VoIP gateway or an IP PBX that supports the T.38 protocol and a server that also supports T.38. T.38 allows for IP-based fax transmissions for IP network-based hosts, for example, client computers, printers with built-in faxing capabilities, and servers such as a Mailbox server.

Important:
Sending and receiving faxes using T.38 or G.711 isn't supported in an environment where Unified Messaging and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server are integrated.

When a PBX receives a call, it forwards the call to the appropriate extension. If there is no answer at the user's extension number, the PBX forwards the call to a VoIP gateway, and the gateway forwards the call to the appropriate Mailbox server. The Mailbox server determines whether the call is a voice call or a fax call, based on the protocol that's used for the call. When the SIP protocol is used, the Mailbox server processes the call as a voice message. However, if the T.38 protocol is used from the VoIP gateway, the Mailbox server recognizes that the call is for a fax and processes it as described in the following paragraph. A Mailbox server forwards incoming fax calls to a dedicated fax partner server, which then establishes the fax call with the fax sender and receives the fax on behalf of the UM-enabled user. The fax partner server then sends the fax included as a .tif attachment in an SMTP message to the recipient's mailbox.

When a Mailbox server receives an incoming T.38 fax signal from a VoIP gateway, the Mailbox server queries the directory using the Lightweight Directory Access Protocol (LDAP) to determine if the intended recipient of the incoming fax call is allowed to receive incoming fax messages and to determine the SIP address of the fax partner server. After it has checked this information, the Mailbox server sends a fax call referral request to the VoIP gateway or SIP peer, which then forwards the fax request on to the fax partner server. After the fax call has been successfully established, the fax sender sends the fax media and data to the fax partner server. After the fax partner server receives the fax media and data, it uses SMTP to send an email message to a Mailbox server. This message contains a .tif image of the fax message and special X-headers to the intended fax recipient.

After the fax message is authenticated and is sent from a valid fax partner server, the UM mailbox assistant on the Mailbox server issues an RPC call to the Microsoft Exchange Unified Messaging service. This ensures that the fax message properties match those of fax messages that are created by a Mailbox server. Finally, the Mailbox server again submits the final fax message, which includes the email message and .tif attachment for the incoming fax. Then, using MAPI RPC, the completed and formatted version of the fax message is delivered to the Mailbox server that contains the intended recipient's mailbox.

Return to top

Fax call referral methods

An incoming fax call can be signaled to a Mailbox server through SIP re-INVITE from the VoIP gateway or SIP peer (media gateway), CNG (calling fax tone) notification by the SIP peer, or CNG detection by the Mailbox server. The following subsections detail the fax call referral in each case.

Re-INVITE from a SIP peer

An incoming call to a UM pilot number is directed to UM as an INVITE with a voice (RTP/audio) SDP profile

1. UM accepts the invitation and media streams are established. After the call has been established, fax transmission is initiated by the caller.

2. The SIP peer detects the calling fax tone (CNG). The SIP peer issues a re-INVITE to the Mailbox server, this time specifying a fax (T.38 or G.711) profile in the SDP.
3. UM responds to the invitation with a 200 OK that places the SIP peer “on hold”.
4. UM issues a REFER, referring the SIP (CNG) peer to a fax partner solution end point, obtained from its configuration data.
5. The SIP peer sends the fax session INVITE to the fax partner solution.
6. The fax partner solution accepts the invitation, and a media session is established with the SIP peer.

CNG notification by a SIP peer

An incoming call to a UM pilot number is directed to UM as an INVITE with a voice (RTP/audio) SDP profile.

1. UM accepts the invitation and media streams are established. After the call has been established, fax transmission is initiated by the caller.
2. The SIP peer detects the calling fax tone (CNG).The SIP peer notifies the UM server of the fax by sending a CNG notification in the RTP stream, compliant with RFC 4733.
3. UM responds to the notification by immediately issuing a REFER and referring the SIP peer to a fax partner solution end point, obtained from its configuration data.
4. The SIP peer sends the fax session INVITE to the fax partner solution.
5. The fax partner solution accepts the invitation.
6. A media session is established with the SIP peer.

CNG detection by a Mailbox server

An incoming call to a UM pilot number is directed to UM as an INVITE with a voice (RTP/audio) SDP profile. UM accepts the invitation and media streams are established.

1. After the call has been established, fax transmission is initiated by the caller.
2. The Mailbox server detects the fax tone (CNG) in the RTP audio stream.
3. UM responds to the notification by immediately issuing a REFER and referring the SIP peer to a fax partner solution end point, obtained from its configuration data.
4. The SIP peer sends the fax session INVITE to the fax partner solution.
5. The fax partner solution accepts the invitation.
6. A media session is established with the SIP peer.

Return to top

Configuring faxing

By default, when you install the Mailbox server, the server isn't configured to allow incoming fax calls to be processed or delivered to a UM-enabled user. To configure UM with a fax partner server, you must configure the UM mailbox policy and configure authentication between the Mailbox server and the fax partner server. For more information, see Setting up incoming faxing.

Return to top

Telephone numbers and faxing

Exchange Unified Messaging offers the following options when you're configuring UM-enabled users to receive fax messages:

- A Direct Inward Dial (DID) telephone number for an individual user that's used for both faxes and voice mail.
- A separate DID telephone number for an individual user that's used for receiving faxes.
- A central fax telephone number for all users that receives all faxes.

A single DID telephone number

When you enable a user for Unified Messaging by using the **Enable UM mailbox** page or the **Enable-UMMailbox** cmdlet, you must specify at least a single extension number for the user. This extension number is enabled on a per-user basis and must be unique within a given dial plan. The extension is used by Unified Messaging to locate the correct user and to deliver voice and fax messages to the user's mailbox. For more information, see Enable-UMMailbox.

Using a single DID number, you can configure faxing so that a user uses a single DID number for both voice and fax. This configuration is easy to administer and doesn't waste additional DID numbers. If the user is away or on the phone when a fax call arrives, UM answers the call, detects the fax tone, creates the fax message, and sends it to the user.

If a user whose faxing is configured using a single DID telephone number receives a call from a fax machine when they’re not away or on the phone, they can:

- Not answer the telephone when it rings so that the fax call will be forwarded and answered by a Mailbox server and the fax message will be created and forwarded to the user's mailbox.
- Answer the fax call, and then transfer it to himself or herself so that the call will be forwarded and answered by a Mailbox server and the fax message will be created and forwarded to the user's mailbox.
- Wait for the caller to retry sending the fax and let the fax call be transferred to a Mailbox server.

In summary, using a single DID number requires the user to perform additional actions to be able to receive fax messages.

Return to top

Multiple DID telephone numbers

When you enable a user for Unified Messaging, you must enter at least a single extension number for that user. You can add multiple extension numbers for a UM-enabled user by using the Exchange Administration Center (EAC). For more information, see Add an extension number.

Adding multiple extension numbers is useful when a UM-enabled user:

- Receives many faxes.
- Doesn't want to be bothered with answering the phone to receive a fax.
- Doesn't want to hear a fax tone when they answer their phone.

Adding multiple extensions is more complex than using a single extension and may require additional configuration settings on a PBX or an IP PBX. To configure multiple extension numbers for a UM-enabled user, you must have DID extension numbers available that aren't being used in your organization. It isn't a good idea to use multiple numbers for a UM-enabled user if your organization has a limited number of DID extension numbers available.

The benefit of using multiple DID extension numbers is that a UM-enabled user receives voice calls on one DID extension number and fax calls on the another DID extension number. Using separate DID numbers for voice mail and fax calls is easier for the user.

If you configure two DID extension numbers for a specific user, the DID extension numbers can come from separate UM dial plans. To use two DID numbers, you can create a dial plan and use a Mailbox server as a dedicated server that will receive fax calls and forward fax messages to users. For more information, see Create a UM dial plan.

You have the following options when you're configuring multiple DID extension numbers for UM-enabled users:

- **One extension for fax without Unified Messaging and one for voice** This type of configuration is enabled on a per-user basis and is used when you have extra or unused DID extension numbers available. One DID extension number is published as the user's voice mail number and the other DID extension number is published as the user's fax number. Voice calls that are answered because the user doesn't answer the phone or a busy signal is encountered are forwarded to a Mailbox server, and a voice message is created and sent to the UM-enabled user's mailbox. The other extension number can be connected to a fax machine or to another computer that has a fax modem. With this configuration, a Mailbox server doesn't process fax calls, and fax messages aren't sent to the UM-enabled user's mailbox.
- **One extension for fax and one for voice** This type of configuration is enabled on a per-user basis and can be used when your organization has many DID extension numbers available. In this configuration, both DID extension numbers that are answered because the user doesn't answer the phone or a busy signal is encountered are forwarded to a Mailbox server, which creates a voice or fax message depending on the DID extension number that's called. Although the user publishes one number for voice and one for fax, the Mailbox server detects the type of call that's being received on the DID extension number and can create a voice or fax message from calls to either of the DID extension numbers. This is very useful when a user doesn't have a separate fax machine or a dedicated computer that has a fax modem to answer incoming fax calls.
- **One "phantom" extension for fax and one for voice** This type of configuration is enabled on a per-user basis. It's essentially the same as the configuration that uses two DID numbers (one for fax and one for voice). However, in this configuration, the number that's published for fax calls for the UM-enabled user is configured on the PBX as a "phantom" extension. Incoming calls that are received on this "phantom" DID extension number are always forwarded to a Mailbox server.

The advantage of this type of configuration is that incoming fax calls are answered by a Mailbox

server. When the phone rings but isn't answered, a fax is created and forwarded by the Mailbox server to the UM-enabled user's mailbox without disturbing the user. This happens automatically because no telephone or fax device is positioned close to the user, and the user doesn't hear the ring of the incoming call.

The disadvantages of this kind of configuration are that you must have additional DID extensions available and you must configure the PBX or IP PBX to forward the call to a Mailbox server.

Central fax telephone number

When you enable a user for Unified Messaging by using the **Enable UM Mailbox** page or the **Enable-UMMailbox** cmdlet, you must specify at least a single extension number for the user. However, if you need to configure a central fax number for incoming faxes, you must set up a separate UM enabled mailbox that is used to receive all faxes.

In some organizations, especially those that receive many faxes each day, you might want to publish one fax number for the whole organization. This fax number would be used by all callers when they submit faxes to users in the organization.

Publishing one fax number for the whole organization enables your organization to control the types of faxes that are received by users. The advantage of this configuration is that it requires only a single DID extension number or an external telephone number. Also, it doesn't require a separate DID number for faxing for each UM-enabled user. However, it does require a "fax secretary" or other person to distribute the incoming faxes to users within the organization based on information that's included on the fax cover page or in the fax message itself.

Note:
Using a central fax number with optical character recognition (OCR) isn't available in Exchange Unified Messaging. This kind of configuration can use a central fax number. However, instead of having to be routed to the recipient by a person, the faxing software receives the fax, performs OCR, and then tries to locate the recipient based on the information on the cover page or fax message.

Using a single fax number for the whole organization is useful in the following situations:

- A user within the organization receives too many faxes in their mailbox to manage them effectively.
- A user receives too many spam faxes in their mailbox.
- Business needs are too complex to warrant creating a transport rule to accept incoming faxes and route them to the intended mailbox. This might not be practical, for example, if your organization requires that you route certain faxes to one group and other faxes to another group. For more information, see Transport rules.
- Filtering fax messages by using Outlook isn't effective.

Return to top

Journaling UM fax messages

Many organizations that implement journaling may also use Unified Messaging to consolidate their email, voice mail, and fax infrastructure. However, you may not want the journaling process to generate journal reports for messages that are generated by Unified Messaging. In this case, you can decide whether to journal voice mail messages and missed call notification messages that are handled by a Mailbox server or to skip such messages. If your organization doesn't require journaling of voice mail and missed call notifications, you can reduce the hard disk space that's required to store journal reports by skipping such messages. When you enable or disable the journaling of voice mail messages and missed call notification messages, your change is applied to all Transport services in your organization. For more information, see Journaling.

Note:

Messages that contain faxes that are generated by a Mailbox server are always journaled, even if you configure a journal rule that specifies not to journal Unified Messaging voice mail and missed call notification messages.

Return to top

Fax advisor for Exchange UM

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

Microsoft Unified Messaging (UM) relies on certified fax partner solutions for enhanced fax functionality such as outbound fax or fax routing. By default, users aren't configured to allow incoming fax messages to be delivered to a UM-enabled user. Exchange servers send the fax requests to a certified fax partner solution. The fax partner's server receives the fax data and then sends it to the recipient's mailbox in an email message with the fax included as a .tif attachment. For details, see Enable voice mail users to receive faxes.

Important:

We recommend that all customers who plan to deploy Unified Messaging obtain the assistance of a Unified Messaging specialist. A Unified Messaging specialist helps you ensure that there's a smooth transition to Unified Messaging from a legacy voice mail system. Performing a new deployment or upgrading a legacy voice mail system requires significant knowledge about PBXs and Unified Messaging. For more information about how to contact a Unified Messaging specialist, see the Microsoft Exchange Server 2013 Unified Messaging (UM) Specialists or Microsoft Pinpoint for Unified Messaging.

Exchange Unified Messaging Fax Partner Program

To become a fax partner certified for interoperability with Exchange UM, the partner must implement the requirements contained in the Fax Partner Interoperability Specification and the fax solution must be certified by an independent certification vendor. For more information about certifying a fax product to work with Exchange Unified Messaging, submit a request to Fax Partners for Unified Messaging.

Fax partner solutions certified as interoperable with Unified Messaging

If you've already deployed Exchange Unified Messaging and are looking for a fax partner that can enable incoming faxes for your organization, see Microsoft Pinpoint for Fax Partners. These software vendors have been certified as interoperable with Exchange Server and include certified software solutions for Unified Messaging.

VoIP, media gateway, and IP PBX support

Correctly configuring VoIP gateways for your organization is a difficult deployment task that must be completed to successfully deploy Exchange Unified Messaging with incoming faxing. To help answer questions and get the most up-to-date VoIP gateway configuration information, see Telephony advisor for Exchange 2013. Configuration notes for supported VoIP gateways, IP PBXs, and PBXs provides VoIP gateway configuration notes and files that you must have to correctly configure your organization's VoIP gateways, IP PBXs, and SBCs to work with Exchange Unified Messaging.

Interoperability testing of Exchange Unified Messaging with VoIP gateways is now integrated with the Microsoft Unified Communications Open Interoperability Program. For more information, see Microsoft Unified Communications Open Interoperability Program.

The Microsoft Unified Communications Open Interoperability Program qualification program for VoIP gateways and IP PBXs ensures that customers have a seamless setup and support experience when they're using qualified telephony gateways and IP PBXs with Microsoft Unified Communications software.

Important:
Sending and receiving faxes using T.38 or G.711 isn't supported in an environment where Unified Messaging and Communications Server 2007 R2 or Microsoft Lync Server are integrated.

Deploying and configuring faxing

UM forwards incoming fax calls to a dedicated fax partner solution, which then establishes the fax call with the fax sender and receives the fax on behalf of the UM-enabled user. However, to allow

UM-enabled users to receive fax messages in their mailbox, you must configure the fax partner server, and then configure the UM dial plans, UM mailbox policies, and enable UM-enabled users to receive faxes. For details, see Setting up incoming faxing.

Setting up incoming faxing

Unified Messaging > Set up client voice mail features > Enable voice mail users to receive faxes >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

Microsoft Exchange Unified Messaging (UM) relies on certified fax partner solutions for enhanced fax features such as outbound fax or fax routing. By default, Exchange servers aren't configured to allow incoming faxes to be delivered to a user that's enabled for UM. Instead, an Exchange server redirects incoming fax calls to a certified fax partner solution. The fax partner's server receives the fax data and then sends it to the user's mailbox in an email message with the fax included as a .tif attachment.

For more information about fax partners, see Microsoft Pinpoint for Fax Partners

Deploying and configuring faxing

UM forwards incoming fax calls to a dedicated fax partner solution, which then establishes the fax call with the fax sender and receives the fax on behalf of the UM-enabled user. However, to allow UM-enabled users to receive fax messages in their mailboxes, you must first enable incoming faxing and set the fax partner's URI on the UM mailbox policy that's linked to the UM-enabled user or users. You can allow or prevent incoming faxing on UM dial plans, UM mailbox policies, and on the mailbox for a UM-enabled user. For details, see the following topics:

- Allow users in the same dial plan to receive faxes
- Prevent users in the same dial plan from receiving faxes
- Enable faxing for a group of users
- Disable faxing for a group of users
- Enable a user to receive faxes
- Prevent a user from receiving faxes

Step 1: Deploy Unified Messaging

Before you can set up faxing for your on-premises or hybrid organization, you need to successfully deploy Client Access and Mailbox servers and configure your supported Voice over IP (VoIP) gateways to allow faxing. For details about how to deploy UM, see Deploy Exchange 2013 UM. For

details about how to deploy VoIP gateways and IP Private Branch eXchanges (PBXs), see Connect UM to your telephone system.

Important:

Sending and receiving faxes using T.38 or G.711 isn't supported in an environment where Unified Messaging and Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server are integrated.

Step 2: Configure fax partner servers

Next, you need to enable incoming faxing and configure the fax partner's URI on each UM mailbox policy that you require in your organization. To successfully deploy incoming faxing, you must integrate a certified fax partner solution with Exchange Unified Messaging. For details, see Fax advisor for Exchange UM. For a list of certified fax partners, see Microsoft Pinpoint for Fax Partners

Note:

Because the fax partner server is external to your organization, firewall ports must be configured to allow the T.38 protocol ports that enable faxing over an IP-based network. By default, the T.38 protocol uses TCP port 6004. It can also use User Datagram Protocol (UDP) port 6044, but this will be defined by the hardware manufacturer. The firewall ports must be configured to allow fax data that uses the TCP or UDP ports or port ranges defined by the manufacturer.

Step 3: Enable faxing on Unified Messaging

Three components must be configured correctly for users to be able to receive faxes by using Unified Messaging:

- UM dial plans
- UM mailbox policies
- UM mailboxes

Faxing can be enabled or disabled on UM dial plans, UM mailbox policies, or on an individual UM-enabled user's mailbox. UM mailbox policies can be enabled or disabled for faxing using either the Exchange Administration Center (EAC) or the Exchange Management Shell. Enabling and disabling of dial plans and individual UM-enabled users needs to be done using the Exchange Management Shell. The following table shows the options that are available and the cmdlets and parameters that are used for enabling and disabling faxing.

UM component	Enable/disable using the EAC?	Shell example for enabling faxing
Dial plan	No	`Set-UMDialPlan -id MyUMDialPlan -faxenabled $true`
UM mailbox policy	Yes	`Set-UMMaiboxPolicy -id MyPolicy -AllowFax $true`

UM-enabled user	No	`Set-UMMailbox -id tonysmith -faxenabled $true`

By default, although the UM dial plan and the user's mailbox allow incoming faxes, you must first enable inbound faxing on the UM mailbox policy that's assigned to the UM-enabled user and then enter the fax partner server's URI.

To enable UM-enabled users to receive faxes, you must do the following:

- Verify that each UM dial plan allows the users who are associated with the dial plan to receive faxes. By default, all users who are associated with a dial plan can receive faxes. For UM-enabled users to receive fax messages in their mailbox, each VoIP gateway or IP PBX must be configured to accept incoming fax calls. You must also enable fax messages to be received by users who are linked with the dial plan. For more information about how to enable users linked with a dial plan to receive faxes or to prevent them from doing this, see Enable a user to receive faxes.

Note:
If you prevent fax messages from being received on a dial plan, no users who are associated with the dial plan will be able to receive faxes, even if you configure an individual user's properties to allow them to receive faxes. Enabling or disabling faxing on a UM dial plan takes precedence over the settings for an individual UM-enabled user.

- Configure the UM mailbox policy that's associated with the UM-enabled user. The UM mailbox policy must be configured to allow incoming faxes, including the fax partner's URI and the name of the fax partner's server. The *FaxServerURI* parameter must use the following form: sip:<*fax server URI*>:<*port*>;<*transport*>, where "fax server URI" is either a fully qualified domain name (FQDN) or an IP address of the fax partner server. The "port" is the port on which the fax server listens for incoming fax calls and "transport" is the transport protocol that's used for the incoming fax (UDP, TCP, or Transport Layer Security (TLS)). For example, you might configure a UM mailbox policy to receive a fax as follows.

```
Set-UMMailboxPolicy MyUMMailboxPolicy -AllowFax $true -
FaxServerURI "sip:faxserver.abc.com:5060;transport=tcp"
```

- For details, see Set the partner fax server URI to allow faxing.

Warning:
Although you can include multiple entries in the format for the *FaxServerURI* by separating them with a semicolon, only one entry will be used. This parameter allows only one entry to be used, and adding multiple entries won't enable you to load balance fax requests.

- Verify that the mailbox that's UM-enabled can receive fax messages. By default, all users who are associated with a dial plan can receive faxes. However, there may be situations when a user can't receive faxes because the ability to receive faxes has been disabled on their mailbox. For more information about how to enable a UM-enabled user to receive faxes, see Enable a user to receive faxes.

You can prevent an individual user who's associated with a dial plan from receiving fax messages. To do this, configure the properties for the user by using the **Set-UMMailbox** cmdlet in the Shell. You can also use the **Set-UMMailboxPolicy** cmdlet to prevent multiple users from receiving fax

messages. For more information about how to prevent a user or users from receiving fax messages, see Prevent a user from receiving faxes.

Step 4: Configure authentication

In addition to configuring your UM dial plans, UM mailbox policies, and UM-enabled users, you have to configure authentication between your Exchange servers and the fax partner server. The Exchange servers must be able to authenticate the origin of the messages that claim to be coming from the fax partner server. Any unauthenticated messages claiming to have come from a fax partner server won't be processed by an Exchange server.

To authenticate the connection from the fax partner server to the Exchange servers, you can use:

- Mutual TLS
- Sender ID validation
- A dedicated receive connector

A receive connector should be sufficient for authenticating the fax partner servers deployed in your organization. The receive connector will ensure that the Exchange servers treats all traffic coming from the fax partner server as authenticated.

The receive connector will be configured on an Exchange server that's used by the fax partner server to submit SMTP fax messages, and must be configured with the following values:

- *AuthMechanism: ExternalAuthoritative*
- *PermissionGroups: ExchangeServers, PartnersFax*
- *RemoteIPRanges: {the fax server's IP address}*
- *RequireTLS: False*
- *EnableAuthGSSAPI: False*
- *LiveCredentialEnabled: False*

For details, see Connectors.

If the fax partner server sends network traffic to an Exchange server over a public network, for example, a service-based fax partner server hosted in the cloud, it's a good idea to authenticate the fax partner server using a sender ID check. This type of authentication ensures that the IP address that the fax message came from is authorized to send email messages on behalf of the fax partner domain that the message claims to have come from. DNS is used to store the sender ID records (or sender policy framework (SPF) records) and fax partners must publish their SPF records in the DNS forward lookup zone. Exchange will validate the IP addresses by querying DNS. However, the sender ID agent must be running on a Mailbox server to be able to perform the DNS query.

You can also use TLS to encrypt the network traffic, or mutual TLS for encryption and authentication between the fax partner server and Exchange servers.

Faxing procedures

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-15*

Set the partner fax server URI to allow faxing

Applies to: *Exchange Online*

Topic Last Modified: *2013-02-22*

You can enable and disable inbound faxes for users associated with a Unified Messaging (UM) mailbox policy. By default, when you enable users for UM, users can't receive fax messages until you enable inbound faxing on the UM mailbox policy and specify the URI for the partner fax server. If the URIs are configured on the UM mailbox policy but the option to allow incoming faxes is disabled on the UM dial plan or for an individual user, UM-enabled users linked to the UM mailbox policy still won't be able to receive faxes.

For more information about fax partners, see Microsoft PinPoint for Fax Partners.

For additional management tasks related to faxing, see Faxing procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.

- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to set the fax partner URI

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
2. On the **UM dial plan** page, under **UM Mailbox Policies**, select the policy you want to modify, and then click **Edit** .
3. On the **UM mailbox policy** page > **General**, in the **Partner fax server URI** box, enter the TCP or TLS URI. For example: *sip:faxserver1.contoso.com:5060;transport=tcp* or *sip:faxserver2.contoso.com:5061;transport=tls*

Note:

Although the box can contain more than one fax server URI, only one will be used. If you enter two URIs, only the first will be used.

4. Click **Save** to save your changes.

Use the Shell to set the fax partner URI

This example allows users who are linked with the UM mailbox policy `UMDialPlan Default Policy` to use TCP with port 5060 for the partner fax server `faxserver1`.

```
Set-UMMailboxPolicy "UMDialPlan Default Policy" -
FaxServerURI sip:faxserver1.contoso.com:5060;transport=tcp
```

This example allows users who are linked with the UM mailbox policy `UMDialPlan Default Policy` to use TLS with port 5061 for the partner fax server `faxserver2`.

```
Set-UMMailboxPolicy "UMDialPlan Default Policy" -
FaxServerURI sip:faxserver2.contoso.com:5061;transport=tls
```

Include text with the email message sent when a fax message is received

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can include additional text in the email message that's sent when a fax message is received by a user who is enabled for Unified Messaging (UM) voice mail and is fax-enabled, and when the UM mailbox policy has been configured correctly to use a fax partner provider. By default, the text included when a UM-enabled user receives a fax message indicates only that the user has received a fax message. However, you can create a custom message by adding text in the **When a user receives a fax message** box on a UM mailbox policy. For example, the text can include information about system security policies and describe the correct way to handle fax messages in your organization. After you add the text, it will be included in each email message that's sent when UM-enabled users who are associated with the UM mailbox policy receive a fax message.

Note:

The custom text that accompanies a fax message is limited to 512 characters, and can include simple HTML text.

For more information about fax partners, see Microsoft PinPoint for Fax Partners.

For additional management tasks related to faxing, see Faxing procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the text included with a fax message

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **Message text**, in the text box for **When a user receives a fax message**, enter the text you want to include in the email message that's sent when users receive a fax message in their mailbox.
4. Click **Save**.

Use the Shell to change the text included with a fax message

This example enables UM-enabled users who are associated with a UM mailbox policy to receive additional instructions on how to open a fax message that they've received in their mailbox.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
FaxMessageText "To open this fax message, double-click the
file attachment."
```

Allow users in the same dial plan to receive faxes

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can enable all users who are linked with a Unified Messaging (UM) dial plan to receive fax messages in their mailboxes. By default, users who are enabled for Unified Messaging and are linked with a UM dial plan can receive fax messages. To allow UM-enabled users to receive fax messages in their mailboxes, the dial plan must be configured to accept incoming fax calls. You must also enable faxing on the UM mailbox policy and for the user. By default, faxing is enabled on

dial plans, UM mailbox policies, and for users. However, there may be times when these default settings have changed and UM-enabled users can't receive fax messages.

If you prevent fax messages from being received on a dial plan, all users who are associated with the dial plan won't be able to receive fax messages, even if you configure an individual user's properties to allow them to receive fax messages. Enabling or disabling faxing on a UM dial plan takes precedence over the settings for faxing on a UM mailbox policy or an individual UM-enabled user.

Note:

You can use the EAC to configure fax settings on a UM mailbox policy. However, you must use the Shell to configure fax settings on dial plans or for individual users.

For more information about fax partners, see Microsoft PinPoint for Fax Partners.

For additional management tasks related to faxing, see Faxing procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to allow users who are linked to a dial plan to receive faxes

This example enables UM-enabled users who are linked with the UM dial plan named `MyUMDialPlan` to receive incoming faxes.

```
Set-UMDialPlan -Identity MyUMDialPlan -FaxEnabled $true
```

Prevent users in the same dial plan from receiving faxes

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can prevent UM-enabled users who are linked with a Unified Messaging (UM) dial plan from receiving fax messages. By default, users who are enabled for Unified Messaging and are linked with a UM dial plan can receive fax messages. However, there may be times when you want to prevent users who are associated with a specific UM dial plan from receiving faxes.

You can prevent UM-enabled users from receiving faxes by configuring the UM dial plan, the UM mailbox policy, or the UM-enabled user's mailbox. If you disable incoming fax message delivery on a UM dial plan, all users who are associated with the dial plan will be prevented from receiving fax messages. Enabling or disabling faxing on a UM dial plan takes precedence over the settings for an individual UM-enabled user.

Note:

You can use the EAC to configure fax settings on a UM mailbox policy. However, you must use the Shell to configure fax settings on dial plans or for individual users.

For more information about fax partners, see Microsoft PinPoint for Fax Partners.

For additional management tasks related to faxing, see Faxing procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to prevent users who are linked to a dial plan from receiving faxes

This example prevents UM-enabled users associated with the UM dial plan named `MyUMDialPlan` from receiving faxes.

```
Set-UMDialPlan -Identity MyUMDialPlan -FaxEnabled $false
```

Enable faxing for a group of users

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-11*

You can enable inbound faxes for users linked with a Unified Messaging (UM) mailbox policy. By default, when you enable users for Unified Messaging, users can't receive fax messages until you specify the URI for the fax partner server, deploy a fax partner server for your organization, and enable faxing on a UM mailbox policy. If the option to allow incoming faxes is disabled on the UM dial plan, the users linked with the UM mailbox policy still won't be able to receive faxes. Similarly, if the option to allow incoming faxes is disabled on an individual user, that user won't be able to receive faxes.

For more information about fax partners, see Microsoft PinPoint for Fax Partners.

For additional management tasks related to faxing, see Faxing procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable inbound faxing

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then click **Edit** .

2. On the **UM dial plan** page, under **UM Mailbox Policies**, select the mailbox policy you want to modify, and then click **Edit** .
3. On the **UM mailbox policy** page > **General**, select the check box next to **Allow inbound faxes**.
4. Click **Save** to save your changes.

Use the Shell to enable inbound faxing

This example allows users who are linked with the UM mailbox policy `MyUMMailboxPolicy` to use inbound faxing.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -AllowFax
$true
```

Disable faxing for a group of users

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can disable inbound faxes for users associated with a Unified Messaging (UM) mailbox policy. By default, when you enable users for Unified Messaging, users can't receive fax messages until you specify the URI for the fax partner server , deploy a fax partner server for your organization, and enable faxing on a UM mailbox policy. If the option to allow incoming faxes is disabled on the UM dial plan, the users linked with the UM mailbox policy still won't be able to receive faxes. Similarly, if the option to allow incoming faxes is disabled on an individual user, that user won't be able to receive faxes.

For more information about fax partners, see Microsoft PinPoint for Fax Partners.

For additional management tasks related to faxing, see Faxing procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to disable inbound faxing

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
2. On the **UM dial plan** page, under **UM Mailbox Policies**, select the mailbox policy you want to modify, and then click **Edit** .
3. On the **UM mailbox policy** page > **General**, clear the check box next to **Allow inbound faxes**.
4. Click **Save** to save your changes.

Use the Shell to disable inbound faxing

This example prevents users who are linked with the UM mailbox policy `MyUMMailboxPolicy` from using inbound faxing.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -AllowFax $false
```

Enable a user to receive faxes

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can enable a Unified Messaging (UM) user to receive faxes. By default, when you enable a user for Unified Messaging, they will be able to receive faxes if you enable faxing and configure a fax partner's URI on the UM mailbox policy that is linked to the user. Faxing can be enabled or disabled on UM dial plans, UM mailbox policies, or the UM-enabled user's mailbox.

By default, the user's mailbox and the dial plan that is linked with the user allow incoming faxes. However, for a user to receive faxes you must first enable inbound faxing on the UM mailbox policy that's associated with the UM-enabled user and enter the fax partner's URI.

Note:

You can use the EAC to configure fax settings on a UM mailbox policy. However, you must use the Shell to configure fax settings on dial plans or for individual users.

For more information about fax partners, see Microsoft PinPoint for Fax Partners.

For additional management tasks related to faxing, see Faxing procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the UM mailbox policy assigned to the user has faxing enabled and the fax partner's URI is properly configured.
- Before you perform these procedures, confirm that the user is enabled for Unified Messaging. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to enable a UM user to receive faxes

This example enables Tony Smith to receive incoming faxes.

```
Set-UMMailbox -Identity tonysmith@contoso.com -FaxEnabled
$true
```

Prevent a user from receiving faxes

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can prevent a Unified Messaging (UM) user from receiving faxes. By default, when you enable a user for Unified Messaging, they will be able to receive faxes if you enable faxing and configure a fax partner's URI on the UM mailbox policy that is linked to the user. Faxing can be enabled or disabled on UM dial plans, UM mailbox policies, or the UM-enabled user's mailbox.

By default, the user's mailbox and the dial plan that is linked with the user allow incoming faxes. However, for a user to receive faxes you must first enable inbound faxing on the UM mailbox policy that's associated with the UM-enabled user and enter the fax partner's URI.

Note:

You can use the EAC to configure fax settings on a UM mailbox policy. However, you must use the Shell to configure fax settings on dial plans or for individual users.

For more information about fax partners, see Microsoft PinPoint for Fax Partners.

For additional management tasks related to faxing, see Faxing procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user is enabled for Unified Messaging. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to prevent UM-enabled user from receiving faxes

This example prevents a UM-enabled user named Tony from receiving fax messages in his mailbox.

```
Set-UMMailbox -Identity tony@contoso.com -FaxEnabled $false
```

Protect voice mail

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-16*

Some legacy Private Branch eXchange (PBX) and IP PBX telephony systems allow the caller to mark a voice mail message as private, blocking the intended recipient of the message from forwarding it to others. In integrated voice mail systems, a voice message can be accessed in multiple ways, which makes it more of a challenge to prevent voice messages marked private from being exposed to unintended listeners.

Unified Messaging (UM) can be configured to use Active Directory Rights Management Services (AD RMS) to protect voice messages for an organization. This feature is known as Protected Voice Mail.

When a voice message is protected, the recipient is not only blocked from forwarding the message, but UM also ensures that only the intended recipient or recipients of the message can access its content. Protected voice messages can be accessed by using Microsoft Outlook 2010 or later, Outlook Web App, or Outlook Voice Access.

Contents

Overview of Protected Voice Mail

The Protected Voice Mail feature is available with Exchange 2010 and later versions of Unified Messaging (UM). It can be configured on a UM mailbox policy, and all Protected Voice Mail settings can be configured by using the Exchange Management Console or the Shell in Exchange 2010 or by using the Exchange admin center (EAC) or cmdlets in the Shell in Exchange 2013.

Protected Voice Mail is implemented by applying Information Rights Management (IRM) to voice messages. When voice messages are protected by UM:

- Users can reply to protected voice messages.
- Recipients of a voice message can't forward it.
- Users can't save a copy of the voice message.
- Users can't save or copy the attached audio of the voice message.
- A voice message can be opened only by the intended recipient or recipients.

Both call-answering voice messages and interpersonal voice messages (voice messages that are

sent to a user using Outlook Voice Access) can be protected by UM. However, protection won't be applied to the following types of messages:

- Fax messages.
- Non-voice messages. For example, email messages or meeting requests, even when they're created using Outlook Voice Access (voice replies).

Return to top

Overview of Active Directory Rights Management Services

AD RMS, a component of Windows Server 2008 and later versions, is available to help protect files so that only the users who the sender intends to view a file can do so. AD RMS protects a file by specifying the rights that a user must have to access the file. Rights can be configured to allow a user to open, modify, print, forward, or take other actions with the rights-managed information. With AD RMS, you can safeguard data when it's distributed outside your network.

An AD RMS system has both a server and a client component, including the following:

- A server that has Windows Server 2008 R2 or a later version installed that’s running the Active Directory Rights Management Services server role, which handles certificates and licensing.
- A database server.
- The AD RMS client. The latest version of the AD RMS client is included as part of the Windows 7 and Windows 8 operating systems.

The server component is made up of several web services that run on a Microsoft server such as Windows Server 2008 or a later version. The client component can be run on either a client or server operating system and includes functions that enable an application to encrypt and decrypt content, retrieve templates and revocation lists, and acquire licenses and certificates from a server.

By using AD RMS and the AD RMS client, you can augment an organization's security strategy by protecting information through persistent usage policies that remain with the information, regardless of where it's moved. You can use AD RMS to help prevent sensitive information—such as financial reports, product specifications, customer data, and confidential email and voice mail messages—from intentionally or accidentally getting into the wrong hands. For detailed information, see AD RMS Overview.

In Exchange UM you can use Information Rights Management (IRM) features to apply persistent protection to messages and attachments.

Using the IRM features and Protected Voice Mail, your organization and your users can control the rights recipients have to access email and voice mail messages. IRM can be also used to restrict recipient actions such as forwarding a message to other recipients, printing a message or attachment, or extracting message or attachment content by copying and pasting.

IRM requirements

Before you can implement IRM in Exchange, you must first deploy and configure your AD RMS

infrastructure. For detailed information, see Active Directory Rights Management Services. To implement IRM to support Protected Voice Mail in your Exchange organization, your deployment must meet the following requirements.

Server	Requirement
AD RMS Cluster	• Windows Server 2008 R2 Standard or Enterprise with SP1 or Windows Server 2012 Standard or Datacenter. For more information about system requirements, see Exchange 2013 system requirements. • **Service connection point (SCP)** Exchange 2013 and AD RMS-aware applications use the SCP registered in Active Directory to discover AD RMS clusters and URLs. AD RMS allows you to register the SCP within AD RMS setup. If the account used to set up AD RMS isn't a member of the Enterprise Admins security group, SCP registration can be performed after setup. There is only one SCP for AD RMS in an Active Directory forest. • **Permissions** Servers in the Exchange servers group or individual Exchange servers must be assigned Read and Execute permissions to the AD RMS server certification pipeline. The default path is \inetpub\wwwroot_wmcs \certification\ServerCertification.asmx on AD RMS servers. • **AD RMS super users** To enable transport decryption, journal report decryption, IRM in Outlook Web App, and IRM for Exchange Search, you must add the Federated Delivery mailbox, a system mailbox created by Exchange Setup, to the AD RMS super users group on the AD RMS cluster. For detailed information, see Add the Federation Mailbox to the AD RMS

	Super Users Group.

Configuring and testing IRM

You must use the Shell to configure IRM features. To configure individual IRM features, use the Set-IRMConfiguration cmdlet. For more information about how to configure IRM features, see Information Rights Management procedures.

After you've set up an Exchange server, you can use the Test-IRMConfiguration cmdlet to perform end-to-end tests of your IRM deployment. This cmdlet verifies the IRM configuration for an organization and should be run before enabling Protected Voice Mail. The **Test-IRMConfiguration** cmdlet performs the following tests:

- Inspects the IRM configuration for your Exchange organization.
- Checks the AD RMS server for version and hotfix information.
- Verifies whether an Exchange server can be activated for RMS by retrieving a Rights Account Certificate and Client Licensor Certificate (CLC).
- Acquires AD RMS rights policy templates from the AD RMS server.
- Verifies that the specified sender can send IRM-protected messages.
- Retrieves a super user use license for the specified recipient.
- Acquires a pre-license for the specified recipient.

Return to top

Client support and end-user features

The email client software that's used to listen to a Protected Voice Mail message must support IRM and know how to read a UM-protected voice message. Email clients that are supported include Microsoft Outlook 2010 or later versions, Outlook Web App, and Outlook Voice Access. The following table contains a list of email clients and whether they're supported.

Email client	Description
Outlook	• Protected voice messages are supported in Outlook 2010 and later versions.
Outlook Web App	• Outlook Web App in Exchange 2010 or later versions supports Protected Voice Mail messages. Earlier versions of Outlook Web App, known as Outlook Web Access, don't support them.
Outlook Voice Access	• Outlook Voice Access in Exchange 2010 and later versions supports Protected Voice Mail. Outlook Voice Access included with Exchange

	2007 doesn't support Protected Voice Mail. • The user's mailbox must reside on a Mailbox server in Exchange 2010 or a later version.
Windows Mobile or Windows Phone	• Windows Mobile doesn't support Protected Voice Mail. However, Windows Phone 7 and Windows Phone 8 support Protected Voice Mail.
Exchange ActiveSync	• Protected Voice Mail is supported in Exchange 2010 SP1 and later versions.
Other email clients	• Protected Voice Mail isn't supported.

Return to top

Protected voice message structure

There are actually two messages involved for each Protected Voice Mail message. The first message is the outer message, which isn't encrypted. It contains an attachment named message.rpmsg. The attachment contains the IRM-protected voice message and internal rights management control data. The rights management control data includes a content key and rights information that specifies who can access the voice message and how those users can access it.

Protected voice messages are shown in the user's Inbox in the **Voice Mail** search folder. The user can listen to the voice messages by using the embedded audio player just as they would listen to a regular voice message, except that the Forward button will be disabled and a note will be shown at the top of the message stating that it's protected and that it can't be forwarded.

For email clients that don't support Protected Voice Mail, the body of the outer message will be displayed. Administrators can include text when the client's software doesn't support Protected Voice Mail by using UM mailbox policies. You can customize the default text that's included in the email message by configuring a UM mailbox policy. For example, you could configure the UM mailbox policy with customized text such as, "*You can't open this voice mail message because it's protected. To view or listen to this voice message, sign in to your mailbox at https://mail.contoso.com or call +1 (425) 555-1234 to call in to Outlook Voice Access.*"

Return to top

Composing a Protected Voice Mail message

There are two situations in which protected voice messages can be created:

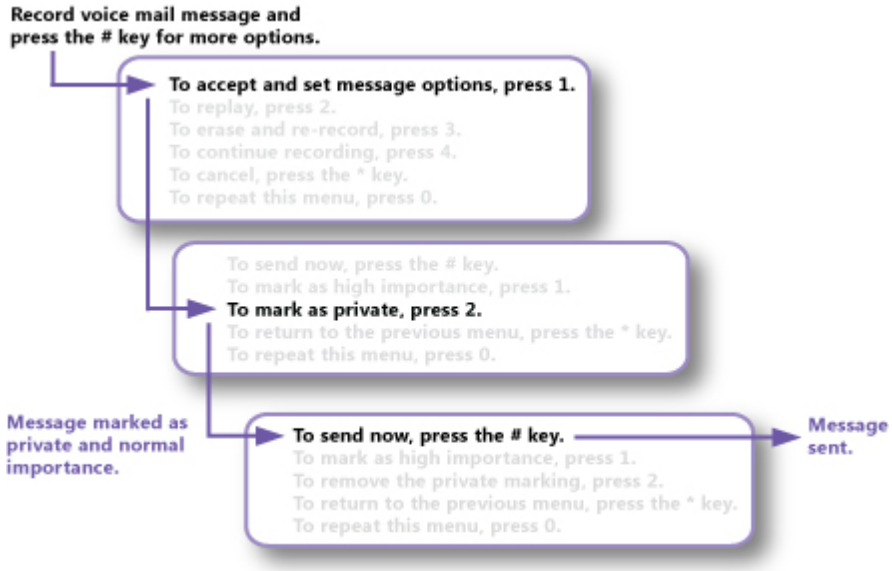
- **Call answering** Call answering occurs when a caller calls a UM-enabled user, but the user isn't available to answer the call or forwards it directly to voice mail. In call-answering scenarios, the

voice mail system will play a series of voice prompts after the caller records a voice message. The caller can then choose from additional message options, including the option to mark the voice message as private by pressing the pound (#) key. If the caller presses the # key, they can follow the instructions provided by UM to mark the message as private, remove the private marking from the private voice message, or mark the voice message with High importance. The following diagram shows the menu options that are available to callers when they leave a private voice message for a user.

Note:

For call-answering calls, UM uses the Protected Voice Mail settings on the UM mailbox policy of the intended recipient of the message, because the caller isn't authenticated.

Create a Protected Voice Mail message using Call Answering



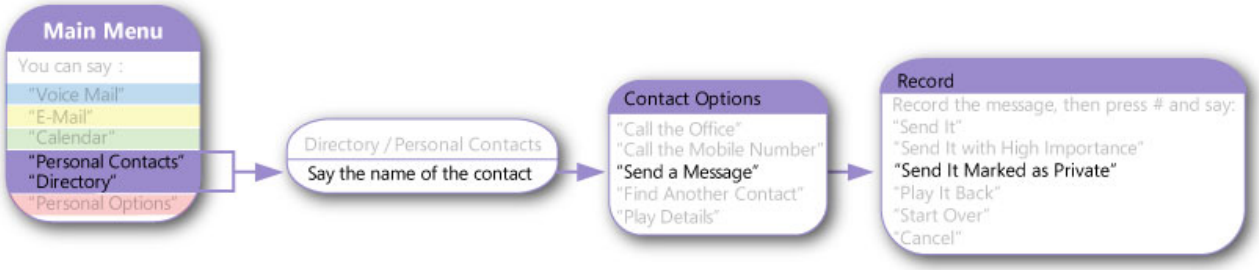
- **Outlook Voice Access** Outlook Voice Access lets UM-enabled users access their mailbox using analog, digital, or cellular telephones by dialing their Outlook Voice Access number. There are two Unified Messaging user interfaces available to UM-enabled users: the telephone user interface (TUI) and the voice user interface (VUI).

Outlook Voice Access users can search for contacts in the directory and send them voice messages. If Protected Voice Mail has been enabled for the UM-enabled recipients, callers can mark the messages as private after they're recorded. Alternatively, administrators can configure a UM mailbox policy to ensure that all voice messages sent by authenticated users are protected by UM.

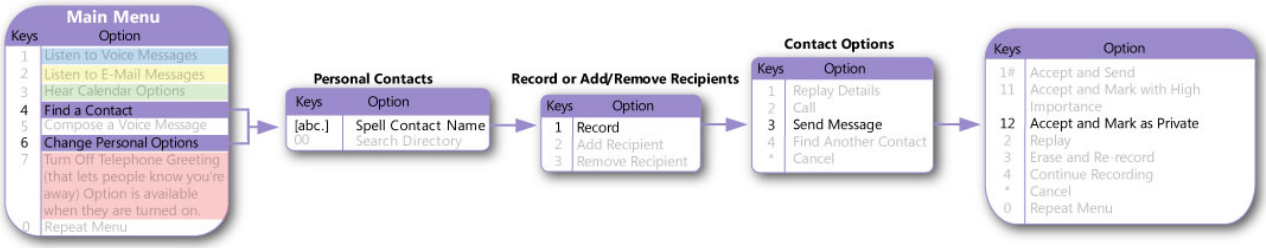
Note:

If a caller is authenticated, the Protected Voice Mail settings on the UM mailbox policy that's linked to the caller are applied, regardless of the UM mailbox policy settings for the intended recipient of the voice message.

Create a Protected Voice Mail message using the voice user interface



Create a Protected Voice Mail message using the telephone user interface



Return to top

UM mailbox policies

You can create a Unified Messaging mailbox policy to apply a common set of UM policy settings, such as PIN policy settings, dialing restrictions, and Protected Voice Mail settings, to a collection of UM-enabled mailboxes. To learn more about UM mailbox policies, see Manage a UM mailbox policy.

You can use the EAC or the **Set-UMMailboxPolicy** cmdlet in the Shell to configure Protected Voice Mail options. The following table lists the settings that can be configured for Protected Voice Mail.

Protected Voice Mail settings

Shell parameter	Setting available in EAC?	Description
ProtectAuthenticatedVoiceMail	Yes	The *ProtectAuthenticatedVoiceMail* parameter specifies whether UM-enabled users can send protected voice messages when they're accessing their mailbox using Outlook Voice Access. The default setting is None. This means that no protection is applied when voice messages are composed and that callers won't have the

		option to mark voice messages as Private. If the value is set to `Private`, only messages marked as Private by the caller are protected. If the value is set to `All`, every voice message is protected, regardless of the option chosen by the caller.
ProtectUnauthenticatedVoiceMail	Yes	The *ProtectUnauthenticatedVoiceMail* parameter specifies whether the Mailbox servers that answer calls for UM-enabled users associated with a UM mailbox policy create protected voice messages. This setting also applies when a message is sent from a UM auto attendant to a UM-enabled user. The default setting is `None`. This means that no protection is applied to voice messages and that the caller won't be offered the option to mark the message as Private. If the value is set to `Private`, only messages marked as Private by the caller are protected. If the value is set to `All`, every voice message is protected, regardless of whether if the message has been marked as

		private by the caller.
ProtectedVoiceMailText	Yes	The *ProtectedVoiceMailText* parameter specifies the text to be included in the body of the outer message of a Protected Voice Mail message. This text will be shown in all email client applications that don't support Protected Voice Mail messages. Note that a default message is always provided by UM when this property is set to `Null` or is empty.
RequireProtectedPlayOnPhone	Yes	The *RequireProtectedPlayOnPhone* parameter specifies whether users associated with the UM mailbox policy will be forced to listen to the protected voice message over the phone (using Play On Phone). The default value is `$false`. When the value is set to `$true`, the audio media player on Protected Voice Mail forms in Outlook or Outlook Web App will be shown as disabled. Note that the preview text for the voice message can always be accessed. The user can't play the audio file using any media player software or use the

		embedded media player to listen to the voice message.
AllowVoiceResponseToOtherMessageTypes	Yes	The *AllowVoiceResponseToOtherMessageTypes* parameter specifies whether callers who have authenticated to Outlook Voice Access to access their email will be able to compose a voice reply to email messages and meeting requests.

For more information about how to manage Protected Voice Mail settings, see Protected Voice Mail procedures or Set-UMMailboxPolicy.

Return to top

Text message notifications and Protected Voice Mail

Users who configure their UM account to send text message notifications (also called SMS notifications) to their mobile phone when voice messages are received will also receive audio transcription (Voice Mail Preview) text as part of the body of the text message. However, for protected voice messages, this represents a security issue because the content of the voice messages should always be protected.

When UM creates a text message notification for a voice message that's protected, it checks whether the voice message is marked as Private. If so, it won't add the transcribed audio text to the text message that it sends to the mobile phone. The following text will be included in the text message instead: "Use Outlook Voice Access to access this protected voice mail message."

Return to top

Protected Voice Mail procedures

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-01-30*

Configure Protected Voice Mail from authenticated callers

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can configure Unified Messaging to answer an incoming call, and then determine whether it will apply protection to voice mail messages by using encryption. When a voice message is protected:

- The message is marked as Private in Microsoft Outlook and Outlook Web App.
- The voice message can be opened only by the intended recipient of the voice message.
- The recipient can reply to the voice message, but can't forward it to someone who wasn't included on the original voice message.

This setting applies to voice messages sent to UM-enabled users when they don't answer their phone. This setting also applies when callers sign in to their mailbox using Outlook Voice Access, and then create and send a voice message.

For additional management tasks related to Protected Voice Mail procedures, see Protected Voice Mail procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure Protected Voice Mail from authenticated callers

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **Protected voice mail**, under **Protect voice message from authenticated callers**, select one of the following options:
   - **None** Use this setting when you don't want protection applied to any voice messages sent to UM-enabled users.
   - **Private** Use this setting when you want Unified Messaging to apply protection only to voice messages that have been marked as private by the caller.
   - **All** Use this setting when you want Unified Messaging to apply protection to all voice messages, including those not marked as private.
4. Click **Save**.

Use the Shell to configure Protected Voice Mail from authenticated callers

This example protects voice messages from all authenticated callers on the UM mailbox policy `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy
ProtectAuthenticatedVoiceMail -All
```

Configure Protected Voice Mail from unauthenticated callers

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can configure Unified Messaging to answer an incoming call, and then determine whether it will apply protection to voice mail messages by using encryption. When a voice mail message is protected:

- The message is marked as Private in Microsoft Outlook and Outlook Web App.
- The voice message can be opened only by the intended recipient of the voice message.
- The recipient can reply to the voice message, but can't forward it to someone who wasn't included on the original voice message.

This setting applies to voice messages sent to UM-enabled users when they don't answer their phone. This setting also applies to voice messages sent directly to UM-enabled users when the caller uses a UM auto attendant.

For additional management tasks related to Protected Voice Mail procedures, see Protected Voice Mail procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure Protected Voice Mail from unauthenticated callers

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want

to manage, and then click **Edit** .

3. On the **UM Mailbox Policy** page > **Protected voice mail**, under **Protect voice message from unauthenticated callers**, select one of the following options:
   - **None** Use this setting when you don't want protection applied to any voice messages sent to UM-enabled users.
   - **Private** Use this setting when you want Unified Messaging to apply protection only to voice messages that have been marked as private by the caller.
   - **All** Use this setting when you want Unified Messaging to apply protection to all voice messages, including those not marked as private.
4. Click **Save**.

Use the Shell to configure Protected Voice Mail from unauthenticated callers

This example protects all voice messages from all unauthenticated callers on the UM mailbox policy `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
ProtectUnauthenticatedVoiceMail -All
```

Enable or disable multimedia playback of protected voice messages

Set up client voice mail features > Protect voice mail > Protected Voice Mail procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can force users who receive protected voice mail messages to use the Play on Phone feature to listen to their messages. Or, if the client software doesn't support rights management, users must use Outlook Voice Access to listen to messages.

To listen to voice messages, Unified Messaging (UM)-enabled users can use the Play on Phone feature or use multimedia software on a computer or mobile device. Multimedia playback allows a UM-enabled user to use a media player over computer speakers or use a media player on a mobile device to hear the voice message.

Note:
Protected voice mail is available only on clients that are using a version of Outlook that supports rights management. If the client software doesn't support rights management, users

must use Outlook Voice Access to listen to their calls.

By default, the value of the **RequireProtectedPlayOnPhone** property on a UM mailbox policy is set to false. This means that UM-enabled users that are associated with that UM mailbox policy can listen to protected voice messages by:

- Using Outlook Voice Access.
- Using the built-in media player or the Play on Phone button in Outlook 2010 or a later version.
- Using the built-in media player or the Play on Phone button in Outlook Web App.

If this value is set to true, multimedia playback of protected voice mail isn't allowed. UM-enabled users associated with a UM mailbox policy on which this value is set to true can listen to protected voice messages only by:

- Using Outlook Voice Access.
- Using the Play on Phone button in Outlook 2010 or a later version.
- Using the Play on Phone button in Outlook Web App.

This setting is especially useful when UM-enabled users use public computers, laptops in public places, or their mobile device's media player to listen to protected voice mail that can contain private information.

For additional management tasks related to Protected Voice Mail procedures, see Protected Voice Mail procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable or disable multimedia playback of

protected voice messages

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. Under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **Protected voice mail**, select the check box next to **Require Play on Phone for protected voice messages** to enable this setting. Clear the check box to disable this setting.
4. Click **Save**.

Use the Shell to enable or disable multimedia playback of protected voice messages

This example allows users who are associated with the UM mailbox policy named `MyUMMailboxPolicy` to play back protected voice messages using a media player.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
RequireProtectedPlayOnPhone $false
```

This example prevents users who are associated with the UM mailbox policy named `MyUMMailboxPolicy` from playing back protected voice messages using a media player.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
RequireProtectedPlayOnPhone $true
```

Specify the text to display for email clients that don't support Windows Rights Management

Set up client voice mail features > Protect voice mail > Protected Voice Mail procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can specify the text that will be sent to a user when they receive a protected voice message but their email client doesn't support Information Rights Management (IRM) or Windows Rights Management.

Protected Voice Mail can be accessed only by email clients that support Windows Rights Management or when a UM–enabled user uses Outlook Voice Access to access a protected voice message.

Protected Voice Mail is encrypted. When a voice message is protected:

- The message is marked as Private in Microsoft Outlook and Outlook Web App.
- The voice message can be opened only by the intended recipient of the voice message.
- The recipient can reply to the voice message, but can't forward it to someone who wasn't included on the original voice message.

If a protected voice message is sent to someone whose email client doesn't support Windows Rights Management and isn't accessing the message using Outlook Voice Access, an email message will be sent to them that includes the text you specify. This text should include instructions about what the called party should do to be able to receive the protected voice message.

For additional management tasks related to Protected Voice Mail procedures, see Protected Voice Mail procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use EAC to specify the text to display for email clients that don't support Windows Rights Management

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** ✎.
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** ✎.

3. On the **UM Mailbox Policy** page > **Protected voice mail**, under **Message to send to users who don’t have Windows Rights Management support**, type the message text in the text box.
4. Click **Save**.

Use the Shell to specify the text to display for email clients that don't support Windows Rights Management

This example specifies the text to display to users associated with the UM mailbox policy named `MyUMMailboxPolicy` who have email clients that don't support Windows Rights Management.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
ProtectedVoiceMailText "Your email client software does not
support Protected Voice Mail. Please contact the Help
Desk."
```

Allow Message Waiting Indicator

Exchange Server 2013 > Unified Messaging > Set up client voice mail features >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-25*

Message Waiting Indicator (MWI) is a feature that's found in most legacy voice mail systems. It lets users know that they have new or unheard voice mail messages. In its most common form, this feature lights a lamp on a user's phone to indicate the presence of a new or unheard voice message.

Contents

Overview

MWI notifications can include any mechanism that indicates the existence of a new or unheard

voice message. The message can be in a new email message or one that's marked as unread. The MWI notification might take any of the following forms:

- A new voice message seen from Microsoft Outlook or Outlook Web App.
- A text or Short Messaging Service (SMS) message sent to a mobile phone that's configured to receive text messages.
- An outbound call made from Exchange Unified Messaging (UM).
- A lamp on a phone.
- A special dial tone.
- Icons or buttons on the display screen of a phone.
- A highlighted notification within a software application.

In Unified Messaging, a user's voice mail is stored in their mailbox. It can be accessed from a telephone using Outlook Voice Access, from a desktop or portable computer using Outlook or Outlook Web App, and from mobile phone clients. When a user receives a new voice message, the message appears in their Voice Mail search folder. If the voice message is accessed using Outlook or Outlook Web App, an email message will be included with the voice message.

When previous versions of UM were deployed in an organization, MWI was supported in either a traditional VoIP gateway environment or an IP PBX environment by using a third-party solution or application, or it was included as part of Exchange UM. In Exchange 2010 or later, MWI is also supported when UM is deployed with Microsoft Lync Server. The MWI notification mechanism used by Lync Server depends on the type of VoIP-based phone that's used by the Enterprise Voice and UM-enabled user, and can be located on any of the following:

- A phone
- A phone display
- A dial pad button

By default, MWI is turned on for all users who are enabled for UM. It's controlled through settings on a UM mailbox policy or on the UM IP gateways that have been created and linked to a UM dial plan. MWI also works with protected voice messages.

To implement MWI in a traditional telephony environment with VoIP gateways, IP PBXs, or SIP-enabled PBXs, a Mailbox server running the Microsoft Exchange Unified Messaging service sends an MWI Session Initiation Protocol (SIP) NOTIFY message to a *SIP peer*, which is either an IP PBX, a VoIP gateway used with a legacy PBX, a SIP-enabled PBX, or if you have deployed Lync Server, Lync servers are also considered SIP peers. The IP PBX or PBX lights the lamp on the desktop phone to notify the user of a new or unheard voice message.

There are two ways callers can leave voice messages: using call answering and Outlook Voice Access. With call answering, a Mailbox server answers an incoming call and allows the caller to leave a voice message for a user. With Outlook Voice Access, when a caller calls an Outlook Voice Access number, they can use the menu system to leave a voice message for a UM-enabled user.

Return to top

The Mailbox server's role in MWI

When a caller calls a UM-enabled user and the user doesn't answer their phone, the Microsoft Exchange Unified Messaging service on a Mailbox server receives MWI state change information and uses a SIP NOTIFY message to send the request for a change of notification to a VoIP gateway, IP PBX, or SIP-enabled PBX. The change of notification includes the following information:

- Message Waiting Indicator enabled (Yes or No).
- Number of new voice messages or voice messages marked unheard.
- Number of old voice messages or voice messages marked heard.
- Number of new urgent voice messages or urgent voice messages marked unheard.
- Number of old urgent voice messages or urgent voice messages marked heard.
- The primary extension number on the primary UM dial plan.
- The IP address or fully qualified domain name (FQDN) of the SIP peer to be used for SIP NOTIFY messages.
- The security type of the UM dial plan (Unsecured, SIP secured, or Secured). This information will be used to determine whether the connection to the VoIP gateway or IP PBX must be SIP over TCP or SIP over TLS. Transport Layer Security (TLS) is supported for MWI SIP NOTIFY.

A Mailbox server uses the diversion information on the header of the incoming call to determine the extension number or phone number of the UM-enabled user. When the extension or phone number is determined, the Mailbox server sends the request to the SIP peer. The SIP peer then changes the MWI state and turns on the notification on the user's phone.

Note:

Although PBX outages should be rare, UM automatically refreshes the MWI state for every mailbox at least once every 12 hours. There is no way to force a refresh, but if the PBX or IP PBX is powered off and all the MWI lamps go off, all lamps should be restored to the correct state within six hours.

Return to top

MWI SIP NOTIFY messages

MWI notifications use SIP NOTIFY messages to communicate with SIP peers. MWI state change information is included in the SIP NOTIFY message and indicates whether MWI notifications will be sent to users. Whenever there's a change in the MWI state, the Mailbox Assistant sends this information to the Microsoft Exchange Unified Messaging service running on a Mailbox server. After the Unified Messaging service receives this information, it parses the message to obtain the target SIP peer and the MWI state change information. It then forms a SIP NOTIFY message with the MWI state change information in the message body and sends this information on to the SIP peer.

MWI is based on RFC 3842. RFC 3842 states that SIP event notifications must be used for message-waiting notifications. MWI is based on the SIP model, and is driven by the endpoints found in a unified messaging system. SIP endpoints, also called SIP peers, that obtain MWI information must

send a SIP SUBSCRIBE message to the Unified Messaging service. The service replies with a NOTIFY message indicating that the subscription has been accepted. All MWI state change information will be conveyed from the Unified Messaging system to a SIP endpoint using NOTIFY messages that are embedded within the subscription that was previously created. The exact syntax of the SIP NOTIFY message to be sent to the SIP peer. is based on the format described in RFC 3842.

When the Unified Messaging service on a Mailbox server sends an MWI NOTIFY message to the SIP peer, the event header is set to “message-summary”, which indicates that this is an MWI-related NOTIFY message. The To header field indicates the SIP endpoint for which the MWI service must be provided. The Subscription-State header must be set to “terminated” instead of “active”. The following actions occur in response to the SIP NOTIFY message:

1. The SIP peer conveys this information to the PBX using a circuit-switched protocol such as SMDI.
2. The PBX sends a success message over a circuit-switched protocol.
3. The SIP peer can respond with either of the following messages: 200 OK (Success) or 480 (Temporarily Unavailable). A Mailbox server can handle both these responses and additional failure responses.

MWI in a traditional telephony environment

In a traditional telephony environment, an incoming call is received by the PBX and then sent on to the VoIP gateway, or is received by the IP PBX or SIP-enabled PBX. The Mailbox server and the Mailbox Assistant are used to determine the MWI state for the UM-enabled user. They're responsible for delivering the SIP notification back to the VoIP gateway and legacy PBX, IP PBX, or SIP-enabled PBX.

In a traditional telephony environment, the call flow and MWI notification are as follows:

1. The incoming call is received by the legacy PBX, forwarded to the VoIP gateway, or to an IP PBX or SIP-enabled PBX, and then forwarded to the extension number of the UM-enabled user. The user doesn't answer and the caller is prompted to leave a voice message.
2. The VoIP gateway or IP PBX submits the call to a Mailbox server running the Microsoft Exchange Unified Messaging service.
3. The Mailbox server performs an LDAP query to locate the UM-enabled user's information, such as the user’s extension number and personal greetings. The greetings for the UM-enabled user are played.
4. The voice message that was created by the Unified Messaging service is submitted to the Microsoft Exchange Transport service on the same Mailbox server.
5. The Transport service on the Mailbox server submits the voice message to the Mailbox server that contains the user’s mailbox. The Mailbox Assistant receives a MAPI event for a new voice message.
6. The Mailbox Assistant reads the UM dial plan and the UM mailbox policy to determine whether MWI notifications should be sent to the UM-enabled user. The Mailbox Assistant queries for all Mailbox servers that are associated with the UM dial plan of the UM-enabled user. The Mailbox Assistant tries to send the RPC event to the first Mailbox server that's returned. If this fails, it tries

the next one. It will keep retrying for five minutes, or until all Mailbox servers have been tried. If all the RPC calls fail, the Mailbox Assistant logs the error in Event Viewer. The Mailbox server queries for all UM IP gateways that are associated with the UM dial plan of the UM-enabled user's mailbox.

7. UM sends a SIP NOTIFY message to the first VoIP gateway or IP PBX that's returned from the query. If this fails, the Mailbox server will choose the next VoIP gateway or IP PBX. The Mailbox server will keep trying for a VoIP gateway or IP PBX for five minutes. If all attempts to find a VoIP gateway or IP PBX fail, the Mailbox server will log an error. If a VoIP gateway or IP PBX is located successfully, the VoIP gateway will send the notification to the PBX, and the PBX in turn will send a notification of the MWI event to the user's phone to light the phone lamp. If an IP PBX is used, the MWI notification is processed by the IP PBX and it will then light the user's phone lamp. Other MWI notification mechanisms are listed in the Overview section. The type of notification depends on the PBX or IP PBX hardware vendor and whether Lync Server is being used. It also depends on the type of phone—analog, digital, or VoIP—that's being used.

Return to top

MWI in a Lync Server environment

In telephony environments that include Lync Server, an incoming call can be sent from an external phone to a Mediation Server, sent from a Lync client, or sent from a unified communications (UC) or other VoIP-based phone. After the call is received, it's sent to the Lync Server front-end server pool. The Mailbox server and the Mailbox Assistant are used to determine the MWI state for the UM-enabled user and to deliver this notification to the Lync client, analog or digital or UC or VoIP-based phone.

In a telephony environment with Lync Server, the call flow and MWI notification are as follows:

1. The call is sent from one of the following:
   - A phone external to the organization (sent to a Mediation Server)
   - A Lync-based client
   - A UC or other VoIP-based phone
2. The incoming call is received by the Lync Server front-end server pool and sent on to the phone or SIP endpoint of the UM-enabled user. The user doesn't answer and the caller is prompted to leave a voice message.
3. The Lync Server front-end server pool submits the call to a Mailbox server within the on-premises network and the user's mailbox is found.
4. The Mailbox server performs an LDAP query to locate information for the UM-enabled user, such as their extension number and greetings. The greetings are played, and the caller is prompted to leave a voice message.
5. The voice message that was created by the Unified Messaging service is submitted to the Transport service on the same Mailbox server within the same site.
6. The Transport service submits the voice message to the Mailbox server that contains the user's mailbox. The Mailbox Assistant receives a MAPI event for the new voice message.

7. The Mailbox Assistant reads the UM dial plan and the UM mailbox policy to decide whether MWI notifications should be sent to the user. The Mailbox Assistant queries for all Mailbox servers that are associated with UM dial plan of the user. The Mailbox Assistant tries to send the RPC event to the first Mailbox server that's returned. If this attempt fails, the Mailbox Assistant tries the next one. It will keep trying to find a Mailbox server for five minutes or until all servers have been tried. If all the RPC calls fail, the Mailbox Assistant will log an error in the Event Viewer. The Mailbox server queries Active Directory for all UM IP gateways that are associated with the UM dial plan of the UM-enabled user.
8. UM sends a SIP NOTIFY message to the first VoIP gateway or IP PBX that's returned from the query. If this fails, the Mailbox server will choose the next VoIP gateway or IP PBX. The Mailbox server will keep trying to find a VoIP gateway or IP PBX for five minutes. If all attempts to contact the VoIP gateway or IP PBX fail, the Mailbox server will log an error. If it's successful, the VoIP gateway or IP PBX will send the notification to the Lync Server front-end server pool, which in turn will send a notification of the MWI event to a SIP endpoint used by the user, the user's phone, or a Lync client.

Return to top

MWI resilience

When you're deploying Mailbox and Client Access servers, UM dial plans, and UM IP gateways, and you’re using MWI for UM-enabled users, it's best to deploy multiple Mailbox and Client Access servers as well as multiple VoIP gateways and IP PBXs to create fault tolerance and resilience. Doing this also creates MWI resilience because it provides multiple ways to send MWI notifications, as described in the preceding section.

To enable fault tolerance for MWI in Unified Messaging, you must create and configure one or more of the following:

- A UM dial plan that's linked with the UM-enabled user who will receive MWI notifications.
- A UM mailbox policy that's linked with the UM-enabled user who will receive MWI notifications.
- A UM IP gateway that's linked with the UM dial plan that's linked with the UM-enabled user who will receive MWI notifications.
- If Lync Server is used, all the Client Access and Mailbox servers must be added to the UM SIP URI dial plan that's linked with the UM-enabled user who will receive MWI notifications.

MWI administration

MWI can be administered by configuring settings on two UM components: UM mailbox policies and UM IP gateways. For both UM components, you can enable or disable MWI notifications by using the **Set-UMMailboxPolicy** cmdlet or the **Set-UMIPgateway** cmdlet in the Exchange Management Shell. You can also configure the settings by using the Exchange Admin Center (EAC). You can view the status of MWI notifications by using the **Get-UMMailboxPolicy** cmdlet and the **Get-UMIPgateway** cmdlet in the Shell, or by viewing the settings in the EAC.

UM mailbox policies and MWI

You can create a UM mailbox policy to apply a common set of UM policy settings to a collection of UM-enabled mailboxes. For example, you can use a UM mailbox policy to apply PIN policy settings, dialing restrictions, and MWI notifications settings. If you enable or disable MWI on a UM mailbox policy, it will be enabled or disabled for all UM-enabled users who are linked with that UM mailbox policy. The MWI setting can also apply to a subset of the users who are linked with a UM dial plan. To learn more about UM mailbox policies, including how to enable or disable MWI for a group of UM-enabled users, see UM mailbox policy procedures.

You can use the EAC or the **Set-UMMailboxPolicy** cmdlet in the Shell to configure the MWI setting, as shown in the following table.

Message Waiting Indicator setting on a UM mailbox policy

Shell parameter	Setting available in the EAC?	Description
AllowMessageWaitingIndicator	Yes	The *AllowMessageWaitingIndicator* parameter specifies whether users who are linked with a UM mailbox policy can receive MWI notifications when they receive a new voice message. The default value is `$true`. When this setting is enabled, MWI notifications are sent to users who are linked with a single UM mailbox policy for calls taken by a UM IP gateway. This setting allows the UM IP gateway to receive and send SIP NOTIFY messages to UM-enabled users' phones or SIP endpoints.

For more information about how to manage MWI settings on a UM mailbox policy, see the following topics:

- Manage a UM mailbox policy
- Enable Message Waiting Indicator (MWI) for users

- Disable Message Waiting Indicator (MWI) for users
- Set-UMMailboxPolicy

UM IP gateways and MWI

If you disable MWI on a UM IP gateway, you'll disable MWI notifications for all users who connect to the VoIP gateway or IP PBX that's represented by the UM IP gateway. Disabling MWI on a single UM IP gateway that's linked to a UM dial plan can disable MWI notifications for all UM-enabled users associated with a single or multiple UM dial plans or a single or multiple UM mailbox policies. To learn more about UM mailbox policies, including how to enable or disable MWI for a group of UM-enabled users, see Manage a UM mailbox policy.

You can use the EAC or the **Set-UMMailboxPolicy** cmdlet in the Shell to configure the MWI setting, as shown in the following table.

Message Waiting Indicator setting on a UM IP gateway

Shell parameter	Setting available in the EAC?	Description
MessageWaitingIndicatorAllowed	Yes	The *MessageWaitingIndicatorAllowed* parameter specifies whether to enable the UM IP gateway to allow SIP NOTIFY messages to be sent to users associated with a UM dial plan. The default value is `$true`. When this setting is enabled, voice mail notifications can be sent to users for calls that are received by the UM IP gateway. This setting allows the UM IP gateway to send message-waiting notifications to UM-enabled users.

For more information about how to manage MWI settings, see the following topics:

- Manage a UM IP gateway
- Allow Message Waiting Indicator (MWI) on a UM IP gateway
- Prevent Message Waiting Indicator (MWI) on a UM IP gateway
- Set-UMIPGateway

Return to top

Text message (SMS) notifications for voice mail messages and missed calls

As mentioned earlier, an MWI notification is any mechanism that indicates the existence of a new voice mail message. In addition to the mechanisms already discussed, users can be notified that they have a voice message waiting via a text message, also called an SMS (Short Message Service) message. This is a different type of MWI notification for new voice messages than the traditional light or other mechanisms.

A text message is sent to a user's mobile phone when a caller leaves a new voice message. Users can also receive a text message that notifies them when they miss a phone call and a voice message isn't left. The missed call notification text message can be sent to the user along with the new voice mail notification.

Note:
The text message that's sent to a user includes voice mail preview.

Text message notifications use different settings than the MWI settings on the UM IP gateway or the UM mailbox policy. Text message notifications for new voice mail and missed calls are configured on UM mailbox policies and UM mailboxes. You can enable or disable text message notifications by using the **Set-UMMailboxPolicy** cmdlet and the **Set-UMMailbox** cmdlet in the Shell. You can view the status of text message notifications by using the **Get-UMMailboxPolicy** cmdlet and the **Get-UMMailbox** cmdlet. It's not possible to configure text message notifications in the EAC.

The following table shows the parameter on a UM mailbox that must be configured for a user to receive text messages for voice mail and missed call notifications:

Text message notification settings on a user's mailbox

UMSMSNotificationOption	No	The *UMSMSNotificationOption* parameter specifies whether a UM-enabled user can receive text message notifications for voice mail only, for voice mail and missed calls, or isn't allowed to receive notifications. The values for this parameter are: `VoiceMail`, `VoiceMailAndMissedCalls`,

		and `None`. The default value is `None`.

For more information about how to manage text message notification settings on a user's mailbox, see the following topics:

- Manage voice mail settings for a user
- Set-UMMailbox

The following table shows the parameter on a UM mailbox policy that must be configured for a user to receive text messages for voice mail and missed call notifications:

Text message and missed call notification settings on a UM mailbox policy

Shell parameter	Setting available in the EAC?	Description
AllowSMSNotification	No	The *AllowSMSNotification* parameter specifies whether UM-enabled users whose mailboxes are associated with the UM mailbox policy are allowed to receive text message notifications on their mobile phones. If this parameter is set to `$true`, you must also use the **Set-UMMailbox** cmdlet and set the *UMSMSNotificationOption* parameter for the UM-enabled user to either `voicemail` or `VoiceMailAndMissedCalls`. The default value is `$true`.

For more information about how to manage text message notification settings, see the following topics:

- Manage a UM mailbox policy
- Set-UMMailboxPolicy

For text message notifications for voice mail and missed calls to work correctly, you must perform the following tasks:

1. Use either the EAC or the Shell to enable the user for UM and link them to the correct UM mailbox policy.
2. On the UM mailbox policy that's linked to the user, verify that the *AllowSMSNotification*

parameter is set to `$true`. To set the parameter to `$true`, run the following command: `Set-UMMailboxPolicy -id MyUMMailboxPolicy - AllowSMSNotification $true`.

3. On the user's mailbox, enable text message notifications by setting the *UMSMSNotificationOption* parameter to `VoiceMailAndMissedCalls` or `VoiceMail`.
4. Because the default setting is `None`, you must run the following command from the Shell and set the text message notification option to either `VoiceMailAndMissedCalls` or `VoiceMail`. For example: `Set-UMMailbox- -id MyUMMailbox -UMSMSNotificationOption VoiceMailAndMissedCalls`.

Important:
The *AllowSMSNotification* parameter on the UM mailbox policy and the *UMSMSNotificationOption* parameter on the user's mailbox must both be set to `$true` for SMS notifications to work.

In addition to your configuring the UM mailbox policy and the user's mailbox to enable text message notifications for new voice mail and missed calls, the user must enable and configure text message notifications when they sign in to Outlook Web App. To set up and configure text message notifications, the user must:

1. Sign in to Outlook Web App and go to **Options** > **Phone** > **Voice mail**.
2. On the **Voice Mail** page, under **Notifications**, click **Set up notifications**.
3. On the **Text messaging** page, click the **Turn on notifications** button.

Warning:
Don't click **Voice mail notifications** or it will take you back to the **Voice mail** page.

4. On the **Text messaging** page, under **Locale**, use the drop-down list to select the locale or location of the text messaging mobile operator.
5. On the **Text messaging** page, under **Mobile operator**, use the drop-down list to select the text messaging mobile operator, and then click **Next**.
6. On the **Text messaging** page, in the **Enter your phone number and click Next** box, enter the mobile phone number that's used for text message notifications, and then click **Next**. A six-digit passcode will be sent to the mobile phone. If you didn't receive a passcode, click **I didn't receive a passcode and need it sent again**.
7. Enter the passcode in the **Passcode** box, and then click **Finish**.
8. After the user enables text message notifications, they can click **Set up voice mail notifications** on the **Text Messaging** page. They'll be taken back to the voice mail page, where they can scroll down to the **Notifications** section and set up text message notification options for missed calls and voice mail.

Return to top

Allow Message Waiting Indicator procedures

Topic Last Modified: *2013-05-03*

Allow Message Waiting Indicator (MWI) on a UM IP gateway

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-23*

You can allow or prevent voice mail notifications to users for calls received by a Unified Messaging (UM) IP gateway. If you enable this setting, the UM IP gateway can receive and send SIP NOTIFY messages for users. Message Waiting Indicator (MWI) is enabled by default and allows message waiting notifications to be sent to users, but you can turn it off depending on your needs.

A message waiting indicator notifies a user about a new or unheard voice message. It appears in the Inbox in clients such as Outlook and Outlook Web App. It can also be a text (SMS) message sent to a registered mobile phone, an outgoing call made from an Exchange server to a number that's been configured for playing new messages, or a lighted lamp on a user's desktop phone.

Tip:
MWI notifications can also be enabled and disabled on a UM mailbox policy for a group of users.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging

permissions topic.

- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to allow Message Waiting Indicator

1. In the EAC, navigate to **Unified Messaging** > **UM IP Gateways**, select the UM IP gateway you want to change, and then click **Edit** .
2. On the **UM IP Gateway** page, select the check box next to **Allow message waiting indicator**.
3. Click **Save**.

Use the Shell to allow Message Waiting Indicator

This example allows the message waiting indicator to appear for users who are associated with the UM IP gateway named `MyUMIPGateway` with an IP address of 10.10.10.1.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address 10.10.10.1
-MessageWaitingIndicatorAllowed $true
```

Prevent Message Waiting Indicator (MWI) on a UM IP gateway

Set up client voice mail features > Allow Message Waiting Indicator > Allow Message Waiting Indicator procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

You can prevent voice mail notifications to users for calls received by a Unified Messaging (UM) IP gateway. If you enable this setting, the UM IP gateway can receive and send SIP NOTIFY messages

for users. Message Waiting Indicator (MWI) is enabled by default and allows message waiting notifications to be sent to users, but you can turn it off depending on your needs.

A message waiting indicator notifies a user about a new or unheard voice message. It appears in the Inbox in clients such as Outlook and Outlook Web App. It can also be a text (SMS) message sent to a registered mobile phone, an outgoing call made from an Exchange server to a number that's been configured for playing new messages, or a lighted lamp on a user's desktop phone.

Tip:

MWI notifications can also be enabled and disabled on a UM mailbox policy for a group of users.

For additional management tasks related to UM IP gateways, see UM IP gateway procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to prevent Message Waiting Indicator

1. In the EAC, navigate to **Unified Messaging** > **UM IP Gateways**, select the UM IP gateway you want to change, and then click **Edit** ✎.
2. On the **UM IP Gateway** page, clear the check box next to **Allow message waiting indicator**.
3. Click **Save**.

Use the Shell to prevent Message Waiting Indicator

This example prevents the message waiting indicator from appearing for users who are associated with the UM IP gateway named `MyUMIPGateway` with an IP address of 10.10.10.1.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address 10.10.10.1
-MessageWaitingIndicatorAllowed $false
```

Enable Message Waiting Indicator (MWI) for users

Set up client voice mail features > Allow Message Waiting Indicator > Allow Message Waiting Indicator procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

You can enable or disable Message Waiting Indicator for users associated with a Unified Messaging (UM) mailbox policy. Message Waiting Indicator is a feature found in most legacy voice mail systems. In its most common form, it lights a lamp on a voice mail subscriber's phone to indicate the presence of a new voice mail message. Message Waiting Indicator can also send a text message to a UM-enabled user's mobile phone. The default setting is enabled.

If Message Waiting Indicator is disabled on the UM IP gateway, the feature isn't available to UM-enabled users associated with the UM mailbox policy.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable Message Waiting Indicator

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. Under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page, select the check box next to **Allow Message Waiting Indicator**.
4. Click **Save**.

Use the Shell to enable Message Waiting Indicator

This example enables Message Waiting Indicator for users associated with the UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowMessageWaitingIndicator $true
```

Disable Message Waiting Indicator (MWI) for users

Set up client voice mail features > Allow Message Waiting Indicator > Allow Message Waiting Indicator procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

You can enable or disable Message Waiting Indicator for users associated with a Unified Messaging (UM) mailbox policy. Message Waiting Indicator is a feature found in most legacy voice mail systems. In its most common form, it lights a lamp on a voice mail subscriber's phone to indicate the presence of a new voice mail message. Message Waiting Indicator can also send a text message to a UM-enabled user's mobile phone. The default setting is enabled.

If Message Waiting Indicator is disabled on the UM IP gateway, the feature isn't available to UM-enabled users associated with the UM mailbox policy.

For additional management tasks related to UM mailbox policies, see UM mailbox policy procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to disable Message Waiting Indicator

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. Under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page, clear the check box next to **Allow Message Waiting Indicator**.
4. Click **Save**.

Use the Shell to disable Message Waiting Indicator

This example disables Message Waiting Indicator for users associated with the UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowMessageWaitingIndicator $false
```

Enable missed call notifications for a user

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-12-09*

You can enable or disable missed call notifications for a Unified Messaging (UM) mailbox policy by using the Shell or the EAC. A missed call notification is an email message that's sent to a user when the user doesn't answer an incoming call and the caller doesn't leave a voice mail message. This is a different email message than the message that contains the voice message that's left for a user.

When you disable missed call notifications on a UM mailbox policy, you prevent all users associated with the UM mailbox policy from receiving an email message when they don't answer an incoming call and the caller doesn't leave a voice message. By default, missed call notifications are enabled for each UM mailbox policy that's created. Also by default, a UM mailbox policy is created every time you create a UM dial plan.

Note:

When you're integrating Unified Messaging and Microsoft Lync Server, missed call notifications aren't available to users that have a mailbox located on an Exchange 2007 or Exchange 2010 Mailbox server when a user disconnects before the call is sent to a Mailbox server running the Microsoft Exchange Unified Messaging service.

For additional management tasks related to UM mailbox policies, see Manage a UM mailbox policy.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable missed call notifications for a UM mailbox policy

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **General**, select the check box next to **Allow missed call notifications**.
4. Click **Save**.

Use the Shell to enable missed call notifications for a UM mailbox policy

This example enables missed call notifications for a UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowMissedCallNotifications $true
```

Disable missed call notifications for a user

Set up client voice mail features > Allow Message Waiting Indicator > Allow Message Waiting Indicator procedures >

Applies to: *Exchange Online*

Topic Last Modified: *2012-12-09*

You can enable or disable missed call notifications for a Unified Messaging (UM) mailbox policy by using the Shell or the EAC. A missed call notification is an email message that's sent to a user when the user doesn't answer an incoming call and the caller doesn't leave a voice message. This is a different email message than the one that contains the voice message that's left for a user.

When you disable missed call notifications on a UM mailbox policy, you prevent all users associated with the UM mailbox policy from receiving an email message when they don't answer an incoming call and the caller doesn't leave a voice message. By default, missed call notifications are enabled for each UM mailbox policy that's created. Also by default, a UM mailbox policy is created every time you create a UM dial plan.

Note:

When you're integrating Unified Messaging and Microsoft Lync Server, missed call notifications aren't available to users that have a mailbox located on an Exchange 2007 or Exchange 2010 Mailbox server when a user disconnects before the call is sent to a Mailbox server running the Microsoft Exchange Unified Messaging service.

For additional management tasks related to UM mailbox policies, see Manage a UM mailbox policy.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to disable missed call notifications for a UM mailbox policy

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **General**, clear the check box next to **Allow missed call notifications**.
4. Click **Save**.

Use the Shell to disable missed call notifications for a UM mailbox policy

This example disables missed call notifications for a UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
AllowMissedCallNotifications $false
```

Set Outlook Voice Access PIN security

Exchange Server 2013 > Unified Messaging >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-16*

When Unified Messaging (UM) users connect to the voice mail system by telephone, they use Outlook Voice Access to navigate the menu system. Before users can access the voice mail system, the system prompts them to enter their PIN. As the administrator, you can configure PIN settings and requirements and perform PIN management tasks. After a user has been enabled for voice mail and a PIN has been generated, the user's PIN is stored encrypted in the user's mailbox.

Note:
Outlook Voice Access users must use touchtone (also called dual tone multi-frequency (DTMF)) inputs to enter their PIN to access their UM-enabled mailbox. Speech recognition isn't available for PIN entry.

Contents

PIN overview

A PIN is a numeric string that's used in certain systems so that a user can be authenticated and gain access to the system. PINs are most frequently used for automatic teller machines (ATMs). They're also used instead of alphanumeric passwords for voice mail systems. The strength of a PIN depends on its length, how well it's protected, and how difficult it is to guess.

In Unified Messaging, Outlook Voice Access users enter their PIN on an analog, digital, or mobile telephone so that they can access email, voice mail, contact, and calendaring information in their Exchange Server mailbox.

In UM, PIN policies are defined and configured on a UM mailbox policy. You can create multiple UM mailbox policies depending on your requirements. When you enable a user for voice mail, you

link the user to an existing UM mailbox policy. The UM PIN policies that are configured on the UM mailbox policy should be based on the security requirements of your organization.

PIN requirements

The following are several PIN configuration settings that you can set on a UM mailbox policy.

Minimum PIN length

The **Minimum PIN length** setting specifies the minimum number of digits that a mailbox PIN must contain. The range is 4 through 24, and the default is 6. If you enter 0, users aren't required to enter a PIN.

Important:
Configuring this setting with zero isn't a recommended practice. If you configure the setting to zero, you greatly decrease the level of security for your network.

If you change the minimum PIN length to a higher value, current Outlook Voice Access users will be prompted to create a new PIN that contains the new minimum number of digits before they can continue.

Note:
Increasing this number creates a more secure UM environment. However, setting it too high can result in users forgetting their PIN.

Enforce PIN lifetime

The **Enforce PIN lifetime** setting controls the time interval, in days, from the date Outlook Voice Access users last changed their PIN to the date they'll be forced to change their PIN again. The range is 0 through 999, and the default is 60 days. If 0 is entered, the PIN won't expire.

Note:
Unified Messaging won't notify users when their PIN is about to expire.

Number of sign-in failures before PIN reset

The **Number of sign-in failures before PIN reset** setting specifies the number of sequential unsuccessful sign-in attempts before the mailbox PIN is automatically reset. To disable this feature, set this setting to unlimited. Otherwise, it must be set to a number lower than the **Number of sign-in failures before lockout** setting. The range is 1 through 998, and the default is 5.

Note:
To increase security for UM-enabled users, enter a number that's less than 5.

Number of sign-in failures before lockout

The **Number of sign-in failures before lockout** setting specifies how many PIN entry errors in successive calls Outlook Voice Access users can make before they're locked out of their mailbox. By default, after 5 attempts are made, the PIN is automatically reset. The range is 1 through 999, and the default is 15.

Note:

To increase security, decrease the number of failed attempts that are allowed. But remember that decreasing it to a number much lower than the default may result in users being locked out unnecessarily. Unified Messaging will generate warning events that can be viewed using Event Viewer if PIN authentication fails for a UM-enabled user or the user is unsuccessful in trying to sign in to the system.

Allow common PIN patterns

The **Allow common PIN patterns** setting is used to either enable or disable the use of common number patterns when creating a PIN. By default, this setting is disabled and won't allow Outlook Voice Access users to enter the following number patterns:

- **Sequential numbers** PIN values that consist completely of consecutive numbers. Examples of sequential numbers for a PIN are 1234 and 65432.
- **Repeated numbers** PIN values that consist of repeated numbers. Examples of repeated numbers are 11111 and 22222.
- **Suffix of mailbox extension** PIN values that consist of the suffix of a user's mailbox extension. If the mailbox extension is 36697, the PIN can't be 6697.

PIN recycle count

The **PIN recycle count** setting configures the number of different PINs a user must use before any PINs that were previously used can be reused. The range is 1 through 20, and the default is 5.

Managing Outlook Voice Access PINs

When planning for Outlook Voice Access PINs, you must choose the appropriate levels of security for your organization. You must carefully consider the Outlook Voice Access PIN requirements and how your PIN security settings meet or exceed your organization's security policy.

Important:

It's a security best practice to implement strong PIN requirements for Outlook Voice Access users. This can be enforced by creating UM mailbox policy PIN policies that require six or more digits for PINs, which increases the level of security for your network.

After you set the Outlook Voice Access PIN requirements, you must create and configure a UM mailbox policy to enforce your organizational PIN requirements. For details about how to create a

UM mailbox policy, see Create a UM mailbox policy. For details about how to manage UM mailbox policies, see Manage a UM mailbox policy.

Note:

After you create the UM mailbox policy, you must link the UM-enabled user or users with the appropriate UM mailbox policy. You can do this by using the **Enable-UMMailbox** cmdlet in the Exchange Management Shell or by using the Exchange Administration Center (EAC). For more information about the Exchange Management Shell cmdlet, see Enable-UMMailbox.

There are situations in which Outlook Voice Access users forget their PIN or are locked out of voice mail access to their mailbox. In either case, it may be necessary for you to reset a UM-enabled user's PIN. For details, see Reset a voice mail PIN.

You can retrieve PIN information for a user who is enabled for Unified Messaging. The information returned to you is calculated by using the encrypted PIN data stored in the user's mailbox. This lets you view PIN information for the user and also indicates whether the user has been locked out of their mailbox. For details, see Retrieve voice mail PIN information.

PIN security procedures

Exchange Server 2013 > Unified Messaging > Set Outlook Voice Access PIN security >

Applies to: *Exchange Online*

Topic Last Modified: *2013-04-16*

Set Outlook Voice Access PIN policies

Reset a voice mail PIN

Retrieve voice mail PIN information

Include text with the email message sent when a PIN Is reset

Set the minimum PIN length for voice mail

Set the PIN lifetime for voice mail

Set the number of previous voice mail PINs to recycle

Disable common PIN patterns for voice mail

Enable common PIN patterns for voice mail

Set the number of sign-in failures before a voice mail PIN is reset

Set the number of sign-in failures before a voice mail user Is locked out

Enable PIN-less sign-Ins for voice mail

Set Outlook Voice Access PIN policies

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can set PIN policies on a Unified Messaging (UM) mailbox policy. UM mailbox policies can be configured to increase the level of security for UM-enabled users that use Outlook Voice Access by requiring users to comply with the predefined PIN policies for your organization.

To set PIN policies for Outlook Voice Access users, you can either create a new UM mailbox policy or modify an existing UM mailbox policy. After a new UM mailbox policy is created, you can then configure the UM mailbox policy by configuring the following PIN settings:

- `MinPasswordLength`
- `PINLifetime`
- `LogonFailuresBeforePINReset`
- `MaxLogonAttempts`
- `AllowCommonPatterns`
- `PINHistoryCount`

It's a security best practice to implement strong PIN requirements for UM users. This can be enforced by creating UM PIN policies that require 6 or more digits for PINs and increase the level of security for your network.

When you change the PIN policy, the new PIN setting is applied to users who are currently associated with the UM mailbox policy. For example, if you modify the UM mailbox policy and change the minimum PIN length from 7 to 10 digits, the next time users log on they'll be forced to change their PIN to comply with the changed PIN requirement.

For additional tasks related to Outlook Voice Access PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to set PIN policies for Outlook Voice Access users

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, click the UM dial plan you want to edit, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to edit, and then click **Edit** .
3. Click **Properties**.
4. On the **UM mailbox policy** page, click **PIN policies**.
5. On the **PIN Policies** page, configure the PIN settings for the Outlook Voice Access users associated with this UM mailbox policy, and then click **Save**.

Use the Shell to set PIN policies for Outlook Voice Access users

This example sets the PIN settings for users associated with the UM mailbox policy `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
LogonFailuresBeforePINReset 8 -MaxLogonAttempts 12 -
MinPINLength 8 -PINHistoryCount 10 -PINLifetime 60 -
ResetPINText "The PIN used to allow you access to your
mailbox using Outlook Voice Access has been reset."
```

Reset a voice mail PIN

Unified Messaging > Set Outlook Voice Access PIN security > PIN security procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

When a Unified Messaging (UM)-enabled voice mail user is locked out of their mailbox using

Outlook Voice Access because they tried to sign in using an incorrect PIN multiple times or they forgot their PIN, you can use one of the following procedures to reset the user's PIN. When you reset a user's Outlook Voice Access PIN, you can configure UM to automatically generate a PIN or you can manually specify the PIN. The new PIN is sent to the user in email. You can specify additional PIN options such as requiring the user to reset their PIN when they first sign in. Users can also reset their UM PIN using Outlook or Outlook Web App.

Note:

To access their UM-enabled mailboxes, Outlook Voice Access users need to use touchtone, also known as dual tone multi-frequency (DTMF), inputs. Speech recognition isn't available for PIN input.

For additional tasks related to Outlook Voice Access PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to reset a Unified Messaging PIN

1. In the EAC, navigate to **Recipients**. In the list view, select the user mailbox that you want to view.
2. In the details pane, under **Phone and Voice Features**, under **Unified Messaging**, click **View details**.
3. On the **UM Mailbox** page, under **UM mailbox settings**, click **Reset PIN**.
4. On the **Reset UM Mailbox PIN** page, use the following options to reset the UM-enabled user's PIN:
   - **Automatically generate a PIN** Use this option to automatically generate the PIN that's used by the user to gain access to their mailbox using Outlook Voice Access. By default, this setting is enabled.

The automatically generated PIN will be sent in an email message to the user's mailbox. After they receive the PIN and sign in to their mailbox, they'll be prompted to change the PIN to a PIN that's more familiar to them.

Outlook Web App and Microsoft Outlook also let the user reset their PIN. The PIN is automatically

generated based on the PIN policies that are configured on the UM mailbox policy that's associated with the user's mailbox. We recommend that you automatically generate PINs for Outlook Voice Access users.

- **Type a PIN** Use this option to manually specify a PIN for an Outlook Voice Access user. By default, this setting is disabled.

If you specify a PIN for a user, the PIN will be sent in an email message to the user's mailbox. After they receive the PIN and sign in to their mailbox, they can change the PIN by configuring personal options in Outlook Voice Access. However, in Outlook Web App and Microsoft Outlook, there is no option to manually specify a PIN.

- **Require the user to reset their PIN the first time they sign in** Use this option to require the user to reset their PIN when they first sign in to Outlook Voice Access. By default, this option is enabled.

If you select the option to automatically generate a PIN for a user, you can enable this option to require users to change their PIN when they first sign in to Outlook Voice Access. This helps protect the user's PIN.

5. Click **Save**.

Use the Shell to reset a Unified Messaging PIN

This example resets the voice mail PIN for Tony Smith to 1985848. However, this PIN must be changed when the user first signs in to Outlook Voice Access.

```
Set-UMMailboxPIN -Identity tonysmith@contoso.com -PIN
1985848 -PinExpired $true
```

Retrieve voice mail PIN information

Unified Messaging > Set Outlook Voice Access PIN security > PIN security procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-03*

You can retrieve PIN information for a user who is enabled for Unified Messaging (UM). After a user has been enabled for UM-enabled and a PIN is generated or created, the PIN is encrypted and stored in the user's mailbox.

When you retrieve PIN information for a UM-enabled user, the information returned to you is calculated by using the encrypted PIN data stored in the user's mailbox. This lets you view information from the user's mailbox and also indicates whether the user has been locked out of the mailbox.

For additional tasks related to PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user's mailbox has been UM-enabled. For detailed steps, see Enable a user for voice mail.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to retrieve PIN information for a UM-enabled user

1. In the EAC, navigate to **Recipients**. In the list view, select the user mailbox that you want to view.
2. In the details pane, under **Phone and Voice Features**, click **View details**.
3. On the **UM Mailbox** page > **UM mailbox settings**, view the **PIN status** for the user. On this page, you can also reset the voice mail PIN for the user.

Use the Shell to retrieve PIN information for a UM-enabled user

This example displays the user ID, whether a PIN is expired, whether the UM mailbox is locked out, and whether Tony is a first-time user.

```
Get-UMMailboxPIN -identity tony@contoso.com
```

Include text with the email message sent when a PIN Is reset

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can include additional text in the email message that's sent to users when their Unified Messaging (UM) or voice mail PIN is reset. You do this by entering custom text in the **When a user's Outlook Voice Access PIN is reset** box on a UM mailbox policy. The customized text can include, for example, security-related information for UM-enabled users.

By default, a PIN used for Outlook Voice Access is reset by the Unified Messaging or voice mail system if the number of failed sign-in attempts exceeds 5. Users can also reset their PINs using the UM features included with Outlook Web App or Outlook 2010 or later, or by using Outlook Voice Access from a telephone.

Note:

The text you enter in this box is limited to 512 characters, and can include simple HTML text.

For additional tasks related to Outlook Voice Access PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to add text to the email message sent to users when their PIN is reset

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to change, and then click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then click **Edit** .
3. On the **UM Mailbox Policy** page > **Message text**, in the text box for **When a user's Outlook Voice Access PIN is reset**, enter the text you want to include in the email message that's sent when a user's PIN is reset.
4. Click **Save**.

Use the Shell to add text to the email message sent to users when their PIN is reset

This example includes the additional text, "Do not share your PIN with other users. Doing so may result in disciplinary action", in the email message sent to users who are associated with the UM mailbox policy `MyUMMailboxPolicy` when their PIN is reset.

```
Set-UMMailboxPolicy -identity MyUMMailboxPolicy -
ResetPINText "Do not share your PIN with other users. Doing
so may result in disciplinary action."
```

Set the minimum PIN length for voice mail

Unified Messaging > Set Outlook Voice Access PIN security > PIN security procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can configure the minimum PIN length for your Outlook Voice Access users who are enabled for Unified Messaging (UM). The PIN settings that you configure on a UM mailbox policy will apply to all UM-enabled users associated with the UM mailbox policy.

Outlook Voice Access is used by UM-enabled users to access their voice mail, email, calendar, and personal contact information located in their mailbox. However, before they can access their mailbox, they must enter a PIN so they can be authenticated by the voice mail system.

Note:

If you change the minimum PIN length value, existing Outlook Voice Access users will be prompted to enter a new PIN that contains the new minimum number of digits before they can continue. The default is 6.

For additional tasks related to Outlook Voice Access PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the minimum PIN length for Outlook Voice Access

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the dial plan you want to change, and then click **Edit** .
3. On the **UM dial plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to change, and then click **Edit** .
4. Click **PIN policies**, and next to **Minimum PIN length**, enter a value between 4 and 24.
5. Click **Save**.

Use the Shell to configure the minimum PIN length for Outlook Voice Access

This example sets the minimum PIN length to 8 digits for Outlook Voice Access users who are associated with the UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
MinPINLength 8
```

This example sets the minimum PIN length to 8 digits and sets the number of times a sign-in can fail before the user's PIN is reset to 3. This applies to UM-enabled users who are associated with the UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
LogonFailuresBeforePINReset 3 -MinPINLength 8
```

Set the PIN lifetime for voice mail

Unified Messaging > Set Outlook Voice Access PIN security > PIN security procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can configure the PIN lifetime for users who are enabled for Unified Messaging (UM). The PIN lifetime is the maximum time that an Outlook Voice Access PIN will be valid for UM-enabled recipients. The PIN lifetime setting is configured on a UM mailbox policy and applies to all UM-enabled users associated with the UM mailbox policy.

Several PIN-related settings can be configured on a UM mailbox policy. The PIN lifetime setting controls the time interval, in days, from the date Outlook Voice Access users last changed their PIN to the date they'll be forced to change their PIN again. The range is 0 through 999, and the default is 60 days. If you enter 0, the user's PIN won't expire. We recommend that you don’t configure this setting to 0, because by doing so you greatly reduce the security of your network.

Important:
Unified Messaging doesn't notify users when their PIN is about to expire.

For additional tasks related to Outlook Voice Access PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For

detailed steps, see Create a UM mailbox policy.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the PIN lifetime

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to change, and then click **Edit** .
3. On the **UM dial plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to change, and then click **Edit** .
4. Click **PIN policies**, and next to **Enforce PIN lifetime (days)**, enter a value between 0 and 999.
5. Click **Save**.

Use the Shell to configure the PIN lifetime

This example sets the number of days that a PIN can be used for Outlook Voice Access users who are associated with a UM mailbox policy named `MyUMMailboxPolicy` to 30.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
PINLifetime 30
```

This example configures the following PIN-related settings for Outlook Voice Access users who are associated with a UM mailbox policy named `MyUMMailboxPolicy`:

- Sets the number of logon failures before the user's PIN is reset to 3.
- Sets the maximum number of logon attempts to 5.
- Sets the minimum PIN length to 9 digits.
- Sets the PIN to expire in 40 days.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
LogonFailuresBeforePINReset 3
-MaxLogonAttempts 5 -MinPINLength 9 -PINLifetime 40
```

Set the number of previous voice mail

PINs to recycle

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

When Outlook Voice Access users dial in to an Outlook Voice Access number, they're prompted to enter their PIN so that the voice mail system can authenticate them. After they're authenticated, they can access the voice mail, email, calendaring, and personal contact information in their mailbox from any telephone.

Several PIN-related settings can be configured on a Unified Messaging (UM) mailbox policy. The **PIN recycle count** setting specifies the number of unique PINs users must use before they can reuse an old PIN. You can set the value of this setting between 1 and 20. For most organizations, this value should be set to 5 PINs, which is the default. Setting this value too high can frustrate users because it can be difficult for users to create and memorize many PINs. Setting it too low may introduce a security threat to your network.

Important:
The PIN recycle count can't be disabled.

For additional tasks related to Outlook Voice Access PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to change the PIN recycle count

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the dial plan you want to change, and then click **Edit** .
3. On the **UM dial plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to change, and then click **Edit** .
4. Click **PIN policies**, and next to **PIN recycle count**, enter a value between 1 and 20.
5. Click **Save**.

Use the Shell to change the PIN recycle count

This example sets the PIN recycle count on the UM mailbox policy `MyUMMailboxPolicy` to 10.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
PINHistoryCount 10
```

Disable common PIN patterns for voice mail

Unified Messaging > Set Outlook Voice Access PIN security > PIN security procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can enable or disable common Unified Messaging (UM) PIN patterns for Outlook Voice Access users. If you enable or disable the common PIN patterns setting on a UM mailbox policy, the setting will apply to all UM-enabled users associated with the UM mailbox policy. By default, UM-enabled users can't use common patterns when they create a PIN.

You can configure several PIN-related settings on a UM mailbox policy. The **Allow Common PIN Patterns** setting is used to allow or prevent the use of common number patterns when users create a PIN. By default, this setting is disabled and prevents users from using the following number patterns:

- **Sequential numbers** These are PIN values that include only consecutive numbers. Examples of consecutive numbers for a PIN are 1234 and 65432.
- **Repeated numbers** These are PIN values that include only repeated numbers. Examples of repeated numbers are 11111 and 22222.
- **Suffix of mailbox extension** These are PIN values that include the suffix of a user's mailbox extension. For example, if a user's mailbox extension is 36697, the user's PIN cannot be 3669712.

Note:
If the **Allow Common PIN Patterns** setting is enabled, only the suffix of the mailbox extension will be rejected.

For additional tasks related to Outlook Voice Access PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to disable common PIN patterns

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then on the toolbar, click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then on the toolbar, click **Edit** .
3. On the **UM Mailbox Policy** page, under **PIN polices**, clear the check box next to **Allow common PIN patterns**.
4. Click **Save**.

Use the Shell to disable common PIN patterns

This example prevents users associated with the UM mailbox policy named `MyUMMailboxPolicy` from using PINs that contain common patterns.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
AllowCommonPatterns $false
```

Enable common PIN patterns for voice

mail

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can enable or disable common Unified Messaging (UM) PIN patterns for Outlook Voice Access users. If you enable or disable the common PIN patterns setting on a UM mailbox policy, the setting will apply to all UM-enabled users associated with the UM mailbox policy. By default, UM-enabled users can't use common patterns when they create a PIN.

You can configure several PIN-related settings on a UM mailbox policy. The **Allow Common PIN Patterns** setting is used to allow or prevent the use of common number patterns when users create a PIN. By default, this setting is disabled and prevents users from using the following number patterns:

- **Sequential numbers** These are PIN values that include only consecutive numbers. Examples of consecutive numbers for a PIN are 1234 and 65432.
- **Repeated numbers** These are PIN values that include only repeated numbers. Examples of repeated numbers are 11111 and 22222.
- **Suffix of mailbox extension** These are PIN values that include the suffix of a user's mailbox extension. For example, if a user's mailbox extension is 36697, the user's PIN cannot be 3669712.

Note:

If the **Allow Common PIN Patterns** setting is enabled, only the suffix of the mailbox extension will be rejected.

For additional tasks related to Outlook Voice Access PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable common PIN patterns

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**. In the list view, select the UM dial plan you want to modify, and then on the toolbar, click **Edit** .
2. On the **UM Dial Plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to manage, and then on the toolbar, click **Edit** .
3. On the **UM Mailbox Policy** page, under **PIN polices** select the check box next to **Allow common PIN patterns**.
4. Click **Save**.

Use the Shell to enable common PIN patterns

This example allows users associated with the UM mailbox policy named `MyUMMailboxPolicy` to use PINs that contain common patterns.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
AllowCommonPatterns $true
```

Set the number of sign-in failures before a voice mail PIN is reset

Unified Messaging > Set Outlook Voice Access PIN security > PIN security procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can configure the number of sign-in failures allowed before the PIN is reset for an Outlook Voice Access user to a value from 1 through 998. The default is 5. The number of sign-in failures allowed before a PIN is reset is configured on a Unified Messaging (UM) mailbox policy and applies to all Outlook Voice Access users associated with the UM mailbox policy.

Note:

You can increase security by configuring the **Number of sign-in failures before PIN reset** setting to a number less than 5. You decrease security if you configure it to a number more than 5.

For additional tasks related to Outlook Voice Access PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the number of sign-in failures before a PIN is reset

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to change, and then click **Edit** .
3. On the **UM dial plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to change, and then click **Edit** .
4. Click **PIN policies**, and next to **Number of sign-in failures before PIN reset**, enter a value between 0 and 999.
5. Click **Save**.

Use the Shell to configure the number of sign-in failures before a PIN is reset

This example sets the number of sign-in failures before the user's PIN is reset to 3 for UM-enabled users who are associated with a UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
LogonFailuresBeforePINReset 3
```

This example sets the number of sign-in failures before the user's PIN is reset to 3, the maximum

number of sign-in attempts to 5, and the minimum PIN length to 9 for UM-enabled users who are associated with a UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
LogonFailuresBeforePINReset 3 -MaxLogonAttempts 5 -
MinPINLength 9
```

Set the number of sign-in failures before a voice mail user Is locked out

Unified Messaging > Set Outlook Voice Access PIN security > PIN security procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can configure the number of sign-in failures allowed before an Outlook Voice Access user is locked out of their mailbox. The number of sign-in failures allowed before a voice mail user is locked out is configured on a Unified Messaging (UM) mailbox policy, and applies to all UM-enabled users associated with the UM mailbox policy. By default it is set to 15.

To increase security, decrease the maximum number of failed attempts. However, remember that if you decrease it to a number much lower than the default, users may be locked out unnecessarily. Unified Messaging will generate warning events you can view using Event Viewer if PIN authentication fails for UM-enabled users or if users are unsuccessful when they try to sign in to the system. This setting must be larger than the setting for the number of sign-in failures before the PIN is reset.

For additional tasks related to Outlook Voice Access PIN security, see PIN security procedures.

What do you need to know before you begin?

- Estimated time to complete: Less than 1 minute.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the number of sign-in failures before a voice mail user is locked out

1. In the EAC, navigate to **Unified Messaging** > **UM dial plans**.
2. In the list view, select the UM dial plan you want to change, and then click **Edit** .
3. On the **UM dial plan** page, under **UM Mailbox Policies**, select the UM mailbox policy you want to change, and then click **Edit** .
4. Click **PIN policies**, and next to **Number of sign-in failures before lockout**, enter a value between 1 and 999.
5. Click **Save**.

Use the Shell to configure the number of sign-in failures before a voice mail user is locked out

This example sets the maximum number sign-in attempts to 10 for UM-enabled users who are associated with a UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
MaxLogonAttempts 10
```

This example sets the number of sign-in failures before the Outlook Voice Access user's PIN is reset to 3, the maximum number of sign-in attempts to 5, and a minimum PIN length to 9 for UM-enabled users who are associated with a UM mailbox policy named `MyUMMailboxPolicy`.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
LogonFailuresBeforePINReset 3
-MaxLogonAttempts 5 -MinPINLength 9
```

Enable PIN-less sign-Ins for voice mail

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-04-16*

You can set up Unified Messaging (UM) so that your users can sign in to their voice mail without using a PIN. By default, Outlook Voice Access users are prompted to enter a PIN to sign in to their mailbox and access their voice mail, email, calendar, personal contacts, the directory, and personal options.

Warning:
Enabling PIN-less sign-ins for a single user or a group of users that are enabled for voice mail decreases the level of security for voice mail and poses a security risk to your organization.

To enable PIN-less sign-ins, you must set the parameter *AllowPinlessVoiceMailAccess* to `$true` on the UM mailbox policy and set the parameter *PinlessAccessToVoiceMailEnabled* to `$true` on the UM mailbox. By default, both parameters are set to `$false`, which requires an Outlook Voice Access user to enter their PIN when they access their voice mail.

Setting both parameters to `$true` allows you to enable PIN-less sign-ins to voice mail for a large group of users who are associated with a UM mailbox and also enable PIN-less sign-ins for a single UM mailbox or a subset of UM mailboxes. Even if you enable PIN-less sign-ins to voice mail for a group of UM-enabled users or a single UM-enabled user, when they access their email, calendar, personal contacts, the directory, or personal options, they'll be prompted to enter their PIN.

To enable PIN-less sign-ins to voice mail for a user, the following conditions must be met:

- You've run the following cmdlet on the UM mailbox policy: `Set-UMMailboxPolicy -id MyUMMailboxPolicy -AllowPinlessVoiceMailAccess $true`
- You've run the following cmdlet on the mailbox of the UM-enabled user: `Set-UMMailbox -id tonys@contoso.com -PinlessAccessToVoiceMailEnabled $true`
- The UM-enabled user is associated with the same UM mailbox policy for which you enabled PIN-less sign-ins.
- The UM-enabled user dials in to Outlook Voice Access from a phone number that's been assigned to them.
- You can only use the Shell to perform this procedure. To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell. To learn how to use Windows PowerShell to connect to Exchange Online, see **Connect to Exchange Online using remote PowerShell**.

For additional tasks related to UM mailbox policies, see UM mailbox policy procedures.

For additional tasks related to UM mailboxes, see Voice mail-enabled user procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.

- Before you perform these procedures, confirm that a UM dial plan has been created. For detailed steps, see Create a UM dial plan.
- Before you perform these procedures, confirm that a UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
- Before you perform these procedures, confirm that the user or users have been enabled for UM and voice mail. For detailed steps, see Enable a user for voice mail.

What do you want to do?

Use the Shell to enable PIN-less access to voice mail for UM-enabled users on a UM mailbox policy

This example enables PIN-less voice mail access on a UM mailbox policy named `MyUMMailboxPolicy` for users associated with the mailbox policy who dial in to Outlook Voice Access.

```
Set-UMMailboxPolicy -id MyUMMailboxPolicy -
AllowPinlessVoiceMailAccess $true
```

Use the Shell to enable PIN-less access to voice mail on a UM-enabled user's mailbox

This example enables PIN-less voice mail access for the user who dials in to Outlook Voice Access to reach the mailbox named `tonys@contoso.com`.

```
Set-UMMailbox -id tonys@contoso.com -
PinlessAccessToVoiceMailEnabled $true
```

Run reports for voice mail calls

Exchange Server 2013 > Unified Messaging >

Applies to: *Exchange Online*

Topic Last Modified: *2013-02-22*

Unified Messaging (UM) call reports provide information about the calls forwarded to or placed by UM. Use these reports to monitor, troubleshoot, and report on UM for your organization. You can

access Unified Messaging call statistic reports by using the Call Statistics tool and access call logs for UM-enabled users by using the User Call Logs tool.

The reports provide aggregated statistical information about calls for Exchange servers and calls for UM-enabled users in your organization. These reports:

- Give on-premises, hybrid, and online administrators the ability to gather statistics about the UM services and UM-enabled users in their organizations.
- Provide summaries from the data that's gathered. This data can be stored for 90 days and archived for up to two years to meet retention requirements.
- Verify the overall audio quality for incoming calls to Exchange servers that are deployed.
- Easily verify the availability of the voice mail system and UM services in the organization for a given period of time.
- Plan for Unified Messaging capacity for an on-premises or hybrid organization.
- Verify how UM services in an organization are used over a given period of time.

You can use the following topics to help you gather call statistics and reports and interpret those results to monitor and troubleshoot UM services in your organization:

- Review the voice mail calls in your organization Use the UM Call Statistics report to monitor the availability and audio quality of UM and to track usage for capacity planning.
- Review the voice mail calls for a user Use user call logs to see details about the calls for a user for the last 90 days.
- Investigate the audio quality of voice calls in your organization If your organization is experiencing problems with the audio quality of UM calls, use the audio quality details from the UM Call Statistics report to help you understand what's causing the problems.
- Investigate the audio quality of voice calls for a user If a user is experiencing problems with the audio quality of UM calls, use the audio quality details from the user call logs to help you understand what's causing the problems.
- Interpret voice mail call records Export more detailed data to diagnose problems with audio quality or rejected calls, and to provide information for audits or reports about your UM service.

UM reports procedures

Exchange Server 2013 > Unified Messaging > Run reports for voice mail calls >

Applies to: *Exchange Online*

Topic Last Modified: *2012-11-09*

Review the voice mail calls in your organization

Review the voice mail calls for a user

Investigate the audio quality of voice calls in your organization

Investigate the audio quality of voice calls for a user

Review the voice mail calls in your organization

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

You can use the Call Statistics report to view information about the type and status of incoming calls handled by the Exchange servers in your organization. The report provides statistical information about the calls forwarded to or placed by Unified Messaging (UM) for your organization. You can use this information to track usage for capacity planning, monitor and troubleshoot the availability and audio quality of UM, and to troubleshoot failed calls.

This topic answers these questions:

- How do I get call statistics for UM?
- How do I interpret UM call statistics?

For additional tasks related to UM reporting, see UM reports procedures.

How do I get call statistics for UM?

1. In the Exchange admin center (EAC), click **Unified messaging** > **More options** ··· > **Call statistics**.
2. Choose the information you want to include in the report. The report automatically updates as you select any of the following options:
   - **Show** Choose what type of call statistics to view:
     - **Daily (90 days)** Select Daily to see details for all calls in the past 90 days.
     - **Monthly (12 months)** Select Monthly to see a summary of calls by month for the last 12 months.
     - **All** Select All to see the combined statistics for all calls received since UM started handling calls.
   - **UM dial plan** If you want to limit the data in the report to only calls in a specific UM dial plan, select that dial plan.
   - **UM IP gateway** If you want to limit the data in the report to only calls in a specific UM IP gateway, select that gateway. If you select a UM dial plan first, only the UM IP gateways associated with the selected UM dial plan are available in the list.
3. To get more details about the audio quality for a row in the report, select the row and click

Audio Quality Details. For more information about how to interpret audio quality, see Investigate the audio quality of voice calls in your organization.

4. To copy the report to the Clipboard, click **Copy**.
5. For Daily reports, you can export the details for a specific day to a .csv file.
   - Select the day and click **Export day**.
   - In the **File Download** confirmation box, click **Open** or **Save**.

The exported file will be named um_cdr_*YYYY-MM-DD*.csv, where *YYYY-MM-DD* is the year, month, and day the report was run. For more information, see Interpret voice mail call records.

Note:
On the report page, you can download a Microsoft Excel template that you can use to import the .csv file for a specific day.

Return to top

How do I interpret UM call statistics?

The UM Call Statistics report includes the following information:

- **DATE** The UTC date for the call data. The date format depends on the type of report you've chosen and your locale settings. You can choose from the following options:
  - **---** All calls are shown.
  - **MMM/YY** The month of the calls. For example, Jan/13.
  - **MM/DD/YY** The day of the calls. For example, 6/23/13.
- **TOTAL** The total number of calls for the selected UM dial plan or UM IP gateway for that date.
- **VOICE MESSAGE** The percentage of incoming calls answered by UM on behalf of users in which callers left a voice message.
- **MISSED** The percentage of incoming calls answered by UM on behalf of users in which callers didn't leave a voice message, resulting in a missed call notification.
- **OUTLOOK VOICE ACCESS** The percentage of incoming calls where users signed in to UM (and were authenticated) to access their email messages, calendars, and voice messages.
- **OUTGOING** The percentage of calls that were placed or transferred by UM on behalf of authenticated or unauthenticated users. This statistic includes Find Me, Play on Phone, and Play on Phone Greetings call types.
- **AUTO ATTENDANT** The percentage of incoming calls that were answered by UM auto attendants.
- **FAX** The percentage of incoming calls that were redirected to a fax partner.
- **OTHER** The percentage of any other incoming or placed calls that do not fall in any of the above categories. These calls include calls made to Outlook Voice Access numbers where the users didn't sign in and weren't authenticated.
- **FAILED OR REJECTED** The percentage of calls that either failed or were rejected by UM. Note that failed calls aren't counted twice. For example, if a call to Outlook Voice Access fails, it is only counted as a Failed call, and not also as an Outlook Voice Access call.
- **AUDIO QUALITY** A graphical representation of the overall audio quality for the selected period

of time for the organization.

Return to top

For more information

Investigate the audio quality of voice calls in your organization

Interpret voice mail call records

Review the voice mail calls for a user

Unified Messaging > Run reports for voice mail calls > UM reports procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-22*

User call logs are used to view the following information about specific Unified Messaging (UM) users:

- Details about the UM calls for a user over the last 90 days.
- Audio quality of each call. Audio quality metrics might not be available for all calls, because the metrics depend on several factors, such as the type and length of the call.

For additional tasks related to UM reporting, see UM reports procedures.

How do I get call logs for a UM-enabled user?

1. In the Exchange Administration Center (EAC), select **Unified messaging** > **More options** ··· > **User call logs**.
2. Click **Select a user**, and then select the user you want data for.
3. To get more details about the audio quality for a row in the report, select the row and click **Audio Quality Details**. For more information about how to interpret audio quality, see Investigate the audio quality of voice calls for a user.
4. To copy the report to the Clipboard, click **Copy all rows to the clipboard.**

How do I interpret the UM user call log?

The user call log includes the following information for each call:

- **DATE AND TIME** The date and time of the call, in the time zone that the selected user has set in Microsoft Outlook Web App.
- **DURATION** How long the call lasted in minutes (MM) and seconds (SS), in the following format: MM:SS.
- **CALL TYPE** The type of call:

- **Call Answering** The call wasn't answered and was forwarded to the Mailbox servers, and the caller left a voice message.
- **Call Answering Missed Call** The call wasn't answered and was forwarded to the Mailbox servers, and the caller didn't leave a voice message.
- **Subscriber Access** A call was made to the subscriber access number. The caller signed in and was authenticated to UM with their extension and password to access email messages, calendars, and voice messages over the phone.
- **Auto Attendant** The call was answered by a UM auto attendant. These calls are typically calls in which the caller dialed your organization's main phone number.
- **Fax** A call was received in which a fax tone was detected. If you've configured fax partners, this call was sent to the partner.
- **PlayonPhone** A call was placed by UM because the user clicked the Play on Phone button in a voice message in Microsoft Outlook Web App or Outlook.
- **FindMe** An outbound call was placed by UM as a result of a Find Me rule in a call answering rule.
- **Unauthenticated Pilot Number** A call was placed to the Outlook Voice Access number. The caller didn't sign in and wasn't authenticated.
- **Greetings Recording** A call was placed by UM to record personal greetings for a user.
- **None** A call was placed but the type wasn't defined.

- **CALLING NUMBER** The phone number or SIP address of the caller.
- **CALLED NUMBER** The phone number or SIP address (for users in SIP dial plans, such as Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server users) of the intended recipient of the call.
- **UM IP GATEWAY** The UM IP gateway that took the call.
- **AUDIO QUALITY** The overall audio quality of the call. For more details about audio quality, select the row and click **Audio Quality Details**.

Investigate the audio quality of voice calls in your organization

Unified Messaging > Run reports for voice mail calls > UM reports procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

If your organization is experiencing problems with the audio quality of Unified Messaging (UM) calls and voice mail messages, use the Call Statistics report to help you understand what's causing the problems.

Note:

The audio quality of a call can be affected by factors that aren't covered in the reports. For example, if your Exchange servers are experiencing a heavy memory load or CPU load, users may report poor call quality, even though the reports show excellent audio quality.

For additional tasks related to call statistics see UM reports procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM call data and summary report cmdlets" entry in the Unified Messaging permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the EAC to get audio quality statistics for your organization

1. In the EAC, navigate to **Unified messaging** > **More options** ··· > **Call statistics**.
2. Choose the call statistics to include in the report. The report automatically updates as you select any of the following options.
   - **Show** Choose what type of call statistics to view:
     - **Daily (90 days)** Select Daily to see details for all calls in the past 90 days.
     - **Monthly (12 months)** Select Monthly to see a summary of calls by month for the last 12 months.
     - **All** Select All to see the combined statistics for all calls received since UM started handling calls.
   - **UM dial plan** If you want to limit the data in the report to only calls in a specific UM dial plan, select that dial plan.
   - **UM IP gateway** If you want to limit the data in the report to only calls in a specific UM IP gateway, select that UM IP gateway. If you select a UM dial plan first, only the UM IP gateways associated with the selected UM dial plan are available in the list.
3. To get more details about the audio quality for a row in the report, select the row and click **Audio Quality Details**. The following information is available:
   - **DATE AND TIME** The UTC date and time that the call statistics were captured.
   - **UM DIAL PLAN** The dial plan for the calls included in the statistics.
   - **UM IP GATEWAY** The UM IP gateway that took the calls included in the statistics.
   - **NMOS** The Network Mean Opinion Score (NMOS) for the call. The NMOS indicates how good the audio quality was on the call as a number on a scale from 1 to 5, with 5 being excellent.

Note:
The maximum NMOS possible for a call is dependent on the audio codec being used. The NMOS may not be available for very short calls that are less than 10 seconds long.

- **NMOS DEGRADATION** The amount of audio degradation of the NMOS from the top value possible for the audio codec being used. For example, if the NMOS degradation value for a call was 1.2 and the NMOS reported for the call was 3.3, the maximum NMOS for that particular call would be 4.5 (1.2 + 3.3).
- **JITTER** The average variation in the arrival of data packets for the call.
- **PACKET LOSS** The average percentage of data packet loss for the selected call. Packet loss is an indication of the reliability of the connection.
- **ROUND TRIP** The average round trip score, in milliseconds, for audio on the selected call. The round-trip score measures latency on the connection.
- **BURST LOSS DURATION** The average duration of packet loss during bursts of losses for the selected call.
- **NUMBER OF SAMPLES** The number of calls that were sampled to calculate the averages.

4. For detailed audio quality metrics for specific calls, see Investigate the audio quality of voice calls for a user.

Investigate the audio quality of voice calls for a user

Unified Messaging > Run reports for voice mail calls > UM reports procedures >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-21*

If a user reports problems with the audio quality of their Unified Messaging (UM) calls, you can use the User Call Logs report to help you understand what's causing the problems.

Note:
The audio quality of a call can be affected by factors that aren't covered in the reports. For example, if your Exchange servers are experiencing a heavy memory or CPU load, users may report poor call quality, even though the reports show excellent audio quality.

For additional tasks related to UM reports, see UM reports procedures

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM call data and summary report cmdlets" entry in the

Unified Messaging permissions topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the EAC to get call logs for a UM-enabled user

1. In the EAC, navigate to **Unified Messaging** > **More options** ··· > **User call logs**.
2. Click **Select a user**, and then select the user you want data for.
3. To get more details about the audio quality for a row in the report, select the row and click **Audio Quality Details**. The following information is available:
   - **DATE AND TIME** The date and time of the call, in the time zone that the selected user has set in Outlook Web App.
   - **USER** The selected user.
   - **UM DIAL PLAN** The dial plan for the call.
   - **UM IP GATEWAY** The UM IP gateway that was used for the call.
   - **AUDIO CODEC** The audio codec that was used during the call.
   - **NMOS** The Network Mean Opinion Score (NMOS) for the call. The NMOS indicates how good the audio quality was on the call as a number on a scale from 1 to 5, with 5 being excellent.

Note:

The maximum NMOS possible for a call depends on the audio codec being used. The NMOS may not be available for very short calls that are less than 10 seconds long.

   - **NMOS DEGRADATION** The amount of audio degradation of the NMOS from the top value possible for the audio codec being used. For example, if the NMOS degradation value for a call was 1.2 and the NMOS reported for the call was 3.3, the maximum NMOS for that particular call would be 4.5 (1.2 + 3.3).
   - **JITTER** The average variation in the arrival of data packets for the call.
   - **PACKET LOSS** The average percentage of data packet loss for the selected call. Packet loss is an indication of the reliability of the connection.
   - **ROUND TRIP** The average round trip score, in milliseconds, for audio on the selected call. The round-trip score measures latency on the connection.
   - **BURST LOSS DURATION** The average duration of packet loss during bursts of losses for the selected call.

Interpret voice mail call records

Applies to: *Exchange Online*

Topic Last Modified: *2013-02-22*

To view detailed information about calls handled by the Exchange servers on a specific day, export the call data for that day from the Call Statistics report. Daily call data, which is available for the past 90 days, can help you diagnose problems with audio quality or rejected calls, and provide information for audits or reports on Exchange servers in your organization.

For additional tasks related to UM reporting, see UM reports procedures.

Use the EAC to export daily UM call records

1. In the EAC, navigate to **Unified messaging** > **More options** ··· > **Call statistics**.
2. Under **Show**, click **Daily (90 days)**, and then choose the UM dial plan or UM IP gateway, or both, if you want. The report automatically updates as you choose options.
3. Select the day for which you want to export call records, and then click **Export day**.
4. In the **File Download** confirmation box, click **Open** or **Save**.

The exported file will be named um_cdr_*YYYY-MM-DD*.csv, where *YYYY-MM-DD* is the year, month, and day the report was run.

Note:
On the report page, you can download a Microsoft Excel template that you can use to import the .csv file for a specific day.

5. Use an application such as Excel to process the .csv file and build your own custom reports.

Interpret UM call data

The UM call data that you export includes the following detailed information about each call that UM handled on that day.

Note:
In the Call Statistics report, the days are in UTC time.

- **CallStartTime** The date and time that UM handled the call, in UTC. The UTC time and date is represented in the following format: *YYYY-MM-DD hh:mm:SSZ*, where *YYYY* = year, *MM* = month, *DD* = day, *hh* = hour, in 24-hour time, *mm* = minutes, ss = seconds. Z signifies Zulu, which is a way to denote UTC (like *+hh:mm* or *-hh:mm*, which gives the time offset from UTC). Because all call times in this report are in UTC time, this will always be Z.

For example, for a call placed on June 23, 2013 at 2:23pm, the call start time is shown as 2013-06-23 14:23:11Z.

- **Call Type** The type of call:
  - **Call Answering Voice Message** The call wasn't answered and was forwarded to the Exchange servers, and the caller left a voice message.
  - **Call Answering Missed Call** The call wasn't answered and was forwarded to the Exchange servers, and the caller didn't leave a voice message.

- **Subscriber Access** A call was made to the subscriber access number. The caller signed in and was authenticated to UM with their extension and password to access email messages, calendars, and voice messages over the phone.
- **Auto Attendant** The call was answered by a UM auto attendant. These calls are typically calls in which the caller dialed your organization's main phone number.
- **Fax** A call was received in which a fax tone was detected. If you've configured fax partners, this call was sent to the fax partner.
- **PlayOnPhone** A call was placed by UM because the user clicked the Play on Phone button in a voice message in either Microsoft Outlook Web App or Outlook.
- **Find Me** An outbound call was placed by UM as a result of a Find Me rule in a call answering rule.
- **Unauthenticated Pilot Number** A call was placed to the Outlook Voice Access number. The caller didn't sign in and wasn't authenticated.
- **Greetings Recording** A call was placed by UM to record personal greetings for a user.
- **None** A call was placed but the type wasn't defined.

- **CallIdentity** The SIP call identity, as provided by the UM IP gateway.
- **ParentCallIdentity** The SIP Session Identity of the session that originated this call. This box is used when using the Call Answering Rules Find Me feature or call transfer calls, including call transfers between UM auto attendants.
- **UMServerName** The name of the Mailbox server handling the call, if any. This information is provided only when you have an on-premises Mailbox server.
- **DialPlanName** The UM dial plan that handled the call.
- **Call Duration** The total duration of the call.
- **IPGatewayAddress** The fully qualified domain name (FQDN) of the IP gateway that handled the call.
- **CalledPhoneNumber** The phone number or SIP address of the intended recipient of the call (for users in SIP dial plans with Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server) .
- **CallerPhoneNumber** The phone number or SIP address of the caller.
- **OfferResult** The status of the call:
  - **Answer** UM successfully answered or placed a call. The call was neither transferred nor redirected. These calls include completed calls to Outlook Voice Access, Play on Phone, or UM auto attendants, and calls that UM handled when the called extension didn't answer the phone.
  - **Failed** UM accepted or placed a call, but the call failed. These calls include calls where the called number or address is busy, doesn't answer, or doesn't exist; where the caller hung up before the call was connected; where the UM dial plan or UM mailbox policy settings prevented the call; or where the VoIP gateway or IP PBX on your telephone system couldn't be reached.
  - **Rejected** UM rejected the call, usually because of a configuration error. These calls include calls where the UM IP gateway isn't associated with a UM dial plan, or where there are incompatibility issues.
  - **Redirected** UM accepted the call, but redirected it to another Mailbox server. These calls include calls where the caller used the UM menu to call a contact in the directory or personal

contacts, or where the caller called an Outlook Voice Access number using a phone number that isn't associated with the user's mailbox. In these cases, UM transfers the call to the Exchange server that's associated with that user's account.
  - **None** The call status is unknown.
- **DropCallReason** The reason the call was disconnected, if UM was able to determine the reason. For example, if the caller hung up, this shows Graceful Hangup.
- **ReasonForCall** How the call was connected:
  - **Direct** The caller dialed the called number directly.
  - **DivertForward** The caller dialed a number, and the person being called redirected the call to UM voice mail.
  - **DivertBusy** The caller dialed a number, and the phone was busy, so the call was redirected to UM voice mail.
  - **DivertNoAnswer** The caller dialed a number, and the person didn't answer, so the call was redirected to UM voice mail.
  - **Outbound** The call was placed by UM, for example, to play a voice message using Play on Phone.
  - **None** No reason was reported for the call.
- **DialedString** The address or phone number of the person to whom this call was either referred or transferred. This value also refers to the address or phone number called for Play on Phone calls.
- **CallerMailboxAlias** The mailbox alias (the portion of the email address that precedes the @ symbol) of the caller. This value is only available if the caller signed in to Outlook Voice Access.
- **CallerMailboxAlias** The mailbox alias of the intended recipient of the call, if the intended recipient is a UM-enabled user.
- **Auto Attendant Name** The name of the auto attendant related to this call.
- **NMOS Score** The Network Mean Opinion Score (NMOS) for the call. The NMOS indicates how good the audio quality was on the call as a number on a scale from 1 to 5, with 5 being excellent.

Note:
Note The maximum NMOS possible for a call depends on the audio codec being used. The NMOS may not be available for very short calls that are less than 10 seconds long.

- **NMOSDegradation** The amount of audio degradation of the NMOS from the top value possible for the audio codec being used. For example, if the NMOS degradation value for a call was 1.2 and the NMOS reported for the call was 3.3, the maximum NMOS for that particular call would be 4.5 (1.2 + 3.3).
- **NMOSDegradation Jitter** The total NMOS degradation due to jitter.
- **NMOSDegradation PacketLoss** The total NMOS degradation because of packet loss.
- **Jitter** The average variation in the arrival of data packets for the call.
- **PacketLoss** The average percentage of data packet loss for the selected call. Packet loss is an indication of the reliability of the connection.
- **Round Trip** The average round trip, in milliseconds, for audio on the selected call. The round-trip score measures latency on the connection.
- **BurstDensity** The percentage of packets lost and discarded within a burst (high loss rate) period.
- **Burst Gap duration** The average duration of packet loss during bursts of losses for the selected

call.
- **Audio Codec** The audio codec used during the call.

Test and troubleshoot voice mail

Exchange Server 2013 > Unified Messaging >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-06-25*

After you've installed Client Access and Mailbox servers in your organization, you may have to test the functionality of Unified Messaging (UM). The following are situations in which you will use the diagnostic tools and cmdlets included with Exchange 2013 or the UM Troubleshooting Tool:
- Troubleshooting errors and events.
- Testing the Client Access and Mailbox servers to make sure that the UM services are available and running.
- Testing whether VoIP gateways, PBXs, IP PBXs, SBCs, and Microsoft Lync Server are functioning correctly.

The following topics will help you test and troubleshoot Unified Messaging:
- Testing and troubleshooting with the UM Troubleshooting Tool
- Testing and troubleshooting with the Test-UMConnectivity cmdlet

Testing and troubleshooting with the UM Troubleshooting Tool

Exchange Server 2013 > Unified Messaging > Test and troubleshoot voice mail >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-06-26*

The Microsoft Exchange 2010 UM Troubleshooting Tool is an Exchange Management Shell cmdlet named **Test-ExchangeUMCallFlow**. You can use the cmdlet to diagnose configuration errors specific to call answering scenarios and to test whether voice mail is functioning correctly in both on-premises and cross-premises Microsoft Exchange Server 2010 Service Pack 1 (SP1) or later UM deployments. You can use this cmdlet in deployments with Microsoft Office Microsoft Lync Server 2010 or later or in UM deployments with Vo IP gateways, IP PBXs or session border controllers (SBCs).

Overview

This cmdlet emulates calls and runs a series of diagnostic tests that help on-premises administrators to identify configuration errors in telephony equipment, Exchange 2010 SP1 or later Unified Messaging settings, and connectivity issues between on-premises and cross-premises deployments of Exchange 2010 SP1 or later Unified Messaging.

When you run the cmdlet, it states the reason and possible solutions for issues that have been detected. It also outputs general audio quality metrics for diagnosing audio quality issues related to network connectivity, such as jitter and average packet loss. The **Test-ExchangeUMCallFlow** cmdlet supports testing UM components in `Secured`, `SIP Secured`, and `Unsecured` calls, and it can be run either in `Gateway` or `SIPClient` modes.

By default, when you're running the UM Troubleshooting Tool, it uses the credentials that are used when you log on to the computer. The credentials used are those that are specified for the calling party. You must set or specify the credentials to be used when you're running the UM Troubleshooting Tool in `SIPClient` mode. However, you don't need to set the credentials when running the UM Troubleshooting Tool in `Gateway` mode. If you will be using the UM Troubleshooting Tool in `SIPClient` mode, several other Office Communications Server 2007 R2 or Lync Server requirements and prerequisites must be met. For more information, see Checklist: Deploy Office Communications Server 2007 R2 and Exchange 2010 Unified Messaging or Checklist: Integrate Exchange 2013 UM with Lync Server.

Important:
The **Test-ExchangeUMCallFlow** cmdlet must be used to test only the voice mail functionality of a Microsoft Exchange Server 2010 Unified Messaging server that has Exchange 2010 Service Pack 1 (SP1) installed or Microsoft Exchange 2013.

The **Test-ExchangeUMCallFlow** cmdlet can be installed on a local Exchange 2010 Unified Messaging server, an Exchange 2013 Mailbox server, or on another 64-bit computer running:

- The Windows 7 or Windows 8 operating system
- The Windows Server 2008 or Windows Server 2008 R2 operating system
- The Windows Server 2012 or Windows Server 2012 R2 operating system

The **Test-ExchangeUMCallFlow** cmdlet requires the following components to be installed on a Windows 7, Windows 8, Windows Server 2008, or Windows Server 2012 64-bit computer before installing the cmdlet:

- Microsoft .NET Framework 3.5 Service Pack 1 (SP1). To download the service pack, see Microsoft .NET Framework 3.5 Service Pack 1.
- Microsoft .NET Framework 3.5 Family Update for Windows Vista x64 and Windows Server 2008 x64 updates if the tool will be run on a Windows Vista or Windows Server 2008 computer. To download the update, see Microsoft .NET Framework 3.5 Family Update for Windows Vista x64, and Windows Server 2008 x64.
- Windows Remote Management (WinRM) 2.0 and Windows PowerShell V2 (Windows6.0-KB968930.msu). For more information, see Microsoft Knowledge Base article 968930, Windows

Management Framework core package (Windows PowerShell 2.0 and WinRM 2.0).
- Unified Communications Managed AP1 2.0, Core Runtime (64-bit). To download the UcmaRuntimeWebDownloadX64.msi program file, see Unified Communications Managed API 2.0, Core Runtime (64-bit).

The **Test-ExchangeUMCallFlow** cmdlet isn't included on the Exchange 2010 SP1 DVD, the Exchange 2010 SP1-only download, or the Exchange 2013 installation media; however, you can download the cmdlet from the Microsoft Download Center.

For more information about syntax and parameters, see Test-ExchangeUMCallFlow.

Learn about the Exchange UM Troubleshooting Tool

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-06-27*

The Microsoft Exchange 2010 Unified Messaging Troubleshooting Tool is an Exchange Management Shell cmdlet named **Test-ExchangeUMCallFlow**. You can use this tool to conduct a series of diagnostic tests for Unified Messaging (UM) in your organization. If any of the tests fail, the tool reports the reason for the failure and possible solutions to fix the problem. You can only use the UM Troubleshooting Tool on Exchange 2010 or later servers.

The UM Troubleshooting Tool can be used to test whether voice mail is functioning correctly in both on-premises and cross-premises deployments. You can use this tool in UM deployments that include Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server 2010 or later, or in UM deployments that include VoIP gateways, IP Private Branch eXchanges (IP PBXs), or session border controllers (SBCs).

Note:

The UM Troubleshooting Tool is used for testing and troubleshooting. The **Test-UMConnectivity** cmdlet, on the other hand, should be used for monitoring. The **Test-UMConnectivity** cmdlet is used with System Center Operations Manager (SCOM) management packs that are used for monitoring Exchange 2010 UM servers or Exchange 2013 Client Access and Mailbox servers and the telephony components. The **Test-UMConnectivity** cmdlet performs local SIP tests and local logon tests to mailboxes, and can be run as an SCOM task.

To download the UM Troubleshooting Tool, see Unified Messaging Troubleshooting Tool.

Contents

Overview

The UM Troubleshooting Tool simplifies testing and troubleshooting in UM deployments. When the UM Troubleshooting Tool is run, it automatically generates a set of trace files that are stored in the C:\Users\%UserProfile%\AppData\Roaming\Microsoft Exchange 2010 UM Troubleshooting folder. The following trace files are generated by the tool:

- **UMTool_Collaboration** Includes RTC stack traces.
- **UMTool_DiagnosticLog** Lists all the tests that are run and their results.
- **UMTool_S4** Includes the S4: signaling stack traces.
- **UMTool_SIPMessageLogs** Includes the full SIP traces for the test call that's made.

The UM Troubleshooting Tool connects directly to an on-premises session border controller (SBC), if one exists, or connects to an SBC in a datacenter and emulates an incoming call as if the call was coming from a PBX through a VoIP gateway or an IP PBX. The UM Troubleshooting tool can be used to diagnose:

- Incorrect settings in on-premises or cross-premises UM deployments in which Office Communications Server 2007 R2 or Microsoft Lync Server is deployed.
- Incorrect settings on on-premises or cross-premises telephony equipment that includes VoIP gateways and PBXs or IP PBXs.
- Issues with Domain Name System (DNS).
- Certificate issues when you're using SIP secured or Secured UM dial plans.
- Signaling and media issues for DTMF (also known as touchtone) and audio.

If the UM Troubleshooting Tool detects a failure in your configuration, the tool reports the reason for the error and the possible solutions for the issues that have been detected. The errors that can be reported when the UM Troubleshooting Tool is used in an on-premises deployment include the following:

- The maximum call limit has been reached.
- The user isn't enabled for Unified Messaging.
- The UM IP gateway, dial plan, or hunt group information can't be located.
- The security type doesn't match the UM dial plan.
- There are no worker processes available to process the call.
- The UM service or UM Call Router services are stopped.
- The Active Directory forest couldn't be located.

- No disk space is available.
- Invalid SIP headers were used in the request.
- A call was made to an Office Communications Server 2007 R2 server or Lync Server server.
- The UM IP gateway is disabled.
- The URI for the user who is being called isn't valid.

When the UM Troubleshooting Tool is used in a cross-premises deployment, the errors that can be reported include the following:

- The user isn't enabled for Unified Messaging.
- The UM IP gateway is disabled.
- The URI for the user is invalid.
- The security type doesn't match the UM dial plan.
- Invalid SIP headers were used in the request.
- The UM IP gateway, dial plan, or hunt group information can't be located.

The UM Troubleshooting Tool sends a sample .wav file for 15 seconds. After the audio file and RTP audio stream are sent and played back, the tool reports general audio quality metrics for diagnosing audio quality issues related to network connectivity, such as jitter and average packet loss. These reports include the media stream quality to and from a UM server and contain the following:

- Network Mean Opinion Score (NMOS)
- Codec
- Latency in milliseconds (ms)
- Jitter in milliseconds (ms)
- % of packet loss
- The NMOS classification and rating that will be used to determine the audio quality will be:
  - NMOS less than 2 = Poor
  - NMOS greater than 2 but less than 3 = Average
  - NMOS greater than 3 but less than 4 = Good
  - NMOS greater than 4 but less than 5 = Excellent

The UM Troubleshooting Tool supports testing UM dial plans that use Secured, SIP Secured, and Unsecured calls. If you choose Secured or SIP Secured, the thumbprint of the certificate that's used is checked to determine whether the certificate is expired and the type of certificate that's used for TLS (Transport Layer Security) communications. The certificate is used to correctly identify and ensure the identity of the remote computer. When Secured or SIP Secured mode is selected, the UM Troubleshooting Tool verifies whether the following are true:

- The local certificate was found in the local computer store.
- The certificate being used is trusted.
- The target name specified in the certificate is valid.
- The certificate has expired.
- The remote computer trusts the certificate.
- The certificate has been revoked.
- The certificate doesn't have the required enhanced key usage.

The UM Troubleshooting Tool can be run in either Gateway or SIPClient mode, depending on whether Office Communications Server 2007 R2 or Lync Server is deployed or whether VoIP gateways and PBXs or IP PBXs are used with Unified Messaging servers. When either Gateway or SIPClient mode is used, the UM Troubleshooting Tool supports making calls using the following formats. The format that's used depends on the URI type of the UM dial plan:

- Telephone extension 425-555-1010
- E.164 phone numbers +1 (425) 555-1010
- SIP addresses tonysmith@contoso.com

When SIPClient mode is used, the UM Troubleshooting Tool makes a voice memo call. This is a call that doesn't ring a phone or a Unified Communications (UC) endpoint. Instead, it sends the call directly to voice mail. When the UM Troubleshooting Tool is run in SIPClient mode, it will determine:

- Which target user is being called.
- Whether the SIP call was established successfully.
- Whether the SIP call was accepted by an Exchange 2010 Unified Messaging server or Exchange 2013 Mailbox server.
- Whether the correct DTMF sequence was received.
- Whether the diagnostic .wav file was sent and received by an Exchange 2010 UM server or Exchange 2013 Mailbox server.
- The metrics that were used when the media or audio quality stream was received.

The UM Troubleshooting Tool emulates incoming calls and runs a series of diagnostic tests that help on-premises administrators and tenant administrators test call flow for call answering and identify configuration errors. Although the UM Troubleshooting Tool can be used in call answering scenarios, it can't be used to test the following types of calls:

- Outlook Voice Access calls, including calls that access voice mail, email, calendar, the directory, personal contacts, or personal options
- UM auto attendants
- Play on Phone
- Call Answering Rules
- Faxing
- Prompt provisioning

Return to top

UM troubleshooting architecture

The UM Troubleshooting Tool can help you troubleshoot, diagnose, and repair configuration issues in cross-premises deployments, and you can also use it in on-premises Unified Messaging deployments. In cross-premises deployments, the tool also validates onsite SBC configurations. The administrator can test all the Unified Messaging components that are used by Unified Messaging, including the SBCs.

VoIP gateway and IP PBX deployments

In the following example, Gateway mode is used to test call flow in an environment that doesn't include Office Communications Server 2007 R2 or Lync Server. This example tests the telephony equipment, including VoIP gateways, PBXs and IP PBXs, and the Unified Messaging components. This example sets the Voice over IP (VoIP) security mode to Unsecured, uses the IP address 10.1.1.1 as the next hop, and includes an extension number in the diversion information.

```
Test-ExchangeUMCallFlow -Mode Gateway -VoIPSecurity
Unsecured -NextHop 10.1.1.1 -Diversion 12345
```

Return to top

Office Communications Server 2007 R2 and Microsoft Lync Server deployments

The UM Troubleshooting Tool can be used in on-premises or cross-premises deployments that include Office Communications Server 2007 R2 or Microsoft Lync Server when SIPClient mode is set. The following example uses SIPClient mode and tests the call flow with a secured UM dial plan in an environment that contains Office Communications Server 2007 R2 or Lync Server servers. By default, when you run the UM Troubleshooting Tool, it uses the credentials of the user who is currently logged on to the computer. When you run the following example, you'll be prompted for the credentials you want to use when you run the UM Troubleshooting Tool. For details, see Set the credentials to use with the Exchange UM Troubleshooting Tool.

```
Test-ExchangeUMCallFlow -Mode SIPClient -VoIPSecurity
Secured -CallingParty tony@contoso.com -CalledParty
david@contoso.com -Credential $get
```

Installing the UM Troubleshooting Tool

The UM Troubleshooting Tool can be installed on a local Unified Messaging server or on another 64-bit computer running either:

- The Windows 7 or Windows 8 operating systems.
- The Windows Server 2008 or Windows Server 2008 R2 operating systems.
- The Windows Server 2012 or Windows Server 2012 R2 operating systems.

If you're using the UM Troubleshooting Tool on a 64-bit version of Windows 7, Windows 8, or the 64-bit edition of Windows Server 2008, the following components must be installed before you can install the UM Troubleshooting Tool:

- Microsoft .NET Framework 3.5 Service Pack 1 (SP1) See Microsoft .NET Framework 3.5 Service

Pack 1.

Note:
If the tool will be run on a Windows Vista or Windows Server 2008 computer, see Microsoft .NET Framework 3.5 Family Update for Windows Vista x64, and Windows Server 2008 x64.

- Windows Remote Management (WinRM) 2.0 and Windows PowerShell V2 (Windows6.0-KB968930.msu) See Microsoft Knowledge Base article 968930, Windows Management Framework Core package (Windows PowerShell 2.0 and WinRM 2.0).
- Microsoft Unified Communications Managed API 2.0 Core Runtime (UcmaRuntimeWebDownloadX64.msi) See Unified Communications Managed API 2.0, Core Runtime (64-bit).

The UM Troubleshooting Tool (**Test-ExchangeUMCallFlow** cmdlet) isn't included on the Exchange 2010 SP1 DVD, the download that only includes Exchange 2010, or the Exchange 2013 installation media. However you can download the UM Troubleshooting Tool from the Microsoft Download Center.

For details, see Install the Exchange UM Troubleshooting Tool.

Return to top

Cmdlet parameters

The following table includes the parameters you can use with the **Test-ExchangeUMCallFlow** cmdlet and descriptions of those parameters. You can also use the Shell command `Get-help Test-ExchangeUMCallFlow -detailed` to find detailed information about each parameter that can be used with the **Test-ExchangeUMCallFlow** cmdlet, along with usage examples.

Parameters

Parameter	Description
CalledParty	The *CalledParty* parameter specifies the SIP URI of the Office Communications Server 2007 R2 or Lync Server user who has been enabled for Enterprise Voice. This is the user who the **Test-ExchangeUMCallFlow** cmdlet will make the voice call to, for example: `-CalledParty tonysmith@contoso.com`. Use this parameter if you're running the tool in SIPClient mode.
CallingParty	The *CallingParty* parameter specifies the SIP URI of the Office Communications Server 2007 R2 or Lync Server user who has been enabled for

	Enterprise Voice. This is the user who's making the incoming call, for example: `-CallingParty tonysmith@contoso.com`. Use this parameter if you're running the tool in SIPClient mode.
Diversion	The *Diversion* parameter specifies the string that should be sent as diversion information for the incoming call. This can be in the form of a Diversion or History-Info header. The diversion information that is included in the incoming call can be an extension number or can include additional diversion information. When you provide diversion information as a History-Info header, verify the following: • There are at least two different entries with different user parts. • The last entry contains the pilot number of the user's associated UM dial plan. • The second-to-last entry includes a UM-enabled user's extension number. This entry must also include the appropriate Reason text. This text must be escaped correctly in accordance with standard URL parameter escaping rules.
Mode	The *Mode* parameter specifies whether the VoIP gateway, IP PBX, or Office Communications Server 2007 R2 or Lync Server mode is to be used. You can specify either Gateway mode when your UM deployment includes VoIP gateways or IP PBXs or SIPClient mode when your UM deployment includes Office Communications Server 2007 R2 or Lync Server.

NextHop	The *NextHop* parameter specifies the IP address or fully qualified domain name (FQDN) of the next hop and can also include the TCP port of the next hop that the **Test-ExchangeUMCallFlow** cmdlet must connect to while emulating the VoIP gateway or IP PBX. When you include the TCP port, you must specify either port 5060 for Unsecured mode or port 5061 for Secured or SIP Secured mode. For example: `gateway.contoso.com:5061`.
CertificateThumbprint	The *CertificateThumbprint* parameter specifies the thumbprint of the certificate used for TLS. This is required if either SIP Secured or Secured mode is configured on the UM dial plan. This certificate thumbprint is the certificate that was exported from the VoIP gateway, IP PBX, or SBC. Also, the computer that has the UM Troubleshooting Tool installed and is being used to test for call flow must trust the certificate of authority for the next hop.
Credential	The *Credential* parameter specifies the credentials that will be used to run the cmdlet.
HuntGroup	The *HuntGroup* parameter specifies the UM hunt group associated with the VoIP gateway that's being emulated. This is typically an extension number. Use this parameter if you're running the tool in Gateway mode.
VoIPSecurity	The *VoIPSecurity* parameter specifies the security mode when using the cmdlet in Gateway mode. You can use one of the following VoIP security modes:

	• Secured (TLS/SRTP) • Unsecured (TCP/RTP) (default) • SIP Secured (TLS/RTP)

Return to top

Run the Exchange UM Troubleshooting Tool on Windows 7 or Windows 8

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-06-27*

The Microsoft Exchange 2010 UM Troubleshooting Tool is an Exchange Management Shell cmdlet named **Test-ExchangeUMCallFlow**. You can use the cmdlet to diagnose configuration errors specific to call answering scenarios and to test whether voice mail is functioning correctly in both on-premises and cross-premises Microsoft Exchange Server 2010 Service Pack 1 (SP1) or later UM deployments. You can use this cmdlet in deployments with Microsoft Office Microsoft Lync Server 2010 or later or in UM deployments with Vo IP gateways, IP PBXs or session border controllers (SBCs).

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM server" or “UM services” entry in the Unified Messaging permissions topic.
- Make sure your Exchange 2010 or Exchange 2013 organization meets the following requirements:
  - A UM dial plan has been created. For detailed steps, see Create a UM dial plan.
  - A UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
  - A UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
  - An Exchange 2010 UM server has been added to UM dial plan. If you are using Exchange 2013 with Lync Server, add all Client Access and Mailbox servers to the SIP URI dial plans. For detailed steps, see Add a UM Server to a Dial Plan or Add Mailbox and Client Access servers to a SIP URI dial plan.
- If you're running the UM Troubleshooting Tool on a local UM server with Exchange 2010 SP1 or later or on an Exchange 2013 Mailbox server, you may not have to install all the prerequisites listed below. They may have already been installed along with the UM server role. However, if

you're installing the UM Troubleshooting Tool on a 64-bit computer other than a server that is running the UM server role, you will need to install the following components:
  - Microsoft .NET Framework 3.5 Service Pack 1 (SP1). See Microsoft .NET Framework 3.5 Service Pack 1.
  - If the tool will be run on a Windows Vista or Windows Server 2008 computer, see Microsoft .NET Framework 3.5 Family Update for Windows Vista x64, and Windows Server 2008 x64.
  - Windows Remote Management (WinRM) 2.0 and Windows PowerShell V2 (Windows6.0-KB968930.msu). See Microsoft Knowledge Base article 968930, Windows Management Framework Core package (Windows PowerShell 2.0 and WinRM 2.0).
  - Microsoft Unified Communications Managed API 2.0 Core Runtime (UcmaRuntimeWebDownloadX64.msi). See Unified Communications Managed API 2.0, Core Runtime (64-bit).
- Download and install the UM Troubleshooting Tool.
  - Download Unified Messaging Troubleshooting Tool from the Microsoft Download Center.
  - Install the tool. For details, see Install the Exchange UM Troubleshooting Tool.

Important:

If you will be using the UM Troubleshooting Tool in `SIPClient` mode, there are several other Office Communications Server 2007 R2 or Microsoft Lync Server requirements and prerequisites that must be met. For more information, see Checklist: Deploy Office Communications Server 2007 R2 and Exchange 2010 Unified Messaging.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Run the UM Troubleshooting Tool on Windows Vista, Windows 7, or Windows 8

1. Click **Start** > **All Programs** > **Accessories** > **Windows PowerShell**.
2. Right-click **Windows PowerShell**, and from the pop-up menu select **Run as administrator**.
3. At the Windows PowerShell command prompt, go to the folder where the UM Troubleshooting Tool was installed and run the following.

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -
psconsolefile .
\Microsoft.Exchange.UM.TroubleshootingToolsnapin.psc1 -
noexit -command ". '.
\Microsoft.Exchange.UM.TroubleshootingTool.ps1' "
```

4. If you're running the UM Troubleshooting Tool on Windows Vista, Windows 7, or Windows 8, at

the Windows PowerShell command prompt, run the following.

```
Set-ExecutionPolicy RemoteSigned
```

5. From the **Start** menu, open the **Microsoft Exchange 2010 UM Troubleshooting Tool**.
6. In the **Microsoft Exchange 2010 UM Troubleshooting Tool** window, at the prompt, type the following and press Enter.

```
$cred=Get-Credential
```

7. In the **Windows PowerShell Credential Request** window, type the domain\user name and password, and then click **OK**.
8. In the **Microsoft Exchange 2010 UM Troubleshooting Tool** window, specify the necessary cmdlet parameters to test for call flow. For example:

```
Test-ExchangeUMCallFlow -Mode SIPClient -CallingParty
tonysmith@contoso.com - CalledParty jamiestark@contoso.com
NextHop ocsfe.contoso.com -Credential $cred
```

Set the credentials to use with the Exchange UM Troubleshooting Tool

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-06-27*

The Microsoft Exchange 2010 UM Troubleshooting Tool is an Exchange Management Shell cmdlet named **Test-ExchangeUMCallFlow**. You can use the cmdlet to diagnose configuration errors specific to call answering scenarios and to test whether voice mail is functioning correctly in both on-premises and cross-premises Microsoft Exchange Server 2010 Service Pack 1 (SP1) or later UM deployments. You can use this cmdlet in deployments with Microsoft Office Microsoft Lync Server 2010 or later or in UM deployments with Vo IP gateways, IP PBXs or session border controllers (SBCs).

By default, when you're running the UM Troubleshooting Tool, it uses the credentials that are used when you log on to the computer. The credentials used are those that are specified for the calling party. You must set or specify the credentials to be used when you're running the UM Troubleshooting Tool in `SIPClient` mode. However, you don't need to set the credentials when running the UM Troubleshooting Tool in `Gateway` mode.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "UM server" or “UM services” entry in the Unified Messaging permissions topic.
- Make sure your Exchange 2010 or Exchange 2013 organization meets the following requirements:
  - A UM dial plan has been created. For detailed steps, see Create a UM dial plan.
  - A UM mailbox policy has been created. For detailed steps, see Create a UM mailbox policy.
  - A UM IP gateway has been created. For detailed steps, see Create a UM IP gateway.
  - An Exchange 2010 UM server has been added to the UM dial plan. If you’re using Exchange 2013 with Lync Server, add all Client Access and Mailbox servers to the SIP URI dial plans. For detailed steps, see Add a UM Server to a Dial Plan or Add Mailbox and Client Access servers to a SIP URI dial plan.
- Install the UM Troubleshooting Tool. For detailed steps, see Install the Exchange UM Troubleshooting Tool.

Important:

If you will be using the UM Troubleshooting Tool in `SIPClient` mode, there are several other Office Communications Server 2007 R2 or Microsoft Lync Server requirements and prerequisites. For more information, see Checklist: Deploy Office Communications Server 2007 R2 and Exchange 2010 Unified Messaging or Checklist: Integrate Exchange 2013 UM with Lync Server.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Set the credentials to use with the UM Troubleshooting Tool

1. From the **Start** menu, open the **Microsoft Exchange 2010 UM Troubleshooting Tool**.
2. In the **Microsoft Exchange 2010 UM Troubleshooting Tool** window, at the prompt, type the following and press Enter.

```
$cred=Get-Credential
```

3. In the **Windows PowerShell Credential Request** window, type the domain\user name and password, and then click **OK**.
4. In the **Microsoft Exchange 2010 UM Troubleshooting Tool** window, specify the necessary cmdlet parameters to test for call flow. For example:

```
Test-ExchangeUMCallFlow -Mode SIPClient -CallingParty
tonysmith@contoso.com - CalledParty jamiestark@contoso.com
NextHop ocsfe.contoso.com -Credential $cred
```

Install the Exchange UM Troubleshooting Tool

Unified Messaging > Test and troubleshoot voice mail > Testing and troubleshooting with the UM Troubleshooting Tool >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-06-25*

The Microsoft Exchange 2010 UM Troubleshooting Tool is an Exchange Management Shell cmdlet named **Test-ExchangeUMCallFlow**. You can use the cmdlet to diagnose configuration errors specific to call answering scenarios and to test whether voice mail is functioning correctly in both on-premises and cross-premises Microsoft Exchange Server 2010 Service Pack 1 (SP1) or later UM deployments. You can use this cmdlet in deployments with Microsoft Office Microsoft Lync Server 2010 or later or in UM deployments with Vo IP gateways, IP PBXs or session border controllers (SBCs).

The UM Troubleshooting tool can be installed on a local Unified Messaging server, an Exchange 2013 Mailbox server, or on another 64-bit computer.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes
- The UM Troubleshooting tool requires that the following components be installed on a computer running Windows Vista, Windows 7, Windows 8, or the 64-bit edition of Windows Server 2008 or Windows Server 2012 or later before the tool is installed:
  - Microsoft .NET Framework 3.5 Service Pack 1 (SP1) See Microsoft .NET Framework 3.5 Service Pack 1.
  - If the tool will be run on a Windows Vista or Windows Server 2008 computer, see Microsoft .NET Framework 3.5 Family Update for Windows Vista x64, and Windows Server 2008 x64.
  - Windows Remote Management (WinRM) 2.0 and Windows PowerShell V2 (Windows6.0-KB968930.msu). See Microsoft Knowledge Base article 968930, Windows Management Framework Core package (Windows PowerShell 2.0 and WinRM 2.0).
  - Microsoft Unified Communications Managed API 2.0 Core Runtime (UcmaRuntimeWebDownloadX64.msi). See Unified Communications Managed API 2.0, Core Runtime (64-bit).

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Install the UM Troubleshooting Tool

1. Download the Unified Messaging Troubleshooting Tool from the Microsoft Download Center, and then double-click the MicrosoftExchange2010UMTroubleshootingTool.msi installation folder.
2. On the **Welcome to the Microsoft Exchange 2010 UM Troubleshooting Tool Setup Wizard** page, click **Next**.
3. On the **End-User License Agreement** page, review the software license terms, and if you agree, click **I accept the terms in the license agreement** and then click **Next**.
4. On the **Select Installation Folder** page, verify the path to the installation folder and click **Next**.
5. On the **Confirm Installation** page, click **Next** to start installation.
6. On the **Installation Complete** page, click **Close**.

Testing and troubleshooting with the Test-UMConnectivity cmdlet

Exchange Server 2013 > Unified Messaging > Test and troubleshoot voice mail >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-06-25*

After you install the Client Access and Mailbox servers and configure Unified Messaging, you can use multiple diagnostic tests and a software-based telephone application to test telephony connectivity and the operation of the Unified Messaging services.

Test-UMConnectivity

The **Test-UMConnectivity** cmdlet can be used to check connectivity to Client Access and Mailbox servers in several ways, depending on the parameters used with the cmdlet. For testing Unified Messaging functionality, you can use these tests:

- **Local** The **Test-UMConnectivity** cmdlet verifies Voice over IP (VoIP) communication with the Mailbox servers running on the same local computer.
- **Local with TUILogon** The **Test-UMConnectivity** cmdlet tries to establish VoIP communication with the Mailbox server running on the same computer. If it connects, it tries to sign in to one or

more UM-enabled mailboxes by sending the extension number and PIN of the mailbox. If the –*TUILogon* parameter is supplied, the following parameter values must also be supplied with the appropriate information for the test mailbox:
  - *–Phone* This parameter must contain the extension number for the test mailbox.
  - *–PIN* This parameter must contain the PIN of the UM-enabled mailbox.
  - *–UMDialPlan* This parameter must contain the dial plan linked with the test mailbox.
- **Remote** The **Test-UMConnectivity** cmdlet tries to connect to a remote Client Access server by placing a call through a VoIP gateway. After it connects, it performs connectivity checks on the remote Client Access server and the media paths.

Note:

If you receive the following message, you should restart the Microsoft Exchange Unified Messaging service because it has stopped or is not responding: "The Test-UMConnectivity task encountered an error while trying to make a call. Details: Unable to establish a connection."

Test UM operation

Unified Messaging > Test and troubleshoot voice mail > Testing and troubleshooting with the Test-UMConnectivity cmdlet >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-06-25*

This topic explains how to use the Shell to test the operation of your voice mail system. When you perform the following procedure, the Mailbox server running the Microsoft Exchange Unified Messaging service initiates a diagnostic Session Initiation Protocol (SIP) call, and then returns a health state variable of UM services.

This diagnostic test can be run only on a local Mailbox server, and you can't test the operation of the Mailbox server using the EAC.

For additional management tasks related to Client Access and Mailbox servers, see UM services procedures.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox server (UM service)" and “Client Access Server (UM call router service)” entries in the Unified Messaging permissions topic.
- To perform the following procedures, you must log on to the Mailbox server by using an account that's a member of the local Administrators group.
- Verify that the Mailbox server is installed, either on the same computer as the Client Access server or on a separate computer.

- Verify that the Client Access server is installed, either on the same computer as the Mailbox server or on a separate computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to test the operation of the Unified Messaging services

This example performs connectivity and operational tests on the local Mailbox server, and then displays the Voice over IP (VoIP) connectivity information.

```
Test-UMConnectivity
```

This example tests the ability for a local Client Access server to listen for incoming unencrypted SIP requests on TCP port 5060.

```
Test-UMConnectivity -ListenPort 5060
```

This example tests the ability for a local Client Access server to listen for incoming encrypted SIP requests on TCP port 5061.

```
Test-UMConnectivity -ListenPort 5061
```

Note:
Use mode 1 when the `-UMIPGateway` parameter isn't specified.

Note:
You can set the `-Timeout` parameter with a value of less than 5 seconds. However, we recommend that you always configure this parameter with a value of 5 seconds or more.

Test UM connectivity to VoIP gateways and PBXs

Unified Messaging > Test and troubleshoot voice mail > Testing and troubleshooting with the Test-UMConnectivity cmdlet >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-06-25*

You can test the operation of Unified Messaging (UM) and related connected telephony equipment. When you perform the following procedure, the Client Access and Mailbox server tests the full end-to-end operation of the voice mail system. This includes the telephony components connected to the Client Access and Mailbox servers, including VoIP gateways, Private Branch eXchanges (PBXs), IP PBXs, and cabling.

For additional management tasks related to troubleshooting UM, see **Manage Voice Mail Services (Archive)**.

What do you need to know before you begin?

- Estimated time to complete: 3 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox server (UM service)" and “Client Access Server (UM call router service)” entries in the Unified Messaging permissions topic.
- To perform the following procedures, you must log on to the Mailbox server by using an account that's a member of the local Administrators group.
- Verify that the Mailbox server is installed, either on the same computer as the Client Access server or on a separate computer.
- Verify that the Client Access server is installed, either on the same computer as the Mailbox server or on a separate computer.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to test the operation of the Unified Messaging and telephony components

This example tests the ability of the UM IP gateway to listen on TCP port 5060 for incoming SIP requests.

```
Test-UMConnectivity -ListenPort 5060 -UMIPGateway
MyIPGateway
```

This example tests the ability of the local Mailbox server to use an unsecured TCP connection instead of a secured mutual TLS connection to place a call through a UM IP gateway named `MyUMIPGateway` by using the telephone number 56780.

```
Test-UMConnectivity -UMIPGateway MyUMIPGateway -Phone 56780
-Secured $false
```

This example tests the Outlook Voice Access number on a dial plan by using a SIP URI. This example can be used in an environment that includes Lync Server.

```
Test-UMConnectivity -UMIPGateway OCSGateway1 -Phone
"sip:SIPdialplan.contoso.com@contoso.com"
```

Note:

You can set the `-Timeout` parameter with a value of less than 5 seconds. However, we recommend that you always configure this parameter with a value of 5 seconds or more. Use mode 2 when the `UMIPGateway` parameter is specified in the command-line syntax.

UM and voice mail terminology

Exchange Server 2013 > Unified Messaging >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-22*

This topic contains the terms and definitions that are used with Unified Messaging.

audio codec

A digital encoding of an analog voice signal. Most audio codecs provide compression of the data, at the cost of some loss of fidelity when the data is recovered. Audio codecs vary in their perceived sound quality, the bandwidth that is required to use them, and the system requirements that are needed to do the encoding.

audio notes

Text-based notes that can be added to a voice mail message that has been received in Outlook or Outlook Web App.

auto attendant

A software system that answers calls, plays prompts or instructions, and then collects input from the caller as touchtones or speech. Auto attendants can direct a call to telephone numbers or named users or to entities (for example, departments) that the caller specifies, without intervention from a human operator.

Automatic Speech Recognition (ASR)

A technology that enables a computer to match human speech to a predefined set of words or phrases.

call answering

The process by which a caller interacts with a voice mail system if the number they originally called isn't answered. Typically, the system will play a greeting or other prompt, and allow the caller to record a voice message.

Call Answering Rules

A form of call answering in which the user for whom the call is being answered can specify rules to determine the behavior callers experience. The user can specify conditions to be evaluated, greetings, and choices to be provided to the caller, and actions (for example, transfer or leave a message) to be taken as a result of the caller's choice.

circuit-switched network

A network in which there exists a dedicated connection. A dedicated connection is a circuit or channel set up between two nodes so that they can communicate.

conditional call forwarding

A set of conditions that are chosen by a user to be used when they receive an incoming call. The call is redirected based on the conditions that are set.

Dial by Name

A feature that enables a caller to spell a person’s name using the keys on a telephone (ABC=2, DEF=3, etc.).

dial plan

For Unified Messaging, this is a set of telephony-capable endpoints that share a common numbering plan. The details of the plan are determined by the telephone system to which UM is connected. In the simplest case, this can be a private branch exchange (PBX) with its extensions, each with a unique, fixed-length number.

dialing rule group

Dialing rule groups are created to enable telephone numbers to be modified before they're sent to a traditional or SIP-enabled PBX or IP PBX for outgoing calls. Dialing rule groups may remove digits from or add digits to telephone numbers that are being used to place calls by a Unified Messaging server.

Each dialing rule group contains dialing rule entries that determine the types of in-country/region and international calls that users within a dialing rule group can make. Each dialing rule group must contain at least one dialing rule entry.

fax partner

UM fax partners provide applications or services that can accept calls handed off by UM when a fax tone is detected. The partner's product or service then receives the fax data, creates a message, and delivers it to the UM-enabled user as an email message with a .tif attachment. These messages will appear in the Fax search folder in Outlook and Outlook Web App.

hunt group

A set of extensions that are organized into a group, over which a traditional or SIP-enabled PBX or

IP PBX “hunts” to find an available extension. A hunt group is used to direct calls to identically capable endpoints or to an application, such as voice mail.

in-country/region number format

The in-country/region number format specifies how a user's telephone number should be dialed by Unified Messaging from one dial plan to a different dial plan that has the same country code. This is used by an auto attendant and when an Outlook Voice Access user searches and tries to call the user in the directory.

This entry consists of a number prefix and a variable number of characters (for example, 020xxxxxxx).

informational announcement

An audio message that is played when a caller first dials in to a voice mail system, which may describe some item of interest.

international access code

The prefix that is used to direct a call internationally. The international access code is 011 in the United States and 00 in much of the rest of the world.

international number format

The string of digits that is used to define how to dial someone from outside a specific country.

Internet Protocol Private Branch eXchange (IP PBX)

A telephone switch that natively supports voice over IP (VoIP). An IP PBX uses VoIP-based protocols to communicate with IP-based hosts such as VoIP telephones over a packet-switched network. Some IP PBXs can also support the use of traditional analog and digital phones.

matched name selection method

The mechanism used to help a caller differentiate between users with names that match the touchtone or speech input.

message waiting indicator

A signal that indicates the presence of one or more unread voice messages. For voice mail systems, this is often a lamp on the phone or a stutter dial tone.

Microsoft Exchange Unified Messaging Call Router service

A service that directs incoming calls for UM-enabled users to the Microsoft Exchange Unified Messaging service.

Microsoft Exchange Unified Messaging service

A service that implements Unified Messaging capabilities for UM-enabled users.

missed call notification

An email message that is sent to a UM-enabled user that indicates that someone called but did not leave a voice message.

national number prefix

A prefix that is used to direct a call as an in-country call. In the United States, this prefix is 1. In the United Kingdom and most of the rest of the world, this prefix is 0.

number mask

A set of numbers and wildcard characters that is used to determine the telephone number that the Mailbox server will dial. An "X" represents a single digit (0 … 9). An asterisk (*) represents any number of such digits.

numeric extension

A string of digits that doesn't contain a "+" or a country/region code. In dial plans, extensions are required to have a specified length.

outdialing

A process in which Unified Messaging (UM) dials or transfers calls. UM generally receives calls, but sometimes dials calls. For example, outdialing occurs when a UM auto attendant transfers a call to a user's extension, or when a UM-enabled user uses Play on Phone from Outlook.

Outlook Voice Access

A series of voice prompts that allows authenticated callers to access their email, voice mail, calendar, and contact information using a standard analog, digital, or mobile telephone. Outlook Voice Access also enables authenticated callers to navigate their personal information in their mailbox, place calls, locate users, and navigate the system prompts and menus using DTMF, also known as touchtone, or voice inputs.

outside line access code

The prefix that is used by UM (or a person using an internal extension on the PBX or IP PBX) to access an outside line. This prefix is typically 9.

packet switching

A technique that divides a data message into smaller units called packets. Packets are sent to their destination by the best route available, and then they are reassembled at the receiving end.

pilot identifier

A telephone number that points to a hunt group and is the access number for calls that are routed to Unified Messaging. This is also sometimes called a pilot number.

PIN

A passcode that a user enters on the telephone to access their mailbox.

Play on Phone

A Unified Messaging feature that users can use to play their voice messages or play and record personalized voice mail greetings over a telephone.

Private Branch eXchange (PBX)

A private telephone network in an organization. Individual telephone numbers or extension numbers are supported, and calls are automatically routed to them. Users can call each other using

extensions, even across distributed locations.

prompt

An audio message played over the telephone to explain valid options to users.

Protected Voice Mail

A UM feature that uses information rights management to encrypt the contents of voice messages and specify the operations permitted on them. Protection can be caused by caller action (marking the message as private), or by system policy.

public switched telephone network (PSTN)

PSTN is a grouping of the world's public circuit-switched telephone networks. This grouping resembles the way that the Internet is a grouping of the world's public IP-based packet-switched networks.

reset

When a PIN or a password is reset, the system randomly chooses a new, temporary PIN or password. The user is required to change the temporary PIN the next time that they sign in to Outlook Voice Access.

reverse number lookup (RNL)

A method used to try to locate the name of a person, from a directory or other information store, based on a telephone number.

RTAudio codec

An advanced speech codec that is designed for real-time two-way VoIP applications such as gaming, audio conferencing, and wireless applications over IP. RTAudio is the preferred Microsoft audio codec and is the default codec for Microsoft Lync Server platforms.

SIP-enabled PBX

A SIP-enabled PBX is a telephony device that acts as a networking switch for switching calls in a telephony or circuit-switched network. However, the difference between a SIP-enabled PBX and a traditional PBX is that the SIP-enabled PBX can connect to the Internet and use the SIP protocol to make calls over the Internet.

SIP notification

A SIP notification is a SIP message sent from one SIP peer to another to advise it of a change.

SIP peer

A SIP-enabled device that provides telephony communications between a VoIP gateway, IP PBX, SIP-enabled PBX, Microsoft Lync servers, or VoIP phones and Unified Messaging services.

star out

An action a caller can perform when they are dialed in to a Unified Messaging auto attendant but they want to be able to get to Outlook Voice Access to get their email and voice mail. To do this, they press the star (*) key while the auto attendant prompts are being played.

subscriber access number (Outlook Voice Access number)

A number that is configured in a traditional or SIP-enabled PBX or IP PBX and on a UM dial plan that allows users to access their mailbox using Outlook Voice Access. In some cases, this may be configured to be the same number as the subscriber access number or pilot number (also called a pilot identifier) on the traditional or SIP-enabled PBX or IP PBX and the UM hunt group.

system prompt

A short audio recording for Unified Messaging, which is played to callers by the server. System prompts are used to welcome callers and to inform them of their options when they use the voice mail system.

telephone user interface (TUI)

An interface that is used to navigate the menus of a voice mail system using DTMF, also known as touchtone, inputs.

Text-to-Speech (TTS)

Technologies for translating or converting typewritten text into speech.

UM IP gateway

(See IP gateway.) A UM IP gateway is the Exchange Unified Messaging representation of any SIP peer with which it can communicate using VoIP protocols. It may represent a device that interfaces with a traditional or SIP-enabled PBX, an IP PBX, or Microsoft Lync Server.

UM worker process

A process that's created during the startup of the Microsoft Exchange Unified Messaging service. The UM service, on receiving a request to handle an incoming call, immediately redirects the request to a UM worker process, which carries out all subsequent interactions with the caller.

UM Worker Process Manager

A component that handles the creation and monitoring of all the UM worker processes that are created.

Unified Messaging

An application that consolidates a user's voice mail and email into one mailbox, so that the user only needs to check a single location for messages, regardless of type. The email server is used as the platform for all types of messages, making it unnecessary to maintain separate voice mail and email infrastructures.

voice mail

A system that records and stores telephone messages in a user mailbox.

Voice Mail Preview

A feature that provides text, transcribed from the audio recording, on a voice message when it is delivered.

voice message

An electronic message with a primary content of digitized audio.

Voice over IP (VoIP)

The practice of using an IP data network to transmit voice calls.

voice user interface (VUI)

An interface that is used to navigate the menus of a voice mail system using speech inputs.

VoIP gateway

1. A third-party hardware device or product that connects a legacy PBX to a LAN. A VoIP gateway translates or converts TDM or telephony circuit-switched protocols to packet-switched protocols that can be used on a VoIP-based network.
2. The Exchange Unified Messaging representation of any SIP peer with which it can communicate using VoIP protocols. It may represent a device that interfaces with a legacy PBX, an IP PBX, or Microsoft Lync Server.

welcome greeting

A greeting that is played when an external caller calls in to a UM auto attendant or when an Outlook Voice Access user or another caller calls a subscriber access number that is configured on a UM dial plan. The default welcome greetings can be changed by a customer to make them specific to an organization or location.

Mailbox and Client Access servers

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-15*

In Microsoft Exchange Server 2013, the five server roles from Exchange 2010 have been replaced by two main types of servers: Mailbox servers and Client Access servers. The Client Access server is a thin, stateless server that serves as a proxy for client connections to the Mailbox server. The Mailbox server handles the processing for all client connections to the active mailbox database.

Mailbox and Client Access server documentation

The following table contains links to topics that will help you learn about and manage Mailbox and Client Access servers in Exchange 2013.

Topic	Description
Mailbox server	Learn about the Mailbox server, which is responsible for client processing and houses an

	active or passive copy of the mailbox database.
Manage mailbox databases in Exchange 2013	Learn about the management tasks you can perform on a mailbox database after you create at least one database on a Mailbox server.
Client Access server	Learn about the Client Access server, which provides proxying and limited redirection for incoming client connections to the Mailbox server.
Mailbox import and export requests	Learn about importing and exporting information into mailboxes in your Exchange organization.

Mailbox server

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-19*

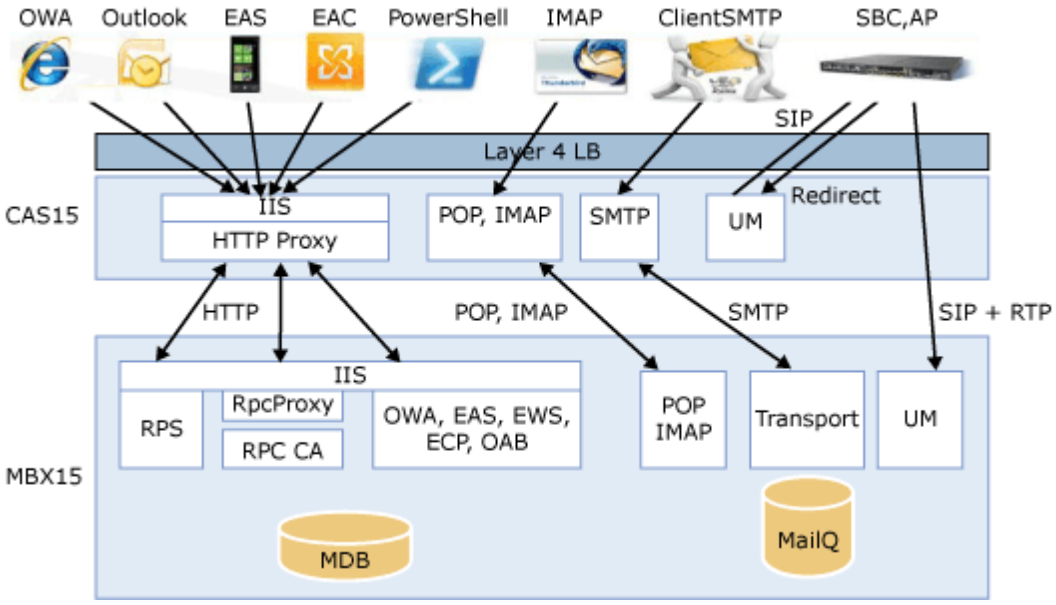
In Microsoft Exchange Server 2010, the Mailbox server role hosted both mailbox and public folder databases and also provided email message storage. Now, in Exchange Server 2013, the Mailbox server role also includes the Client Access protocols, Transport service, mailbox databases, and Unified Messaging components.

In Exchange 2013, the Mailbox server role interacts directly with Active Directory, the Client Access server, and Microsoft Outlook clients in the following process:

- The Mailbox server uses LDAP to access recipient, server, and organization configuration information from Active Directory.
- The Client Access server sends requests from clients to the Mailbox server and returns data from the Mailbox server to the clients. The Client Access server also accesses online address book (OAB) files on the Mailbox server through NetBIOS file sharing. The Client Access server sends messages, free/busy data, client profile settings, and OAB data between the client and the Mailbox server.
- Outlook clients inside your firewall access the Client Access server to send and retrieve messages. Outlook clients outside the firewall can access the Client Access server by using Outlook Anywhere (which uses the RPC over HTTP Proxy component).

- Public folder mailboxes are accessible via RPC over HTTP, regardless of whether the client is outside or inside the firewall.
- The administrator-only computer retrieves Active Directory topology information from the Microsoft Exchange Active Directory Topology service. It also retrieves email address policy information and address list information.
- The Client Access server uses LDAP or Name Service Provider Interface (NSPI) to contact the Active Directory server and retrieve users' Active Directory information.

Mailbox and Client Access server interaction and architecture



For more details, see the “Exchange 2013 architecture” section in What's new in Exchange 2013.

New Mailbox features

The following list briefly describes some new and some improved features in the Mailbox role for Exchange 2013:

- Evolution of the Exchange 2010 database availability group (DAG):
  - Transaction log code has been refactored for fast failover with deep checkpoint on passive database copies.
  - To support enhanced site resiliency, servers can be in different locations.
- Exchange 2013 now hosts some Client Access components, the Transport components, and the Unified Messaging components.
- The Exchange Store has been re-written in managed code to improve performance in additional I/O reduction and reliability.
- Each Exchange 2013 database now runs under its own process.
- Smart Search has replaced the Exchange 2010 multi-mailbox search infrastructure.

Securing Mailbox servers

By default, HTTP, Microsoft Exchange ActiveSync, POP3, and IMAP4 communication between the Mailbox servers and other Exchange server roles, domain controllers, and global catalog servers is

encrypted. In addition, make sure that your Mailbox servers aren't accessible to the Internet.

For more information

Unified Messaging

Mail flow

High availability and site resilience

Messaging policy and compliance

Mailbox moves in Exchange 2013

Manage mailbox databases in Exchange 2013

Messaging policy and compliance

Recipients

Collaboration

Managed Store

Exchange Server 2013 > Mailbox and Client Access servers > Mailbox server >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-14*

All previous versions of Exchange Server, from Exchange Server 4.0 to Exchange Server 2010, have supported running a single instance of the Information Store process (Store.exe) on the Mailbox server role. This single Store instance hosts all databases on the server: active, passive, lagged, and recovery. In the previous Exchange architectures, there is little, if any, isolation between the different databases hosted on a Mailbox server. An issue with a single mailbox database has the potential to negatively affect all other databases, and crashes resulting from a mailbox corruption can affect service for all users whose databases are hosted on that server.

Another challenge with a single Store instance in previous versions of Exchange is that the Extensible Storage Engine (ESE) scales well to 8-12 processor cores, but beyond that, cross-processor communication and cache synchronization issues lead to negative scale. Given today's much larger servers, with 16+ core systems available, this would mean impose the administrative challenge of managing the affinity of 8-12 cores for ESE and using the other cores for non-Store processes (for example, Assistants, Search Foundation, Managed Availability, etc.). Moreover, the previous architecture restricted scale-up for the Store process.

The Store.exe process has evolved considerably throughout the years as Exchange Server itself evolved, but as a single process, ultimately its scalability is limited, and it represents a single point

of failure. Because of these limits, Store.exe is gone in Exchange 2013 and replaced by the Managed Store.

Managed Store

The Managed Store is the name for the Information Store (aka the Store) processes in Exchange Server 2013. The Managed Store uses a controller/worker process model that provides storage process isolation and faster database failover. The Managed Store also includes a new static database caching mechanism that replaces the dynamic buffer algorithm in previous versions of Exchange Server. In the multi-process model used by the Managed Store, there is a single store service controller process (in this case, Microsoft.Exchange.Store.Service.exe aka MSExchangeIS), and one worker process (in this case, Microsoft.Exchange.Store.Worker.exe) for each mounted database. When a database is mounted, a new worker process is instantiated that services only that database. When a database is dismounted, the worker process for that database is terminated.

For example, if you have 40 databases mounted on a server, there will be 41 processes running for the Managed Store, one for each database, and one for the store service process controller.

The store service process controller is very thin and very reliable, but if it dies or is terminated, all of its worker processes die (they will detect the service controller process is gone and exit). The store process controller monitors the health of all store worker processes on the server. A forcible or unexpected termination the Microsoft.Exchange.Store.Service.exe causes an immediate failover of all active database copies. The Managed Store is also tightly integrated with the Microsoft Exchange Replication service (MSExchangeRepl.exe) and Active Manager. The controller process, worker processes, and Replication service work together to provide greater availability and reliability:

- Microsoft Exchange Replication service process (MSExchangeRepl.exe)
  - Responsible for issuing mount and dismount operations to the Store
  - Initiates recovery action on storage or database failures reported by the Store, the Extensible Storage Engine (ESE), and Managed Availability responders
  - Detects unexpected database failures
  - Provides the administrative interface for management tasks
- Store service process/controller (Microsoft.Exchange.Store.Service.exe)
  - Manages each worker process lifetime based on the mount and dismount operations received from the Replication service
  - Handles incoming requests from the Windows Service Control Manager
  - Logs failure items when store worker process problems detected (for example, hang or unexpected exit)
  - Terminates store worker processes in response failover event
- Store worker process (Microsoft.Exchange.Store.Worker.exe)
  - Responsible for executing RPC operations for mailboxes on a database
  - RPC endpoint instance within worker process is the database GUID
  - Provides database cache for a database

Static Database Caching Algorithm

The database caching algorithm known as dynamic buffer allocation that was introduced in Exchange Server 5.5 and also used by the Information Store in Exchange 2000 Server, Exchange Server 2003, Exchange Server 2007 and Exchange Server 2010, is also gone from Exchange 2013. Exchange 2013 uses a very simple and straightforward algorithm for determining database cache. The Managed Store no longer dynamically reallocates cache between databases when failover occurs, which greatly simplifies internal cache management. Instead, the memory allocated for each database cache (e.g., each store worker process) is based on number of local database copies and value of *MaximumActiveDatabases*, if configured. If the value of *MaximumActiveDatabases* is greater than number of current database copies, then the cache calculation is based on the number of database copies.

The static algorithm used by Exchange 2013 allocates memory for each store worker process' ESE cache based on physical RAM. This is referred to as a database's *Max Cache Target*. 25% of total server memory is allocated to the ESE cache. This is referred to as the *Server Cache Size Target*.

Note:

The Server Cache Size Target, and therefore the amount of memory allocated to the Store for ESE cache, can be overridden using *msExchESEParamCacheSizeMax* attribute of the *InformationStore* object in Active Directory (the value configured is the number of 32 KB pages to allocate across all store processes).

A static amount of this cache is allocated to active and passive copies. The store worker process will be allocated the Max Cache Target only when servicing an active database copy. Passive database copies are allocated 20 percent of the Max Cache Target. The remainder is reserved by the Store, and allocated to the worker process if the database transitions from passive to active.

Max Cache Target is calculated only at Store startup. Therefore, if you add or remove databases or database copies, you must restart the Store controller service (MSExchangeIS) so that the cache can be adjusted accordingly. If the service is not restarted, then newly created databases will have a smaller cache size target than databases created prior to the service startup. In this event, the sum of database cache size targets will likely exceed the Server Cache Size Target until MSExchangeIS is restarted.

Example Database Cache Calculations

Below are example database caching calculations that are based on a Mailbox server's memory and database configuration.

Example 1

In this example, the Mailbox server has 48 GB of memory, and it hosts two active databases and two passive databases. In addition, the *MaximumActiveDatabases* parameter is not configured. In this configuration, the amount of database cache is 3 GB for each active database copy worker

process and 0.6 GB for each passive database copy worker process. Here's how these values were obtained.

To get the Server Cache Size Target, multiply the amount memory by 25%:

48 GB X 25% = 12 GB

To get the Database Max Cache Target, divide the Server Cache Size Target by the total number of active and passive databases:

12 GB / 4 databases = 3 GB

To determine the amount of memory used for the passive database copies, multiply the Database Max Cache Target by 20%:

3 GB X 20% = 0.6 GB

Out of the 12 GB of memory assigned to the Server Cache Size Target, 7.2 GB will be in use by database worker processes, and 4.8 GB will be reserved by the Information Store for the two passive database copies in case they become active copies. In that event, they will use their Max Cache Target of 3 GB.

Example 2

In this example, the Mailbox server also has 48 GB of memory and hosts two active databases and two passive databases; however, the *MaximumActiveDatabases* parameter is configured with a value of 2. In this configuration, the amount of database cache is 5 GB for each active database copy worker process and 0.2 GB for each passive database copy worker process. Here's how these values were obtained.

To get the Server Cache Size Target, multiply the amount memory by 25%:

48 GB X 25% = 12 GB

To get the Database Max Cache Target, divide the Server Cache Size Target by the total number of active database plus the total number of passive databases multiplied by 20%:

12 GB / (2A + (2P X 20%)) = 5 GB

To determine the amount of memory used for the passive database copies, multiply the Database Max Cache Target by 20%:

5 GB X 20% = 1 GB

Out of the 12 GB of memory assigned to the Server Cache Size Target, 12 GB will be in use by database worker processes, and no memory will be reserved by the Information Store for the two passive database copies because they cannot become active copies in this configuration (because *MaximumActiveDatabases* is configured with a value of 2, and there are already 2 active database copies on the server).

Manage mailbox databases in Exchange 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-29*

A mailbox database is a unit of granularity where mailboxes are created and stored. A mailbox database is stored as an Exchange database (.edb) file. In Microsoft Exchange Server 2013, each mailbox database has its own properties that you can configure.

This topic shows you how to perform configuration tasks related to managing your mailbox databases in Microsoft Exchange Server 2013.

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox databases" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a mailbox database

Use the EAC to create a mailbox database

1. From the Exchange admin center, navigate to **Servers**.
2. Select **Databases**, and then click the **+** symbol to create a database.
3. Use the new database wizard to create your database.

Use the Shell to create a mailbox database

For an example of how to create a mailbox database, see Example 1 in New-MailboxDatabase.

How do you know this worked?

To verify that you have successfully created a database, do the following:

- From the EAC, verify that the mailbox database you created is listed in the **Databases** page.
- From the Shell, verify that the database was created on server Mailbox01 by running the following command.

```
Get-MailboxDatabase -Server "Mailbox01"
```

Get mailbox database properties

For detailed syntax and parameter information, see Get-MailboxDatabase.

Use the Shell to get mailbox database properties

For an example of how to get mailbox database properties, see Example 2 in New-MailboxDatabase.

How do you know this worked?

To verify that you have successfully retrieved your mailbox database information, do the following:

From the Shell, verify that all your mailbox database information is represented correctly.

Set mailbox database properties

Use the EAC to set mailbox database properties

1. From the EAC, navigate to **Servers**.
2. Select **Databases**, and then click to select the mailbox database you want to configure.
3. Click **Edit** to configure the attributes of a mailbox database.
4. Use the **General** tab to view status about the mailbox database, including the mailbox database path, last backup, and mailbox database status:
   - **Database path** This read-only field displays the full path to the Exchange 2013 database (.edb) file for the selected mailbox database. To view the entire path, you may have to click the path and use the Right Arrow key. You can't use this field to change the path. To change the location of the database files, use the Move-DatabasePath cmdlet.
   - **Last full backup** This read-only field displays the date and time of the last complete backup of the mailbox database.
   - **Last incremental backup** This read-only field displays the date and time of the last incremental backup of the mailbox database.
   - **Status** This read-only field displays whether the mailbox database is mounted or dismounted.
   - **Mounted on server** This read-only field displays which server the database is mounted on.
   - **Master** This read-only field displays the master server for the mailbox database. The Mailbox server that hosts the active copy of a database is referred to as the mailbox database master.
   - **Master type** This read-only field displays the type of mailbox database master.

- **Modified** This read-only field displays the date and time the database was last modified.
- **Servers hosting a copy of this database** This read-only field displays the other servers that have a copy of this database.

5. Use the **Maintenance** tab to configure mailbox database settings, including specifying a journal recipient, setting a maintenance schedule, and mounting the database at startup:
   - **Journal Recipient** Click **Browse** to specify a recipient to enable journaling on this mailbox database. Remove the recipient listed to disable journaling.
   - **Maintenance schedule** Use this list to select one of the preset maintenance schedules. You can also configure a custom schedule. To configure a custom schedule, click **Customize**.
   - **Enable background database maintenance (24 x 7 ESE scanning)** Select this check box to enable online database scanning, which runs continuously in the background. Online database scanning performs a checksum calculation of the database and performs operations that allow Exchange to scan for lost space on the database and recover it. If you select this check box, Exchange scans the database no more than one time per day and will issue a warning event if it can't finish scanning the database in a seven-day period.
   - **Don't mount this database at startup** Select this check box to prevent Exchange from mounting this mailbox database when it starts.
   - **This database can be overwritten by a restore** Select this check box to allow the mailbox database to be overwritten during a restore process.
   - **Enable circular logging** Select this check box to enable circular logging.
6. Use the **Limits** tab to specify the storage limits, the warning message interval, and the deletion settings for a mailbox database:
   - **Issue warning at (GB)** Select this check box to automatically warn mailbox users that their mailbox is approaching its storage limit. To specify the storage limit, select the check box, and then specify in gigabytes (GB) how much content can be stored in the mailbox before a warning email message is sent to the mailbox users. You can enter a value from 0 through 2,097,151 megabytes (MB) (2.0 terabytes).
   - **Prohibit send at (GB)** Select this check box to prevent users from sending new email messages after the size of their mailbox reaches the specified limit. To specify this limit, select the check box, and then type the size of the mailbox in GB at which you want to prohibit the sending of new email messages and notify the user. You can enter a value from 0 through 2,097,151 MB (2.0 terabytes).
   - **Prohibit send and receive at (GB)** Select this check box to prevent users from sending and receiving email messages after their mailbox size reaches the specified limit. To specify this limit, select the check box, and then type the size of the mailbox in GB at which you want to prohibit the sending and receiving of email messages and notify the user. You can enter a value from 0 through 2,097,151 MB (2.0 terabytes).
   - **Keep deleted items for (days)** Select this check box to set the number of days that deleted items are retained in a mailbox. You can enter a value from 0 through 24,855 days.
   - **Keep deleted mailboxes for (days)** Select this check box to set the number of days that deleted mailboxes are retained. You can enter a value from 0 through 24,855 days.
   - **Don't permanently delete items until the database has been backed up** Select this check

box to prevent mailboxes and email messages from being deleted until after the mailbox database has been backed up.

7. Use the **Client Settings** tab to select the offline address book (OAB) for the mailbox:
   - **Offline address book** To select an offline address book, click **Browse**, and then select the offline address book.

Use the Shell to set mailbox database properties

For an example of how to set mailbox database properties, see Example 1 in Set-MailboxDatabase.

How do you know this worked?

To verify that you have successfully set the attributes, do the following:

- Verify that your changes are saved in the EAC.
- From the Shell, run the following command to retrieve mailbox database properties.

```
Get-MailboxDatabase -Identity MailboxDatabase01 -Status | Format-List
```

Move a mailbox database path

For detailed syntax and parameter information, see Move-DatabasePath.

Use the Shell to move a mailbox database path

For an example of how to set mailbox database properties, see Example 1 in Move-DatabasePath.

How do you know this worked?

To verify that you have successfully moved the database path, do the following:

1. From the EAC, select **Servers** > **Databases**, and then click to select the appropriate mailbox.
2. Click the **pen** symbol and verify that the database path is correct.

Mount a mailbox database

For detailed syntax and parameter information, see Mount-Database.

Use the Shell to mount a mailbox database

For an example of how to mount a mailbox database, see Example 1 in Mount-Database.

How do you know this worked?

To verify that you have successfully mounted the mailbox database, do the following.

- From the Shell, run the following command to retrieve mailbox database properties for all mailbox databases.

```
Get-MailboxDatabase -IncludePreExchange2013
```

Dismount a mailbox database

For detailed syntax and parameter information, see Dismount-Database.

Use the Shell to dismount a mailbox database

For an example of how to dismount a mailbox database, see Example 1 in Dismount-Database.

How do you know this worked?

To verify that you have successfully dismounted the database, do the following:
1. From EAC, select **Servers** > **Databases**, and then click to select the appropriate mailbox.
2. Click the **pen** symbol, and verify that the database status is **Dismounted**.

Remove a mailbox database

Use the EAC to remove a mailbox database

1. From the EAC, select **Servers** > **Databases**, and then click to select the appropriate mailbox.
2. Click **Delete** to remove the mailbox database.

Use the Shell to remove a mailbox database

For detailed syntax and parameter information, see Remove-MailboxDatabase.
1. Run the following command to remove the mailbox database MyDatabase.

```
Remove-MailboxDatabase -Identity "MyDatabase"
```

2. When you're prompted about whether you're sure that you want to perform the action, type **Y**.
3. When the dialog box appears stating that the database was removed successfully, note the location of the Exchange 2013 database (.edb) file. If you want to remove this file from the hard drive, you must remove it manually.

How do you know this worked?

To verify that you have successfully removed the mailbox database, do the following:
- From the EAC, select **Servers** > **Databases**.
- Verify that the mailbox database has been removed.

Configure circular logging for a mailbox database

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-24*

When you enable circular logging for a mailbox database, the type of circular logging you get depends on whether or not the mailbox database is replicated using continuous replication:

- If the mailbox database is not replicated, it will use JET circular logging. In this case, enabling or disabling JET circular logging will require a dismount and mount of the database.
- If the mailbox database is replicated, it will use continuous replication circular logging (CRCL). In this case, enabling or disabling CRCL takes effect dynamically; there is no need to dismount and re-mount the database.

For more information about circular logging and CRCL, see Exchange Native Data Protection.

What do you need to know before you begin?

- Estimated time to complete: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox Database Permissions" entry in the Recipients Permissions topic.

Use the EAC to configure circular logging for a database

1. In the EAC, go to **Servers** > **databases**.
2. Select the mailbox database you want to configure and click ✎.
3. Check or uncheck the **Enable circular logging** checkbox, and then click **save**.
4. If a dismount and mount operation are required, a warning message will appear. Click **OK** to close the warning message.
   a. To dismount the database, click **More** ⋯, and then click **Dismount**. Click **yes** when the warning message appears.
   b. To mount the database, click **More** ⋯, and then click **Mount**. Click **yes** when the warning message appears.

Use the Shell to configure circular logging for a database

This example enables circular logging for database DB1.

```
Set-MailboxDatabase DB1 -CircularLoggingEnabled $True
```

This example disables circular logging for database DB1.

```
Set-MailboxDatabase DB1 -CircularLoggingEnabled $False
```

See Set-MailboxDatabase for other mailbox database parameters you can configure.

Understanding Exchange 2013 page zeroing

Topic Last Modified: *2014-05-09*

Page Zeroing in Exchange 2013

Zeroing is a security mechanism that writes either zeros or a binary pattern over deleted data so that the deleted data more difficult to recover. In Exchange Server 2013, an ESE database uses *pages* as its unit of storage, and as a result it implements *page zeroing*. Page zeroing is enabled by default, and it cannot be disabled. Page zeroing operations are recorded in the transaction log files so that all copies of a database are page-zeroed in a similar manner. Zeroing a page on an active database causes the page to get zeroed on a passive copy of the database.

Note:

There is no mechanism for the Extensible Storage Engine (ESE) to prioritize the reutilization of zeroed pages over allocating new space. Tables which have sequential space allocation assigned will intentionally skip fragmented or zeroed pages in favor of using new or free sequential pages. This approach reduces database IOPs.

In Exchange 2013 page zeroing reduces the performance impact on servers when they're performing zeroing functions. This includes:

- **Optimized storage and network capacity** ESE writes a page-zeroing record to the transaction log file instead of logging the entire page image. This approach reduces log write I/O, and reduces the bandwidth requirements for shipping log files.
- **Optimized database disk I/O** In Exchange 2010 RTM and earlier, page zeroing occurred only during a backup, or during scheduled maintenance, and it caused significant database disk I/O. In Exchange 2010 SP1 and later (including Exchange 2013), page zeroing occurs by default and happens at transaction time. In the majority of cases, zeroing occurs immediately after a hard delete. This design allows the database to leverage the checkpoint depth capability of the engine, which ensures that dirty pages stay in the database cache for a certain amount of time so that additional page updates that occur in close time proximity don't cause additional database write I/Os. Because of this design, page zeroing has no significant database I/O impact, which is why it's enabled by default.

Implementation of Page Zeroing in the ESE Database

Page zeroing writes a binary pattern over a hard-deleted record. The page-zeroing pattern is specific to the ESE engine operation and it is different for run-time operations versus maintenance operations. The following table lists the fill patterns that correspond to specific run-time operations.

Fill pattern of page zeroing during ESE run-time

ESE run-time operation	Fill pattern
Replace	R
Record/long value delete	D
Freed page space	H

The following table lists the fill patterns that correspond to specific operations that occur during ESE background database maintenance.

Fill pattern of page zeroing during ESE background database maintenance

ESE background database maintenance operation	Fill pattern
Record delete	D
Long value delete	L
Freed page space of partially used page	Z
Freed page space of unused page	U

Background Database Maintenance

Background database maintenance is a process that continuously checksums and scans each database. Its primary function is to checksum database pages, but it also handles cleaning up space and zeroing out records and pages that were not zeroed out because of a Store crash. Background database maintenance processes approximately 1 MB per second per database. If timely page zeroing is a priority, you can reduce database sizes to ensure page zeroing occurs for the crash recovery cases in a shorter time period (for example, 24 hours).

Background database maintenance is a continuous process, so there are no events associated with its start and completion. You can track the progress of background database maintenance by reading the value of a performance counter:

- MSExchange Database -> Instances -> Database Maintenance Duration

This counter indicates the number of seconds that have passed since maintenance last completed for a given database.

Process of ESE Database Page Zeroing

The following table discusses database delete scenarios, and when page zeroing functions occur.

ESE background database maintenance

Database delete scenario	ESE process and timeframe to zero database data
• Scenario 1: Single item recovery is disabled and user purges item from the Recoverable Items folder. • Scenario 2: Single item recovery is disabled and the Recoverable Items retention period is set to zero. • Scenario 3: Single item recovery is enabled and the item expires based on the deleted item retention period.	An asynchronous thread writes a binary pattern over the deleted data. This action occurs within milliseconds of the record deletion. If the Store process crashes while the asynchronous zeroing work is still outstanding (or version store cleanup is cancelled due to version store growth), the zeroing is completed when background database maintenance processes that section of the database.
View Scenario: Expiration of items from Outlook/ Outlook Web App folder view (for example, Conversation view)	Data zeroing occurs when background database maintenance processes that section of the database.
Move Mailbox/Delete Mailbox Scenario: Source mailbox deleted (expiry of deleted mailbox from dumpster)	Data zeroing occurs when background database maintenance processes that section of the database.

Monitoring Page Zeroing Behavior

You can measure and monitor page zeroing functionality by viewing two ESE counters:

- MSExchange Database -> Database Maintenance Pages Zeroed: Indicates the number of pages zeroed by the database engine since the performance counter was invoked.
- MSExchange Database -> Database Maintenance Pages Zeroed/sec: Indicates the rate at which pages are zeroed.

Note:
To learn how to enable these counters, see How to Enable Extended ESE Performance Counters.

Page zeroing is a database maintenance function, so performance information related to both page zeroing for run-time transactions and page zeroing due to background database maintenance is included in these counters.

Mailbox Data Types without Page Zeroing

The following Mailbox data types have no provisions for page zeroing:

- Mailbox database transaction logs (.log)

When transaction logs are deleted (due to truncation via backup or circular logging), there is no process to zero the blocks in the NTFS file system that stored the deleted log file(s). It's likely that NTFS will quickly re-utilize that free space for newly created logs, but there is no guarantee that this will happen.

- Content index catalog files

Exchange 2013 uses Search Foundation for search indexing functionality. The search index catalog is comprised of several dozen files stored on the same volume as the mailbox database file. When a message is hard-deleted from the mailbox database, the associated content in the search catalog isn't immediately deleted. Content deletion occurs when Search Foundation does a shadow, or master merge, of many small catalog files in to a single larger file. After the master merge completes, the smaller catalog files are deleted. There is no process to zero the blocks which stored the deleted catalog files.

Exchange Search

Exchange Server 2013 > Mailbox and Client Access servers > Mailbox server >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2014-04-24*

With increasing mailbox sizes and increasing amounts of data being stored in mailboxes in the form of messages and attachments, it's crucial for users to be able to quickly search and locate the messages they need. In-Place Archiving helps you reduce or eliminate the use of .pst files by moving old and infrequently accessed items to the archive. This results in more mailbox data being stored by a user, and it makes searching across the user's primary and archive mailboxes an important productivity tool. In-Place eDiscovery allows authorized users to search content in mailboxes across on-premises and cloud-based Exchange organizations to comply with electronic discovery (eDiscovery) requests, regulatory audits, or internal investigations. In-Place eDiscovery also uses the content indexes created by Exchange Search.

Exchange Search is different from full-text indexing available in Exchange Server 2003. Improvements were made to performance, content indexing, and search. New items are indexed in the transport pipeline or almost immediately after they're created or delivered to the mailbox, providing users with a fast, stable, and more reliable way of searching mailbox data. Content indexing is enabled by default, and there's no initial setup or configuration required.

Looking for management tasks related to Exchange Search? See Exchange Search procedures.

What's New

Exchange 2013 introduces the following changes to Exchange Search:

- The underlying content indexing engine has been replaced with Microsoft Search Foundation, which provides performance and functionality improvements and serves as the common underlying content indexing engine in Exchange and SharePoint. The management interface, however, remain the same.
- By default, the Search Foundation handles the most common file formats in email attachments. You no longer need to install Microsoft Office Filter Packs for Exchange Search. For a list of the file formats handled by Exchange Search, see File formats indexed by Exchange Search.

You can add support for any additional file formats by install IFilters, as in Exchange 2010.

- Content indexing is more efficient because it now processes messages in the transport pipeline. As a result, messages addressed to multiple recipients or distribution groups are processed only once. An annotation stream is attached to the message, significantly speeding up content indexing while consuming fewer resources.

File formats indexed by Exchange Search

Mailbox and Client Access servers > Mailbox server > Exchange Search >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-21*

In Microsoft Exchange Server 2013 and Exchange Online, Exchange Search includes filters for indexing most common types of file formats included as message attachments. You can also install filters to index additional file types.

Note:
In Exchange 2013, it isn't required to install and register Microsoft Office Filter Pack.

When managing or using Exchange Search and the dependent features (such as In-Place eDiscovery), consider the difference between unsearchable items and file formats that are disabled

for indexing or contain content that can't be indexed:

- **Unsearchable items** When Exchange Search can't index a particular file type for any reason (for example, if a filter isn't installed), the search for the file type fails. Messages containing such attachments are marked as *partially indexed*. Unsearchable items can be retrieved using the Get-FailedContentIndexDocuments cmdlet. When copying In-Place eDiscovery search results to a discovery mailbox or exporting search results to a PST file, you can include unsearchable items. For more information, see Unsearchable items in Exchange eDiscovery.
- **File formats with content that can't be indexed** Certain file types such as Windows Media Video (WMV) don't contain content that can be indexed and therefore aren't indexed. Messages containing attachments of such file types are also returned as unsearchable items in In-Place eDiscovery searches.
- **Disabled file formats** In on-premises organizations, an administrator can disable indexing of a specified file format. Messages that contain an attachment that is of a disabled format are returned as unsearchable items.

Important:
Although a message attachment may be unsearchable or is of a file format that can't be indexed, the message subject, message body and other metadata may be indexed so that the message can be returned in searches.

For additional management tasks related to Exchange Search in on-premises organizations, see Exchange Search procedures.

Default filters

The following table lists the default search filters installed on an Exchange 2013 Mailbox server and in Exchange Online. You can retrieve the list of default filters by using the Get-SearchDocumentFormat cmdlet.

Filter	File extension
Email message	.eml
Graphics Interchange Format	.gif
JPEG	.jpeg
Microsoft Excel	.xls, .xlt, .xlsx, .xlsm, .xlb, .xlc, .xlsb
Excel File	odbcexcel
Microsoft InfoPath	.infopathml
Microsoft Office Binder	.obt, obd

Microsoft PowerPoint	.pptx, .pptm, .ppt, .ppsx, .ppsm, .pps, .ppam, .potm, .pot, .potx
Microsoft Publisher	.pub
Microsoft Word	.doc, .docm, .dotx, .dotm, .dot, .docx
Microsoft XML Paper Specification	.xps
OneNote	.one
OpenDocument Presentation	.odp
OpenDocument Spreadsheet	.ods
OpenDocument Text	.odt
Outlook Item	.msg
Portable Document Format	.pdf
Rich Text	.rtf
Text	.txt
vCalendar	.vcs
vCard	.vcf
Visio	.vdw, .vsd, .vss, .vst, .vsx, .vtx, .vssx, .vssm, .vsdm, .vstx, .vstm, .vdx
Web archive	.mhtml
Web page	.html
XML document	.xml
ZIP archive	.zip

Disabled file formats

The following table lists the search filters that are disabled for indexing by default on an Exchange 2013 Mailbox server and in Exchange Online. In Exchange 2013, administrators can disable or re-enable a supported file format for indexing by using the Set-SearchDocumentFormat cmdlet. This cmdlet isn't available in Exchange Online.

Filter	File extension
AVI	.avi
Bitmap	.bmp
MP3	.mp3
MPEG	.mpeg
PNG	.png
Microsoft Windows Wave Audio	.wav

Message properties indexed by Exchange Search

Mailbox and Client Access servers > Mailbox server > Exchange Search >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-13*

Exchange Search indexes many item properties, including sender, recipients, message body, and attachments for email messages.

Properties indexed by Exchange Search

The following table includes a list of all item properties indexed by Exchange Search.

Property	Type	Queryable	Searchable	Retrievable
Account	String	Yes	Yes	No
Assistantname	String	Yes	Yes	No
Attachmentfilenames	String	Yes	Yes	No

Attachmentmetaproperties	String	No	Yes	No
Attachmentcount	Integer	No	No	Yes
Bcc	String	Yes	Yes	No
Body	String	No	Yes	No
Businessaddress	String	Yes	Yes	No
Businessmainphone	String	Yes	Yes	No
Businessphonenumber	String	Yes	Yes	No
Carphonenumber	String	Yes	Yes	No
Categories	String	Yes	Yes	No
Cc	String	Yes	Yes	No
Companyname	String	Yes	Yes	No
Compositeitemid	String	No	No	Yes
Conversationid	Integer	No	No	Yes
Conversationtopic	String	Yes	Yes	No
Departmentname	String	Yes	Yes	No
Displayname	String	Yes	Yes	No
Displaynameprefix	String	Yes	Yes	No
Documentid	Integer	Yes	No	Yes
Emailaddress	String	Yes	Yes	No
Emaildisplayname	String	Yes	Yes	No

Emailoriginaldisplayname	String	Yes	Yes	No
Errorcode	Integer	Yes	No	Yes
Fileas	String	Yes	Yes	No
Firstname	String	Yes	Yes	No
Folderid	String	Yes	No	No
From	String	Yes	Yes	No
Homeaddress	String	Yes	Yes	No
Homephone	String	Yes	Yes	No
Importance	Integer	Yes	No	No
Ispartiallyprocessed	Boolean	Yes	No	Yes
Ispermanentfailure	Boolean	Yes	No	Yes
Itemclass	String	Yes	No	No
Lastattempttime	DateTime	Yes	No	Yes
Lastname	String	Yes	Yes	No
Mailboxguid	String	Yes	No	Yes
Manager	String	Yes	Yes	No
Meetinglocation	String	Yes	Yes	No
Middlename	String	Yes	Yes	No
Mobilephonenumber	String	Yes	Yes	No

Nickname	String	Yes	Yes	No
Officelocation	String	Yes	Yes	No
Otheraddress	String	Yes	Yes	No
Primarytelephonenumber	String	Yes	Yes	No
Received	DateTime	Yes	No	No
Receivedby	String	Yes	Yes	No
Receivedrepresenting	String	Yes	Yes	No
Recipients	String	Yes	Yes	No
Sent	DateTime	Yes	No	No
Sharinginfo	String	Yes	No	No
Size	Integer	Yes	No	No
Subject	String	Yes	Yes	No
Tasktitle	String	Yes	Yes	No
Title	String	Yes	Yes	No
To	String	Yes	Yes	No
Umaudionotes	String	Yes	Yes	No
Watermark	Integer	No	No	Yes
Yomicompanyname	String	Yes	No	Yes
Yomifirstname	String	Yes	Yes	No
Yomilastname	String	Yes	Yes	No

Notes about indexed properties:

- **Queryable properties** can be used in search queries (either programmatically or in KQL queries) for In-Place eDiscovery and in AQS queries by search clients such as Outlook Web App in `property:value` pairs, for example, `from:bsuneja@cotoso.com`.
- **Searchable properties** are properties that can't be specified in `property:value` pairs, but a keyword search returns the value if found in any searchable property. For example, you can't use `body:Contoso` to search for the string `contoso` in the message body only. However, a search for that string will return all items where the property is found in any searchable property.
- **Retrievable properties** such as `documenteid` and `ispartiallyprocessed` are returned with every search.

Exchange Search procedures

Mailbox and Client Access servers > Mailbox server > Exchange Search >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-05*

Disable or enable Exchange Search

Diagnose Exchange Search issues

Reseed the search catalog

Disable or enable Exchange Search

Mailbox server > Exchange Search > Exchange Search procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-07*

By default, Exchange Search is enabled for all new mailbox databases and doesn't require additional configuration. However, if you want to stop Exchange Search from indexing mailbox content, you can disable it for individual mailbox databases or for an entire Mailbox server.

Caution:
Disabling Exchange Search impacts the functionality and performance of the full-text searches that are performed by your users using Outlook in online mode or on Windows mobile devices. In-Place eDiscovery also relies on Exchange Search. If you disable Exchange Search for a mailbox database or for a Mailbox server, In-Place eDiscovery searches won't return any messages from the database or server.

For additional management tasks related to Exchange Search, see Exchange Search procedures.

What do you need to know before you begin?

- Estimated time to complete each procedure: 1 minute
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- You can enable or disable Exchange Search for servers or mailbox databases but not for individual mailbox users.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Disable or enable Exchange Search for a mailbox database

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange Search" entry in the Recipients Permissions topic.

Note:
You can't use the EAC to disable or enable Exchange Search for a mailbox database.

This command disables Exchange Search for a mailbox database named EXCH01.

```
Set-MailboxDatabase "Mailbox Database (EXCH01)" -
IndexEnabled $false
```

This command enables Exchange Search for a mailbox database named EXCH01.

```
Set-MailboxDatabase "Mailbox Database (EXCH01)" -
IndexEnabled $true
```

For detailed syntax and parameter information, see Set-MailboxDatabase.

Disable or enable Exchange Search for a Mailbox server

To disable Exchange Search for a Mailbox server, you have to disable and stop the Microsoft Exchange Search service. Similarly, to enable Exchange Search for a Mailbox server, you have to enable and start the Microsoft Exchange Search service. You can use either the Services console or the Shell to do this.

You need to be assigned permissions before you can perform this procedure or procedures. To see

what permissions you need, see the "Manage Exchange Search service on a Mailbox server" entry in the Recipients Permissions topic.

Use the Services console

1. Go to **Start** > **Administrative Tools** > **Services**.
2. In the **Services** details pane, right-click the **Microsoft Exchange Search** service, and then select **Properties**.
3. On the **General** tab, in the **Startup type** list, select **Disabled** to disable the service or **Automatic** to start it automatically.

Note:

The startup type impacts the service the next time an attempt is made to start it, either automatically after the server is restarted or by manually starting the service. In the next step, the service is stopped or started manually.

4. Click **Stop** to stop the service or **Start** to start the service.
5. Click **OK** to save the changes.

Use the Shell

Run the following commands to stop and disable the Microsoft Exchange Search service.

```
Stop-Service MSExchangeFastSearch

Set-Service MSExchangeFastSearch -StartupType Disabled
```

Run the following commands to configure the Exchange Search service to start automatically and then start the service.

```
Set-Service MSExchangeFastSearch -StartupType Automatic

Start-Service MSExchangeFastSearch
```

Diagnose Exchange Search issues

Mailbox server > Exchange Search > Exchange Search procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-09*

Exchange Search indexes mailboxes and supported attachments in Exchange mailboxes. With increasing volumes of e-mail, increasing mailbox sizes and storage quotas, provisioning of personal archive mailboxes for users, and the introduction of Multi-Mailbox Search for performing discovery searches, Exchange Search is a critical component of the Mailbox servers in your Microsoft

Exchange Server 2010 organization. Issues with Exchange Search can affect user productivity and impact Multi-Mailbox Search functionality.

To learn more about Exchange Search, see Exchange Search.

Looking for management tasks related to managing Exchange Search? See Exchange Search procedures.

Using the Test-ExchangeSearch Cmdlet

Step 5 of the procedure in this topic describes running the **Test-ExchangeSearch** cmdlet to help diagnose Exchange Search issues. You can use the **Test-ExchangeSearch** cmdlet to test Exchange Search functionality for a Mailbox server, a mailbox database, or a specific mailbox. The cmdlet delivers a test message to the specified mailbox (or to a database's system mailbox if a mailbox isn't specified), and then performs a search to determine whether the message is indexed, including the time taken to index it. Under normal conditions, Exchange Search indexes a message within about 10 seconds of the message being created or delivered to a mailbox. The test message is automatically deleted after the test.

Exchange 2010 includes the following enhancements to the **Test-ExchangeSearch** cmdlet:

- The *Mailbox* parameter has been added to the standard output.
- When you specify a server name, the cmdlet simultaneously tests all mailbox databases on the Mailbox server. For databases that are replicated to other Mailbox servers in a database availability group (DAG), if you run the command on a Mailbox server that doesn't contain the active database copy, the test is automatically performed against the server that contains the active database copy.
- When you use the cmdlet with the *MonitoringContext* parameter, it provides additional data that can be used by monitoring software such as Microsoft System Center Operations Manager 2007.
- When you use the cmdlet with the *Verbose* switch, the cmdlet returns detailed results and status for every step, and additional diagnostic information to help you troubleshoot issues related to search.

For detailed syntax and parameter information, see Test-ExchangeSearch.

Retrieving Unsearchable Items

You can use the **Get-FailedContentIndexDocuments** cmdlet to retrieve a list of unsearchable mailbox items that couldn't be successfully indexed by Exchange Search. You can run the cmdlet against a Mailbox server, a mailbox database, or a specific mailbox. The cmdlet returns details about each item that couldn't be searched. There are several reasons why a mailbox item can't be searched; for example, an e-mail message includes an attachment file type for which a search filter isn't installed. If a search filter for that file type is available, you can install it on your Exchange servers.

Important:

Search filters provided by Microsoft are tested and supported by Microsoft. We recommend that you test any third-party search filters in a test environment before installing them on Exchange servers in a production environment.

Note:

Messages that contain an attachment file format that's listed on the safe list aren't returned in the list of unsearchable items. For more details, see "Exchange Search and Attachments" in Exchange Search.

For detailed syntax and parameter information, see Get-FailedContentIndexDocuments.

Diagnose Exchange Search Issues

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange Search" entry in the Recipients Permissions topic.

1. **Check service state** Is the Microsoft Exchange Search (MSExchangeFastSearch) service started on the Mailbox server? If yes, go to Step 2. If no, use the Services MMC snap-in to verify that the MSExchangeFastSearch service is running as follows:
   a. Click **Start**, point to **Administrative Tools**, and then click **Services**.
   b. In **Services**, verify that the **Status** for the **Microsoft Exchange Search** service is listed as **Started**.
2. **Check mailbox database configuration** Is the *IndexEnabled* parameter set to true for the user's mailbox database? If yes, go to Step 3. If no, run the following command in the Shell to verify that the *IndexEnabled* flag is set to true.

```
Get-MailboxDatabase | Format-Table Name,IndexEnabled
```

For detailed syntax and parameter information, see Get-MailboxDatabase.

3. **Check mailbox database crawl state** Has the Exchange database been crawled? If yes, go to Step 4. If no, use Reliability and Performance Monitor to check the **Crawler: Mailboxes Remaining** counter of the **MSExchange Search Indexes** performance object. Perform the following steps:
   a. Open Reliability and Performance Monitor (perfmon.exe).
   b. In the console tree, under **Monitoring Tools**, click **Performance Monitor**.
   c. In the Performance Monitor pane, click **Add** (green plus sign).
   d. In **Add Counters**, in the **Select counters from computer** list, select the server on which the mailbox database you want to monitor is located.
   e. In the unlabeled box below the **Select counters from computer** list, select the **MSExchange Search Indexes** performance object.
   f. In the **Instances of selected object** box, select the instance for the user's mailbox database.
   g. Click **Add**, and then click **OK**.

In the Performance Monitor pane, the **MSExchange Search Indexes** performance object is listed in the **Object** column, and its various counters are listed in the **Counter** column.

h. View the **Crawler: Mailboxes Remaining** counter. Any value of 1 or higher indicates that mailboxes in the database are still being crawled. When the crawl is complete, the value is **0**.

For information about using Performance Monitor, see Performance and Reliability Monitoring Step-by-Step Guide for Windows Server 2008

4. **Check the database copy indexing health** Is the content index healthy? Use the **Get-MailboxDatabaseCopyStatus** cmdlet to check the content indexing health for a database copy.

```
Get-MailboxDatabaseCopyStatus -Server $env:ComputerName |
Format-Table Name,Status,ContentIndex* -Auto
```

For detailed syntax and parameter information, see Get-MailboxDatabaseCopyStatus.

5. **Run the Test-ExchangeSearch cmdlet** If the mailbox database has already been crawled, you can run the **Test-ExchangeSearch** cmdlet for the mailbox database or for a specific mailbox.

```
Test-ExchangeSearch -Identity AlanBrewer@contoso.com
```

For detailed syntax and parameter information, see Test-ExchangeSearch.

6. **Check the Application event log** Using Event Viewer or the Shell, check the Application event log for search-related error messages. Check the **Source: MSExchangeSearch Indexer** and **msftesql-Exchange** events. For more information, follow the link in the event log entry.
7. **Restart the Microsoft Exchange Search service** Use the Services MMC snap-in or the Shell to stop and then restart the Microsoft Exchange Search (MSExchangeFastSearch) service:
   a. Click **Start**, point to **Administrative Tools**, and then click **Services**.
   b. In **Services**, right-click **Microsoft Exchange Search**, and then click **Stop**. After the service is stopped, right-click the service again, and then click **Start**.
8. **Reseed the search catalog** In some cases, such as when the search catalog is corrupted, you may need to reseed the catalog. When a search catalog needs to be reseeded, Exchange Search notifies you by logging entries in the Application event log. For more information about reseeding the Search catalog, see Reseed the search catalog.

Reseed the search catalog

Mailbox server > Exchange Search > Exchange Search procedures >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-04*

If the content index catalog for a mailbox database copy gets corrupted, you may need to reseed the catalog. Corrupted content indexes are indicated in the Application event log by the following event.

Event ID	Level	Source	Details

123	Error	ExchangeStoreDB	At <*timestamp*> the Microsoft Exchange Information Store Database <*identity*> copy on this server experienced a corrupted search catalog. Consult the event log on the server for other "ExchangeStoreDb" and "MSExchange Search Indexer" events for more specific information about the failure. Reseeding the catalog is recommended via the 'Update-MailboxDatabaseCopy' task.

If the mailbox database copy is located on a server that is part of a database availability group (DAG), you can reseed the content index catalog from another DAG member.

If the mailbox database copy is the only copy, you have to manually create a new content index catalog.

For other management tasks related to Exchange Search, see Exchange Search procedures.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes. Actual reseeding time may vary depending on the size of the content index catalog being reseeded.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange Search" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see

Keyboard shortcuts in the Exchange admin center.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Reseed the content index catalog if the mailbox database is part of a DAG

Use one of the following procedures if the mailbox database is located on a server that is part of a DAG.

Reseed the content index catalog from any source

This example reseeds the content index catalog for the database copy DB1 on Mailbox server MBX1 from any source server in the DAG that has a copy of the database.

```
Update-MailboxDatabaseCopy -Identity DB1\MBX1 -CatalogOnly
```

For detailed syntax and parameter information, see Update-MailboxDatabaseCopy.

Reseed the content index catalog from a specific source

This example reseeds the content index catalog for the database copy DB1 on Mailbox server MBX1 from Mailbox server MBX2, which also has a copy of the database.

```
Update-MailboxDatabaseCopy -Identity DB1\MBX1 -SourceServer
MBX2 -CatalogOnly
```

For detailed syntax and parameter information, see Update-MailboxDatabaseCopy.

Reseed the content index catalog if there is only one copy of the mailbox database

If there is only one copy of the mailbox database, you have to manually reseed the search catalog by recreating the content index catalog.

1. Run the following commands to stop the Microsoft Exchange Search and Microsoft Exchange Search Host Controller services.

```
Stop-Service MSExchangeFastSearch

Stop-Service HostControllerService
```

2. Delete, move, or rename the folder that contains the Exchange content index catalog. This folder is named `%ExchangeInstallPath\Mailbox\<name of mailbox database>_Catalog\<GUID>12.1.Single`. For example, you might rename the folder `C:\Program Files\Microsoft\Exchange Server\V15\Mailbox\Mailbox Database 0657134726_Catalog\F0627A72-9F1D-494A-839A-D7C915C279DB12.1.Single_OLD`.

Note:
Deleting this folder will make additional disk space available. Alternatively, you might want to rename or move the folder to keep it for troubleshooting purposes.

3. Run the following commands to restart the Microsoft Exchange Search and Microsoft Exchange Search Host Controller services.

```
Start-Service MSExchangeFastSearch

Start-Service HostControllerService
```

After you restart these services, Exchange Search will rebuild the content index catalog.

How do you know this worked?

It might take a while for Exchange Search to reseed the content index catalog. Run the following command to display the status of the reseeding process.

```
Get-MailboxDatabaseCopyStatus | FL Name,*Index*
```

When the reseeding of the search catalog is in progress, the value of the *ContentIndexState* property is **Crawling**. When the reseeding is complete, this value is changed to **Healthy**.

Mailbox moves in Exchange 2013

Exchange Server 2013 > Mailbox and Client Access servers > Mailbox server >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-25*

When you move a mailbox, you're moving it from a *source mailbox database* to a *target mailbox database*. The target mailbox database can be on the same server, on a different server, in a different domain, in a different Active Directory site, or in another forest.

Reasons for moving mailboxes

You may need to move mailboxes in the following scenarios:

- **Upgrade** When you upgrade from an existing Microsoft Exchange Server 2007 or Exchange Server 2010 organization to Exchange Server 2013, you move mailboxes from the existing Exchange servers to an Exchange 2013 Mailbox server.
- **Realignment** You can move mailboxes for realignment purposes. For example, you may want to move a mailbox from one database to a database that has a larger mailbox size limit.
- **Investigate an issue** If you need to investigate an issue with a mailbox, you can move that mailbox to a different server. For example, you can move all mailboxes that have high activity to another server.
- **Corrupted mailboxes** If you encounter corrupted mailboxes, you can move the mailboxes to a different server or database. The corrupted messages won't be moved.
- **Physical location changes** You can move mailboxes to a server in a different Active Directory site. For example, if a user moves to a different physical location, you can move that user's mailbox to a server closer to the new location.
- **Separation of administrative roles** You may want to separate Exchange administration from Windows operating system account administration. To do this, you can move mailboxes from a single forest into a resource forest scenario. In this scenario, the Exchange mailboxes reside in one forest and their associated Windows user accounts reside in a separate forest.
- **Outsource email administration** You may want to outsource the administration of email and retain the administration of Windows user accounts. To do this, you can move mailboxes from a single forest into a resource forest scenario.
- **Integrate email and user account administration** You may want to change from a separated or outsourced email administration model to a model in which email and user accounts can be managed from within the same forest. To do this, you can move mailboxes from a resource forest scenario to a single forest. In this scenario, the Exchange mailboxes and Windows user accounts reside in the same forest.

Exchange 2013 moves

Microsoft Exchange Server 2013 introduces the concept of *batch moves* and *migration endpoints*. Migration endpoints are management objects that describe the remote server and the connections that can be associated with one or more batches. And, the new batch move architecture improves on Mailbox Replication service (MRS) moves with enhanced management capability. Batch moves architecture in Exchange 2013 features the following capabilities:

- Ability to move multiple mailboxes in large batches.
- Email notification during move with reporting.
- Automatic retry and automatic prioritization of moves.
- Primary and personal archive mailboxes can be moved together or separately.
- Option for manual move request finalization, which allows you to review your move before you complete it.
- Periodic incremental syncs to update migration changes.

In Exchange 2013 you must move mailboxes between Exchange 2007 and Exchange 2010 and Exchange 2013 from the Exchange 2013 admin center (EAC) and the Exchange Management Shell.

Note:
You can still perform single mailbox moves in Exchange 2013 similar to Exchange Server 2010 by using either the EAC or the move request or migration batch cmdlets.

Note:
You can't move on-premises mailboxes from Exchange Server 2003 to Exchange 2013.

For more information about managing new and existing moves, see Manage on-premises moves.

Migration endpoints

Migration endpoints capture the remote server information and persist the required credentials for migrating the data and the source throttling settings. You can use migration endpoints to configure settings for remote and cross-forest moves. Local moves don't require the use of an endpoint. (Local moves are moves that move mailboxes between two different on-premises Exchange databases.)

The following list shows the types of moves that support migration endpoints:

- **Cross-forest move** Move mailboxes between two different on-premises Exchange forests. Cross-forest moves require the use of a Exchange RemoteMove endpoint.
- **Remote move** In a hybrid deployment, a remote move involves *onboarding* or *offboarding* migrations. Remote moves require the use of a RemoteMove endpoint. Onboarding moves mailboxes from an on-premises Exchange organization to Exchange Online in Microsoft Office 365, and uses a RemoteMove endpoint as the source endpoint of the migration batch. Offboarding moves mailboxes from Exchange Online in Office 365 to an on-premises Exchange organization and uses a Exchange RemoteMove endpoint as the target endpoint of the migration batch.

The following table shows the migration endpoint types and values that you can manage in Exchange 2013.

Values of migration endpoint types

Exchange RemoteMove	Exchange LocalMove
MaxConcurrentMigrations	MaxConcurrentMigrations
RemoteServer	Site
ArchiveDomain	Database
BadItemLimit	ArchiveDatabase
LargeItemLimit	BadItemLimit
Databases	LargeItemLimit

TargetDeliveryDomain	PrimaryOnly
PrimaryOnly	ArchiveDatabase
ArchiveDatabase	DAG
ArchiveOnly	Forest

For more information about migration endpoints, see the **Migration** user interface in the EAC and New-MigrationEndpoint.

For more information about managing new and existing moves, see Manage on-premises moves.

Manage on-premises moves

Mailbox and Client Access servers > Mailbox server > Mailbox moves in Exchange 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-25*

A move request is the process of moving a mailbox from one mailbox database to another. A local move request is a mailbox move that occurs within a single forest. In Microsoft Exchange Server 2013, mailboxes and personal archive mailboxes can reside on separate databases. Using the move request functionality, you can move the primary mailbox and the associated archive to the same database or to separate ones. The procedures in this topic will help you with on-premises mailbox moves.

Use the following procedures to move mailboxes in your on-premises organization. These procedures use the Exchange Management Shell and the Exchange Center (EAC).

When you use move request to move mailboxes, the move requests are processed by the following two services:

- Microsoft Exchange Mailbox Replication service
- Microsoft Exchange Mailbox Replication Proxy

For more information about the Mailbox replication server and proxy, see Learn more about MRS Proxy.

For more information about mailbox moves, see Mailbox moves in Exchange 2013.

What do you need to know before you begin?

- Estimated time to complete each procedure: 20 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "Mailbox Move and Migration Permissions " entry in Recipients Permissions.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Test whether a mailbox is ready to move

This example uses the *WhatIf* switch to test whether Tony Smith's mailbox is ready to move to the new database DB01 and if there are any errors within the command. When you use the *WhatIf* switch, the system performs checks on the mailbox. If the mailbox isn't ready to move, you receive an error.

```
New-MoveRequest -Identity 'tony@alpineskihouse.com' -
TargetDatabase DB01 -WhatIf
```

For detailed syntax and parameter information, see New-MigrationBatch and New-MoveRequest.

Create a local move request

Use the EAC to create a local move request

To create a local move request, log in to the EAC and perform the following steps:

1. In the EAC, navigate to **Recipients** > **Migration**, and then click **Add +**.
2. In the **New local mailbox move** wizard, select the user you want to move click **OK** and then click **Next**.
3. On the **Move configuration** page, specify a name for the new batch. Select which options you want for the archive mailbox, and mailbox database location and click **New**.

Use the Shell to create a local move request

For an example of how to create a local move request, see Example 2 in New-MoveRequest.

How do you know this worked?

To verify that you have successfully completed your migration, do the following:

- In the EAC, navigate to **Recipients** > **Migration**.
- Verify that your move was successful in the EAC by clicking **Status For All Batches**.
- From the Shell, run the following command to retrieve mailbox move information.

```
Get-MigrationUserStatistics -Identity BatchName -Status |
Format-List
```

For more information, see Get-MigrationUserStatistics.

Create a batch move request

Use the EAC to create a batch move request

log in to the EAC and perform the following steps:

1. In the EAC, navigate to **Recipients** > **Migration**, and then click **Add +**.
2. In the **New local mailbox move** wizard, select the users you want to move, click **OK** and then click **Next**.
3. On the **Move configuration** page, specify a name for the new batch. Select which options you want for the archive mailbox, and mailbox database location and click **New**.

⚠**Warning:**
Make sure that you don't set the Bad Item Limit to over 50 items. If you do, the move may fail. If you want to set the Bad Item Limit over 50 items, you must use the Exchange Management Shell and set the *–AcceptLargeDataLoss* parameter to true.

Use the Shell to create a batch move request

This example creates a migration batch for a local move, where the mailboxes in the specified .csv file are moved to a different mailbox database. This .csv file contains a single column that contains the email address for each of the mailboxes that will be moved. The header for this column must be named **EmailAddress**. The migration batch in this example must be started manually by using the **Start-MigrationBatch** cmdlet or the Exchange Administration Center (EAC). Alternatively, you can use the *AutoStart* parameter to start the migration batch automatically.

```
New-MigrationBatch -Local -Name LocalMove1 -CSVData
([System.IO.File]::ReadAllBytes("C:\Users\Administrator
\Desktop\LocalMove1.csv")) -TargetDatabases MBXDB2 -
TimeZone "Pacific Standard Time"

Start-MigrationBatch -Identity LocalMove1
```

For detailed syntax and parameter information, see New-MigrationBatch and Start-MigrationBatch.

How do you know this worked?

To verify that you have successfully completed your migration, do the following:

- Verify that your move was successful in the EAC by clicking **Status For All Batches**.
- From the Shell, run the following command to retrieve mailbox move information.

```
Get-MigrationUserStatistics -Identity BatchName -Status |
Format-List
```

For more information, see Get-MigrationUserStatistics.

Display migration batches

For an example of how to use the Shell to display a migration batch, see Example 2 in Get-MigrationBatch.

Move only a user's primary mailbox

Use the EAC to move only a user's primary mailbox

1. In the EAC, navigate to **Recipients** > **Migration**, and then click **Add +**.
2. In the **New local mailbox move** wizard, select the user whose primary mailbox you want to move, click **OK** and then click **next**.
3. On the **Move configuration** page, specify a name for the new batch. Select **Move primary mailbox only**, select which options you want for the mailbox database location, and then click **New**.

Use the Shell to move only a user's primary mailbox

This example moves only Tony Smith's primary mailbox to DB01. The archive isn't moved.

```
New-MoveRequest -Identity 'tony@alpineskihouse.com' -
PrimaryOnly -TargetDatabase "DB01"
```

For detailed syntax and parameter information, see New-MoveRequest.

How do you know this worked?

To verify that you have successfully completed your migration, do the following:

- In the EAC, click **Status For All Batches**.
- From the Shell, run the following command to retrieve mailbox move information.

```
Get-MigrationUserStatistics -Identity BatchName -Status |
Format-List
```

For more information, see Get-MigrationUserStatistics.

Create a cross-forest move using a .csv batch file

This example configures the migration endpoint, and then creates a cross-forest batch move from

the source forest to the target forest using a .csv file.

```
New-MigrationEndpoint -Name Fabrikam -ExchangeRemote -
Autodiscover -EmailAddress tonysmith@fabrikam.com -
Credentials (Get-Credential fabrikam\tonysmith)
$csvData=[System.IO.File]::ReadAllBytes("C:\Users
\Administrator\Desktop\batch.csv")
New-MigrationBatch -CSVData $csvData -Timezone "Pacific
Standard Time" -Name FabrikamMerger -SourceEndpoint
Fabrikam -TargetDeliveryDomain "mail.contoso.com"
```

For more information about preparing your forest for cross-forest moves, see the following topics:

- Prepare mailboxes for cross-forest move requests
- Prepare mailboxes for cross-forest moves using sample code
- Prepare mailboxes for cross-forest moves using the Prepare-MoveRequest.ps1 script in the Shell

For detailed syntax and parameter information, see New-MigrationBatch and New-MoveRequest.

How do you know this worked?

To verify that you have successfully completed your migration, do the following:

- From the Shell, run the following command to retrieve mailbox move information.

```
Get-MigrationUserStatistics -Identity BatchName -Status |
Format-List
```

For more information, see Get-MigrationUserStatistics.

Move only an archive mailbox

Use the EAC to move only an archive mailbox

1. In the EAC, navigate to **Recipients** > **Migration**, and then click **Add +**.
2. In the **New local mailbox move** wizard, select the user whose archive mailbox you want to move, click **OK** and then click **Next**.
3. On the **Move configuration** page, specify a name for the new batch. Select **Move archive mailbox only**, select which options you want for the mailbox database location, and then click **New**.

Use the Shell to move only an archive mailbox

This example moves only Tony Smith's archive mailbox to DB03. The primary mailbox isn't moved.

```
New-MoveRequest -Identity 'tony@alpineskihouse.com' -
ArchiveOnly -ArchiveTargetDatabase "DB03"
```

For detailed syntax and parameter information, see New-MigrationBatch and New-MoveRequest.

How do you know this worked?

To verify that you have successfully completed your migration, do the following:

- From the Shell, run the following command to retrieve mailbox move information.

```
Get-MigrationUserStatistics -Identity BatchName -Status |
Format-List
```

For more information, see Get-MigrationUserStatistics.

Move a user's primary mailbox and archive mailbox to separate databases

This example moves Ayla's primary mailbox and archive mailbox to separate databases. The primary database is moved to DB01, and the archive is moved to DB03.

```
New-MoveRequest -Identity 'ayla@humongousinsurance.com' -
TargetDatabase DB01 -ArchiveTargetDatabase -DB03
```

For detailed syntax and parameter information, see New-MigrationBatch and New-MoveRequest.

How do you know this worked?

To verify that you have successfully completed your migration, do the following:

- From the Shell, run the following command to retrieve mailbox move information.

```
Get-MigrationUserStatistics -Identity BatchName -Status |
Format-List
```

For more information, see Get-MigrationUserStatistics.

Move a user's primary mailbox and allow a large bad item limit

Use the EAC to move a user's primary mailbox and allow a large bad item limit

1. In the EAC, navigate to **Recipients** > **Migration**, and then click **Add +**.
2. In the **New local mailbox move** wizard, select the user whose primary mailbox you want to move, click **OK**, and then click **Next**.
3. On the **Move configuration** page, specify a name for the new batch. Select **Move primary mailbox only**, and then select which options you want for the mailbox database location.

4. Click **More Options** ⋯, enter the bad item limit, and then click **OK**.

Use the Shell to move a user's primary mailbox and allow a large bad item limit

This example moves Lisa's primary mailbox to mailbox database DB01 and sets the bad item limit to 100. To set a large bad item limit, you must use the *AcceptLargeDataLoss* parameter.

```
New-MoveRequest -Identity 'Lisa' -PrimaryOnly -
TargetDatabase "DB01" -BadItemLimit 100 -
AcceptLargeDataLoss
```

For detailed syntax and parameter information, see New-MigrationBatch and New-MoveRequest.

How do you know this worked?

To verify that you have successfully completed your migration, do the following:

- From the Shell, run the following command to retrieve mailbox move information.

```
Get-MigrationUserStatistics -Identity BatchName -Status |
Format-List
```

For more information, see Get-MigrationUserStatistics.

Move the Exchange 2010 system mailbox to Exchange 2013

Mailbox and Client Access servers > Mailbox server > Mailbox moves in Exchange 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-06-06*

In Exchange 2010, the Microsoft Exchange system mailbox is an arbitration mailbox used to store organization-wide data such as administrator audit logs, metadata for eDiscovery searches, and Unified Messaging data, such as menus, dial plans, and custom greetings. The Microsoft Exchange system mailbox is named **SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9}**; the display name is **Microsoft Exchange**.

When you upgrade your existing Exchange 2010 organization to Exchange 2013, you have to move the Microsoft Exchange system mailbox to a mailbox database on an Exchange 2013 Mailbox server. You should move this mailbox after you've installed and verified Exchange 2013. If you don't move this system mailbox to Exchange 2013, the following issues will occur when Exchange 2010 and Exchange 2013 coexist in your Exchange organization:

- Exchange 2013 tasks aren't saved to the administrator audit log. When you run the **Search-**

AdminAuditLog cmdlet or try to export the administrator audit log in the EAC, you'll receive an error that says you can't create an administrator audit log search because the system mailbox, SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9}, is located on a server that isn't running Exchange 2013. A Microsoft Exchange error with an Event ID of 5000 is also logged in the Windows Application log each time a command is run.

- You can't run eDiscovery searches using the EAC or the Shell in Exchange 2013. Mailbox searches can be created and queued, but they can't be started. An error with an Event ID of 6 is logged in the MsExchange Management log, stating that the **Start-MailboxSearch** cmdlet failed. However, you can search mailboxes using the Shell and the Exchange Control Panel (ECP) in Exchange 2010.

You also have to move the Microsoft Exchange system mailbox to Exchange 2013 as part of upgrading Exchange 2010 Unified Messaging to Exchange 2013.

For more information about upgrading to Exchange 2013, see the following topics:

- Upgrade from Exchange 2010 to Exchange 2013
- Upgrade Exchange 2010 UM to Exchange 2013 UM

What do you need to know before you begin?

- Estimated time to complete: 20 minutes. The actual time may vary depending on the size of the system mailbox.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox Move and Migration Permissions" entry in Recipients Permissions.
- Run the following command in Exchange 2013 to obtain the identity and version of the Exchange servers and mailbox databases that contain the system mailboxes in your organization.

```
Get-Mailbox -Arbitration | FL
Name,DisplayName,ServerName,Database,AdminDisplayVersion
```

The **AdminDisplayVersion** property indicates the version of Exchange that the server is running. The value `Version 14.x` indicates Exchange 2010; the value `Version 15.x` indicates Exchange 2013.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to move the system mailbox

1. In the EAC, go to **Recipients** > **Migration**.
2. Click **New +**, and then click **Move to a different database**.

3. On the **New local mailbox move** page, click **Select the users that you want to move**, and then click **Add +**.
4. On the **Select Mailbox** page, add the mailbox that has the following properties:
   - The display name is **Microsoft Exchange**.
   - The alias of the mailbox's email address is **SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9}**.
5. Click **OK**, and then click **Next**.
6. On the **Move configuration** page, type the name of the migration batch, and then click **Browse** next to the **Target database** box.
7. On the **Select Mailbox Database** page, add the mailbox database to move the system mailbox to. Verify that the version of the mailbox database that you select is Version 15. x, which indicates that the database is located on an Exchange 2013 server.
8. Click **OK**, and then click **Next**.
9. On the **Start the batch** page, select the options to automatically start and complete the migration request, and then click **New**.

Use the Shell to move the system mailbox

First, run the following command in Exchange 2013 to obtain the names and versions of all mailbox databases in your organization.

```
Get-MailboxDatabase -IncludePreExchange2013 | FL
Name,Server,AdminDisplayVersion
```

After you identify the name of the mailbox databases in your organization, run the following command in Exchange 2013 to move the Microsoft Exchange system mailbox to a mailbox database located on an Exchange 2013 server.

```
Get-Mailbox -Arbitration -Identity "SystemMailbox{e0dc1c29-
89c3-4034-b678-e6c29d823ed9}" | New-MoveRequest -
TargetDatabase <name of Exchange 2013 database>
```

How do you know this worked?

To verify that you've successfully moved the Microsoft Exchange system mailbox to a mailbox database located on an Exchange 2013 server, run the following command in the Shell.

```
Get-Mailbox -Arbitration -Identity "SystemMailbox{e0dc1c29-
89c3-4034-b678-e6c29d823ed9}" | FL
Database,ServerName,AdminDisplayVersion
```

If the value of the **AdminDisplayVersion** property is **Version 15.x (Build xxx.x)**, this verifies that the system mailbox resides on a mailbox database that is located on an Exchange 2013 server.

After you move the Microsoft Exchange system mailbox to Exchange 2013, you'll also be able to successfully perform the following administrative tasks:

- Run the **Search-AdminAuditLog** cmdlet.
- Export the administrator audit log in the EAC.
- Successfully create and start eDiscovery searches using the EAC or the Shell in Exchange 2013.

Prepare mailboxes for cross-forest move requests

Mailbox and Client Access servers > Mailbox server > Mailbox moves in Exchange 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-09*

Microsoft Exchange 2013 supports mailbox moves and migrations using the Exchange Management Shell, specifically the **New-MoveRequest** and **New-MigrationBatch** cmdlets. You can also move the mailbox via the Exchange Administration Center (EAC).

To move a mailbox from an Exchange 2013 forest to another Exchange 2013 forest, the Exchange 2013 target forest must contain a valid mail-enabled user with a specified set of Active Directory attributes. If there is at least one Exchange 2013 Client Access server deployed in the forest, the forest is considered an Exchange 2013 forest.

To prepare for the mailbox move, you must create mail-enabled users with the required attributes in the target forest. Here are two recommended approaches to creating mail-enable users with the necessary attributes:

- If you deployed Microsoft Identity Lifecycle Manager (ILM) for cross-forest global address list (GAL) synchronization, the recommended approach to creating the mail-enabled user is to use Service Pack 1 (SP1) for ILM 2007 Feature Pack 1 (FP1). We've created sample code that you can use to learn how to customize ILM to synchronize the source mailbox user and target mail user.

For more information, including how to download the sample code, see Prepare mailboxes for cross-forest moves using sample code.

- If you created the target mail user using an Active Directory tool other than ILM/MIIS, use the **Update-Recipient** cmdlet with the *Identity* parameter to run the Address List service to generate the **LegacyExchangeDN** for the target mail user. We have created a sample Windows PowerShell script that reads from and writes to Active Directory and calls the **Update-Recipient** cmdlet.

For more information about using the sample script, see Prepare mailboxes for cross-forest moves using the Prepare-MoveRequest.ps1 script in the Shell.

After creating the target mail user, you can then run the **New-MoveRequest** or the **New-MigrationBatch** cmdlets to move the mailbox to the target Exchange 2013 forest.

For more information about remote move requests, see the following topics:

- New-MigrationBatch
- New-MoveRequest

For more information about remote mailbox moves and remote legacy moves, see Mailbox moves in Exchange 2013.

The remainder of this topic describes the Active Directory mail user attributes that are required for a mailbox move. These attributes are configured for you when you use either the code or the script to prepare for the mailbox move. However, you can manually copy these attributes using an Active Directory editor.

Active Directory user attributes required for a mailbox move

To support a remote mailbox move, the mail user object in the target Exchange 2013 forest must have the Active Directory attributes that are described in this section:

- Mandatory attributes
- Optional attributes
- Linked attributes
- Linked user attributes
- Resource mailbox attributes
- Additional attributes

Mandatory attributes

The following table lists the minimum set of attributes that need to be configured in ILM on the target mail user for the **New-MoveRequest** cmdlet to function correctly.

Mail user's attributes

Mail user's Active Directory attributes	Action
displayName	Copy the corresponding attribute of the source mailbox or generate a new value.
Mail	Directly copy the corresponding attribute of the source mailbox.
mailNickname	Copy the corresponding attribute of the source mailbox or generate a new value.
msExchArchiveGUID and	Directly copy the corresponding attribute of the

msExchArchiveName	source mailbox. Attributes are only available if the source mailbox is Exchange 2010.
msExchMailboxGUID	Directly copy the corresponding attribute of the source mailbox.
msExchRecipientDisplayType	-2147483642 (decimal) //equivalent to 0x80000006 (hex).
msExchRecipientTypeDetails	128 (decimal) /0x80 (hex).
msExchUserCulture	Directly copy the corresponding attribute of the source mailbox.
msExchVersion	44220983382016 (decimal).
cn	Copy the corresponding attribute of the source mailbox or generate a new value.
proxyAddresses	Copy source mailbox's **proxyAddresses** attribute. Additionally, copy source mailbox's **LegacyExchangeDN** as an X500 address in the **proxyAddresses** attribute of the target mail user. **Note:** The **proxyAddresses** of the source mailbox user must contain an SMTP address that matches the authoritative domain of the target forest. This allows the **New-MoveRequest** cmdlet to correctly select the **targetAddress** of the source mail-enabled user (converted from the source mailbox user after the mailbox move request is complete) to ensure that mail routing is still functional.
sAMAccountName	Copy the corresponding attribute of the source mailbox or generate a new value. Ensure that the value is unique within the target forest domain that the target mail user belongs

	to.
targetAddress	Set to an SMTP address in the **proxyAddresses** attribute of the source mailbox. This SMTP address must belong to the authoritative domain of the source forest.
userAccountControl	Constant: 514 //equivalent to 0x202, ACCOUNTDISABLE | NORMAL_ACCOUNT.
userPrincipalName	Copy the corresponding attribute of the source mailbox or generate a new value. Because the mail user is logon disabled, this **userPrincipalName** isn't used.

Optional attributes

It isn't mandatory that the following attributes are configured for the **New-MoveRequest** cmdlet to function correctly; however, synchronizing them provides a better end-to-end user experience after moving the mailbox. Because the GAL in the target forest displays this target mail user, you should set the following GAL-related attributes.

GAL-related attributes

Mail user's Active Directory attributes	Action
c	Directly copy the corresponding attribute of the source mailbox.
co	Directly copy the corresponding attribute of the source mailbox.
countryCode	Directly copy the corresponding attribute of the source mailbox.
company	Directly copy the corresponding attribute of the source mailbox.
department	Directly copy the corresponding attribute of the source mailbox.

facsimileTelephoneNumber	Directly copy the corresponding attribute of the source mailbox.
givenName	Directly copy the corresponding attribute of the source mailbox.
homePhone	Directly copy the corresponding attribute of the source mailbox.
info	Directly copy the corresponding attribute of the source mailbox.
initials	Directly copy the corresponding attribute of the source mailbox.
l	Directly copy the corresponding attribute of the source mailbox.
mobile	Directly copy the corresponding attribute of the source mailbox.
msExchAssistantName	Directly copy the corresponding attribute of the source mailbox.
msExchHideFromAddressLists	Directly copy the corresponding attribute of the source mailbox.
otherHomePhone	Directly copy the corresponding attribute of the source mailbox.
otherTelephone	Directly copy the corresponding attribute of the source mailbox.
pager	Directly copy the corresponding attribute of the source mailbox.
physicalDeliveryOfficeName	Directly copy the corresponding attribute of the source mailbox.

postalCode	Directly copy the corresponding attribute of the source mailbox.
sn	Directly copy the corresponding attribute of the source mailbox.
st	Directly copy the corresponding attribute of the source mailbox.
streetAddress	Directly copy the corresponding attribute of the source mailbox.
telephoneAssistant	Directly copy the corresponding attribute of the source mailbox.
telephoneNumber	Directly copy the corresponding attribute of the source mailbox.
title	Directly copy the corresponding attribute of the source mailbox.

Linked attributes

A linked attribute is an Active Directory attribute that references other Active Directory objects in the local forest. You can't directly copy the linked attribute values from a mailbox in the source forest to a mail user in the target forest. First, you must find the Active Directory objects in the source forest that the source mailbox attribute refers to. Then, you must find the corresponding Active Directory objects in the target forest for the above-mentioned Active Directory object in the source forest. And finally, set the target mail user's attribute to refer to the Active Directory objects in the target forest.

Linked attributes

Mail User's Active Directory attributes	Action
altRecipient	Correspond to the source mailbox's **altRecipient** attribute.
deliverAndRedirect	Directly copy the corresponding attribute of the source mailbox. This attribute is a Boolean value that should be set along with

	altRecipient.
Manager (and its backlinks)	Correspond to the source mailbox's manager attribute.
MemberOf (backlinks)	This is the backlink of group member attribute.
publicDelegates (and its backlinks)	Correspond to the source mailbox's **publicDelegates** attribute.

Linked user attributes

If you want to move a mailbox to an Exchange 2013 resource forest, the mailbox in the resource forest is considered a *linked mailbox*. In this scenario, you need to create a linked mail user in the (target) resource forest. To create a linked mail user, you need to set the attributes shown in the following table.

Linked mail user attributes

Mail user's Active Directory attributes	Action
msExchMasterAccountHistory	Directly copy the corresponding attribute of the source mailbox.
msExchMasterAccountSid	If the source mailbox has **msExchMasterAccountSid**, copy it. Otherwise, copy the source mailbox's **objectSid**.
msExchRecipientDisplayType	Constant:-1073741818 (decimal) //equivalent to *unsigned* 0xC0000006.

Note:
A linked mailbox can only be created if there's forest trust between the source forest and target forest.

If the source object is disabled and the **msExchMasterAccountSid** attribute is set to self (resource mailbox, shared mailbox), don't stamp anything on the target user.

If the source object is disabled and the **msExchMasterAccountSid** attribute isn't set, the mailbox is invalid.

If the source object is enabled and the **msExchMasterAccountSid** attribute is set, the mailbox is

invalid.

Resource mailbox attributes

If you want to move a resource mailbox to an Exchange 2013 forest, you need to set the attributes shown in the following table on the target mail user.

Resource mailbox attributes

Mail user's Active Directory attributes	Action

msExchRecipientDisplayType	If the source mailbox is a conference room:
	• **Constant** -2147481850 (decimal) // equivalent to *unsigned* 0x80000706.
	If the source mailbox is an equipment mailbox:
	• **Constant** -2147481594 (decimal) // equivalent to *unsigned* 0x80000806.

msExchResourceCapacity	Directly copy the corresponding attribute of the source mailbox.
msExchResourceDisplay	Directly copy the corresponding attribute of the source mailbox.
msExchResourceMetaData	Directly copy the corresponding attribute of the source mailbox.
msExchResourceSearchProperties	Directly copy the corresponding attribute of the source mailbox.

Additional attributes

In Exchange 2007, the **Move-Mailbox** cmdlet also copied the attributes shown in the following table when moving a mailbox. You can optionally copy these attribute if required by your organization.

Resource mailbox attributes

Mail User's Active Directory attributes	Description
comment	Directly copy the corresponding attribute of the source mailbox.

deletedItemFlags	Directly copy the corresponding attribute of the source mailbox.
delivContLength	Directly copy the corresponding attribute of the source mailbox.
departmentNumber	Directly copy the corresponding attribute of the source mailbox.
description	Directly copy the corresponding attribute of the source mailbox.
division	Directly copy the corresponding attribute of the source mailbox.
employeeID	Directly copy the corresponding attribute of the source mailbox.
employeeNumber	Directly copy the corresponding attribute of the source mailbox.
employeeType	Directly copy the corresponding attribute of the source mailbox.
extensionAttribute1-15	Directly copy the corresponding attribute of the source mailbox.
homePostalAddress	Directly copy the corresponding attribute of the source mailbox.
internationalISDNNumber	Directly copy the corresponding attribute of the source mailbox.
ipPhone	Directly copy the corresponding attribute of the source mailbox.
language	Directly copy the corresponding attribute of the source mailbox.

lmPwdHistory	Directly copy the corresponding attribute of the source mailbox.
localeID	Directly copy the corresponding attribute of the source mailbox.
mAPIRecipient	Directly copy the corresponding attribute of the source mailbox.
middleName	Directly copy the corresponding attribute of the source mailbox.
msDS-PhoneticCompanyName	Directly copy the corresponding attribute of the source mailbox.
msDS-PhoneticDepartment	Directly copy the corresponding attribute of the source mailbox.
msDS-PhoneticDisplayName	Directly copy the corresponding attribute of the source mailbox.
msDS-PhoneticFirstName	Directly copy the corresponding attribute of the source mailbox.
msDS-PhoneticLastName	Directly copy the corresponding attribute of the source mailbox.
msExchBlockedSendersHash	Directly copy the corresponding attribute of the source mailbox.
msExchELCExpirySuspensionEnd	Directly copy the corresponding attribute of the source mailbox.
msExchELCExpirySuspensionStart	Directly copy the corresponding attribute of the source mailbox.
msExchELCMailboxFlags	Directly copy the corresponding attribute of the source mailbox.

msExchExternalOOFOptions	Directly copy the corresponding attribute of the source mailbox.
msExchMessageHygieneFlags	Directly copy the corresponding attribute of the source mailbox.
msExchMessageHygieneSCLDeleteThreshold	Directly copy the corresponding attribute of the source mailbox.
msExchMessageHygieneSCLJunkThreshold	Directly copy the corresponding attribute of the source mailbox.
msExchMessageHygieneSCLQuarantineThreshold	Directly copy the corresponding attribute of the source mailbox.
msExchMessageHygieneSCLRejectThreshold	Directly copy the corresponding attribute of the source mailbox.
msExchMDBRulesQuota	Directly copy the corresponding attribute of the source mailbox.
msExchPoliciesExcluded	Directly copy the corresponding attribute of the source mailbox.
msExchSafeRecipientsHash	Directly copy the corresponding attribute of the source mailbox.
msExchSafeSendersHash	Directly copy the corresponding attribute of the source mailbox.
msExchUMSpokenName	Directly copy the corresponding attribute of the source mailbox.
otherFacsimileTelephoneNumber	Directly copy the corresponding attribute of the source mailbox.
otherIpPhone	Directly copy the corresponding attribute of the source mailbox.

otherMobile	Directly copy the corresponding attribute of the source mailbox.
otherPager	Directly copy the corresponding attribute of the source mailbox.
preferredDeliveryMethod	Directly copy the corresponding attribute of the source mailbox.
personalPager	Directly copy the corresponding attribute of the source mailbox.
personalTitle	Directly copy the corresponding attribute of the source mailbox.
photo	Directly copy the corresponding attribute of the source mailbox.
pOPCharacterSet	Directly copy the corresponding attribute of the source mailbox.
pOPContentFormat	Directly copy the corresponding attribute of the source mailbox.
postalAddress	Directly copy the corresponding attribute of the source mailbox.
postOfficeBox	Directly copy the corresponding attribute of the source mailbox.
primaryInternationalISDNNumber	Directly copy the corresponding attribute of the source mailbox.
primaryTelexNumber	Directly copy the corresponding attribute of the source mailbox.
showInAdvancedViewOnly	Directly copy the corresponding attribute of the source mailbox.

street	Directly copy the corresponding attribute of the source mailbox.
terminalServer	Directly copy the corresponding attribute of the source mailbox.
textEncodedORAddress	Directly copy the corresponding attribute of the source mailbox.
thumbnailLogo	Directly copy the corresponding attribute of the source mailbox.
thumbnailPhoto	Directly copy the corresponding attribute of the source mailbox.
url	Directly copy the corresponding attribute of the source mailbox.
userCert	Directly copy the corresponding attribute of the source mailbox.
userCertificate	Directly copy the corresponding attribute of the source mailbox.
userSMIMECertificate	Directly copy the corresponding attribute of the source mailbox.
wWWHomePage	Directly copy the corresponding attribute of the source mailbox.

Prepare mailboxes for cross-forest moves using the Prepare-MoveRequest.ps1 script in the Shell

requests >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-30*

Microsoft Exchange 2013 supports mailbox moves and migrations using the **New-MoveRequest** and **New-MigrationBatch** cmdlets. You can also move the mailbox via the Exchange Administration Center (EAC). You can move a mailbox from a source Exchange forest to a target Exchange 2013 forest.

To run the **New-MoveRequest** and **New-MigrationBatch** cmdlets, a mail user must exist in the target Exchange forest, and the mail user must have a minimum set of required Active Directory attributes.

The sample Windows PowerShell script described in this topic supports this task by synchronizing mailbox users from an Exchange 2013 source forest to Exchange 2013 target forests as mail-enabled users. The script copies the Active Directory attributes of the mailbox users in the source forest to the target forest, and then uses the **Update-Recipient** cmdlet to turn the target objects into mail-enabled users.

For more information about using and writing scripts, see Scripting with the Exchange Management Shell. For more information about preparing for cross-forest moves, see Prepare mailboxes for cross-forest move requests.

Looking for other management tasks related to remote move requests? Check out Manage on-premises moves.

What do you need to know before you begin?

- Locate the script in the following location: Program Files\Microsoft\Exchange Server\V15\Scripts
- To run the sample script, you need the following:
  - A source forest running Exchange 2013, where the mailbox currently resides.
  - A target forest with Exchange 2013 installed, where the mailbox will be moved to.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Prepare-MoveRequest.ps1 script to prepare mailboxes for cross-forest moves

Run the script from the Shell on a server role running Exchange 2013 in the target Exchange 2013 forest. The script copies the mailbox attributes from the source forest.

To assign a specific authentication credential for the remote forest domain controller, you must first run the Windows PowerShell **Get-Credential** cmdlet and store the user input in a temporary

variable. When you run the **Get-Credential** cmdlet, the cmdlet asks for the user name and password of the account used during authentication with the remote forest domain controller. You can then use the temporary variable in the Prepare-MoveRequest.ps1 script. For more information about the **Get-Credential** cmdlet, see Get-Credential.

Note:
Make sure that you use two separate credentials for the local forest and the remote forest when calling this script.

1. Run the following commands to get the local forest and remote forest credentials.

```
$LocalCredentials = Get-Credential
$RemoteCredentials = Get-Credential
```

2. Run the following commands to pass the credential information to the *LocalForestCredential* and *RemoteForestCredential* parameters in the Prepare-MoveRequest.ps1 script.

```
Prepare-MoveRequest.ps1 -Identity JohnSmith@Fabrikan.com -
RemoteForestDomainController DC001.Fabrikam.com -
RemoteForestCredential $RemoteCredentials -
LocalForestDomainController DC001.Contoso.com -
LocalForestCredential $LocalCredentials
```

Parameter set of the script

The following table describes the parameter set for the script.

Parameter set of the Prepare-MoveRequest.ps1 script

Parameter	Required	Description
Identity	Required	The *Identity* parameter uniquely identifies a mailbox in the source forest. Identity can be any of the following: • Common name (CN) • Alias • **proxyAddress** property • **objectGuid** property • **DisplayName** property
RemoteForestCredential	Required	The *RemoteForestCredential* parameter specifies the

		administrator who has permissions to copy data from the source forest Active Directory.
RemoteForestDomainController	Required	The *RemoteForestDomainController* parameter specifies a domain controller in the source forest where the mailbox resides.
DisableEmailAddressPolicy	Optional	The *DisableEmailAddressPolicy* parameter specifies whether the Email Address Policy (EAP) should be disabled when creating a **MailUser** object in the target forest. When you specify this parameter, the EAP in the target forest won't be applied. **Note:** When you specify this parameter, the **MailUser** object won't have e-mail address mapping in the local forest domain stamped. This is usually stamped by the EAP.
LinkedMailUser	Optional	The *LinkedMailUser* switch specifies whether to create a linked MailUser in the local forest for the mailbox user in the remote forest. If the switch is provided, the script creates a target **MailUser** object linked to the source mailbox. If the switch is omitted, the script

		creates a regular target **MailUser** object.
LocalForestCredential	Optional	The *LocalForestCredential* parameter specifies the administrator with permissions to write data to the target forest Active Directory. We recommend that you explicitly specify this parameter to avoid Active Directory permission issues. If the remote forest and the local forest have a trusted relationship configured, don't use a user account from the remote forest as the local forest credential, even though the remote user account may have permission to modify Active Directory in the local forest.
LocalForestDomainController	Optional	The *LocalForestDomainController* parameter specifies a domain controller in the target forest where the mail-enabled user will be created. We recommend that you specify this parameter to avoid possible domain controller replication delay issues in the local forest that could occur if a random domain controller is selected.
MailboxDeliveryDomain	Optional	The *MailboxDeliveryDomain*

		parameter specifies an authoritative domain of the source forest so that the script can select the correct source mailbox user's **proxyAddress** property as the target mail-enabled user's **targetAddress** property. By default, the primary SMTP address of the source mailbox user is set as the **targetAddress** property of the target mail-enabled user.
OverWriteLocalObject	Optional	The *OverWriteLocalObject* parameter is used for users created by the Active Directory Migration Tool. The properties are copied from the existing mail contact to the newly created mail user. However, after this copy, the script also copies the properties from the source forest user to the newly created mail user.
TargetMailUserOU	Optional	The *TargetMailuserOU* parameter specifies the organizational unit (OU) under which the target mail-enabled user will be created.
UseLocalObject	Optional	The *UseLocalObject* parameter specifies whether to convert the existing local object to the required target mail-enabled user

		if the script detects an object in the local forest that conflicts with the to-be-created mail-enabled user.

Examples

This section contains several examples of how you can use the Prepare-MoveRequest.ps1 script.

Example: Single linked mail-enabled user

This example provisions a single linked mail-enabled user in the local forest, when there is forest trust between the remote forest and local forest.

1. Run the following commands to get the local forest and remote forest credentials.

```
$LocalCredentials = Get-Credential
$RemoteCredentials = Get-Credential
```

2. Run the following command to pass the credential information to the *LocalForestCredential* and *RemoteForestCredential* parameters in the Prepare-MoveRequest.ps1 script.

```
Prepare-MoveRequest.ps1 -Identity JamesAlvord@Contoso.com -
RemoteForestDomainController DC001.Fabrikam.com -
RemoteForestCredential $RemoteCredentials -
LocalForestDomainController DC001.Contoso.com -
LocalForestCredential $LocalCredentials -LinkedMailUser
```

Example: Pipelining

This example supports pipelining if you supply a list of mailbox identities.

1. Run the following command.

```
$UserCredentials = Get-Credential
```

2. Run the following command to pass the credential information to the *RemoteForestCredential* parameter in the Prepare-MoveRequest.ps1 script.

```
"IanP@Contoso.com", "JoeAn@Contoso.com" | Prepare-
MoveRequest.ps1 -RemoteForestDomainController
DC001.Fabrikam.com -RemoteForestCredential $UserCredentials
```

Example: Use a .csv file to bulk-create mail-enabled users

You can generate a .csv file containing a list of mailbox identities from the source forest, which allows you to pipe the content of this file into the script to bulk-create the target mail-enabled users.

For example, the content of the .csv file can be:

Identity

Ian@contoso.com

John@contoso.com

Cindy@contoso.com

This example calls a .csv file to bulk create the target mail-enabled users.

1. Run the following command to get the remote forest credentials.

```
$UserCredentials = Get-Credential
```

2. Run the following command to pass the credential information to the *RemoteForestCredential* parameter in the Prepare-MoveRequest.ps1 script.

```
Import-Csv Test.csv | Prepare-MoveRequest.ps1 -
RemoteForestDomainController DC001.Fabrikam.com -
RemoteForestCredential $UserCredentials
```

Script behavior per target object

This section describes how the script performs in relation to several scenarios for target objects.

Duplicate target mail-enabled object

When the script attempts to create a target mail-enabled user from the source mailbox user, and it detects a duplicate local mail-enabled object, it uses the following logic:

- If the source mailbox user's **masterAccountSid** attribute equals any target object's **objectSid** or **masterAccountSid** attribute:
  - If the target object isn't mail-enabled, the script returns an error because the script doesn't support converting an object that isn't mail-enabled to a mail-enabled user.
  - If the target object is mail-enabled, the target object is a duplicate.
- If an address in the source mailbox user's **proxyAddress** properties (smtp/x500 only) equals an address in a target object's **proxyAddress** properties (smtp/x500 only), the target object is a duplicate.

The script prompts the user about the duplicate objects.

If the target mail-enabled object is a mail-enabled user or contact, which is most likely created by a cross-forest (Identity Lifecycle Management 2007 Service Pack 1-based) global address list (GAL) synchronization deployment, the user can run the script again with the *UseLocalObject* parameter to use the target mail-enabled object for mailbox migration.

Mail-enabled user

If the target object is a mail-enabled user, the script copies the following attributes from the source mailbox user to the target mail-enabled user:

- **msExchMailboxGUID**
- **msExchArchiveGUID**
- **msExchArchiveName**

If the *LinkedMailUser* parameter is set, the script copies the source **objectSid**/**masterAccountSid** attribute.

Mail-enabled contact

If the target object is a mail-enabled contact, the script deletes the existing contact and copies all its attributes to a new mail-enabled user. The script also copies the following attributes from the source mailbox user:

- **msExchMailboxGUID**
- **msExchArchiveGUID**
- **msExchArchiveName**
- **sAMAccountName**
- **userAccountControl** (set to 514 //equivalent to 0x202, ACCOUNTDISABLE | NORMAL_ACCOUNT)
- **userPrincipalName**

If the *LinkedMailUser* parameter is set, the script copies the source **objectSid**/**masterAccountSid** attribute.

LegacyExchangeDN attribute

When the **Update-Recipient** cmdlet is called to convert the target object into a mail-enabled user, a new **LegacyExchangeDN** attribute is generated for the target mail-enabled user. The script copies the **LegacyExchangeDN** attribute of the target mail-enabled user as an x500 address to the **proxyAddress** properties of the source mailbox user.

Prepare mailboxes for cross-forest

moves using sample code

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-03-15*

Microsoft Exchange 2013 supports mailbox moves and migrations using the **New-MoveRequest** and **New-MigrationBatch** cmdlets. You can also move the mailbox via the Exchange admin center (EAC). You can move a mailbox from a source Exchange forest to a target Exchange 2010 forest.

To run **New-MoveRequest**, a mail user must exist in the target Exchange forest and the mail user must have a minimum set of required Active Directory attributes. You can create the required mail user in the target Exchange forest by customizing your Microsoft Identity Lifecycle Manager (ILM) 2007 deployment. The ILM-based rules extension sample code described in this topic demonstrates how to customize your current ILM deployment to create the required mail-enabled users in the target Exchange 2013 forest.

For more information about preparing for cross-forest moves, including descriptions of the required Active Directory attributes, see Prepare mailboxes for cross-forest move requests.

What do you need to know before you begin?

- Download the sample code from the Prepare for Online Mailbox Move page in the Microsoft Download Center.
- To run the sample code, you need ILM 2007 Feature Pack 1 Service Pack 1 (SP1). To download the feature pack, see Microsoft Knowledge Base article 977791, Service Pack 1 (build 3.3.1139.2) is available for Identity Lifecycle Manager 2007 Feature Pack 1.
- You also need the following:
  - A source forest running Exchange 2013, where the mailbox currently resides.
  - A target forest with Exchange 2013 installed, where the mailbox will be moved to.
- To connect to the Exchange 2013 target forest, you must have the appropriate permission to call the **UpdateRecipient** cmdlet. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Install the ILM sample code

1. In Microsoft Visual Studio 2008, open Microsoft.Exchange.Sample.OneWayGALSync.sln to view the sample code. The sample code includes the following:
   - Microsoft.MetadirectoryServicesEx.dll is the binary file that is shipped with ILM 2007 FP1 SP1 under "\Program Files\Microsoft Identity Integration Server\Bin\Assemblies". It's referenced by the sample code.
   - OneWaySync.xml is referenced by the sample code.
   - The ILMServerConfig folder contains the ILM configuration files for the source management agent (MA), target MA, and the ILM Metaverse (MV).
   - Microsoft.Exchange.Sample.OneWayGALSync.MARules.dll and Microsoft.Exchange.Sample.OneWayGALSync.MVRules.dll (built from the sample code) are under "\obj\Debug"
2. On the ILM server, copy the following to \Program Files\Microsoft Identity Integration Server \Extensions:
   - OneWaySync.xml
   - Microsoft.Exchange.Sample.OneWayGALSync.MARules.dll
   - Microsoft.Exchange.Sample.OneWayGALSync.MVRules.dll
3. Edit the file OneWaySync.xml that you copied to the ILM Extensions folder in step 1 to specify the distinguishedName (DN) of the TargetOU container in the target Exchange forest in which you want to create the mail users. You can use LDP.exe or ADSIEdit.exe to browse for the TargetOU container if you don't know what its name is.

Note:

If you're using this sample together with ILM GalSync 2007 exclude this container from the list of containers managed by GalSync2007.

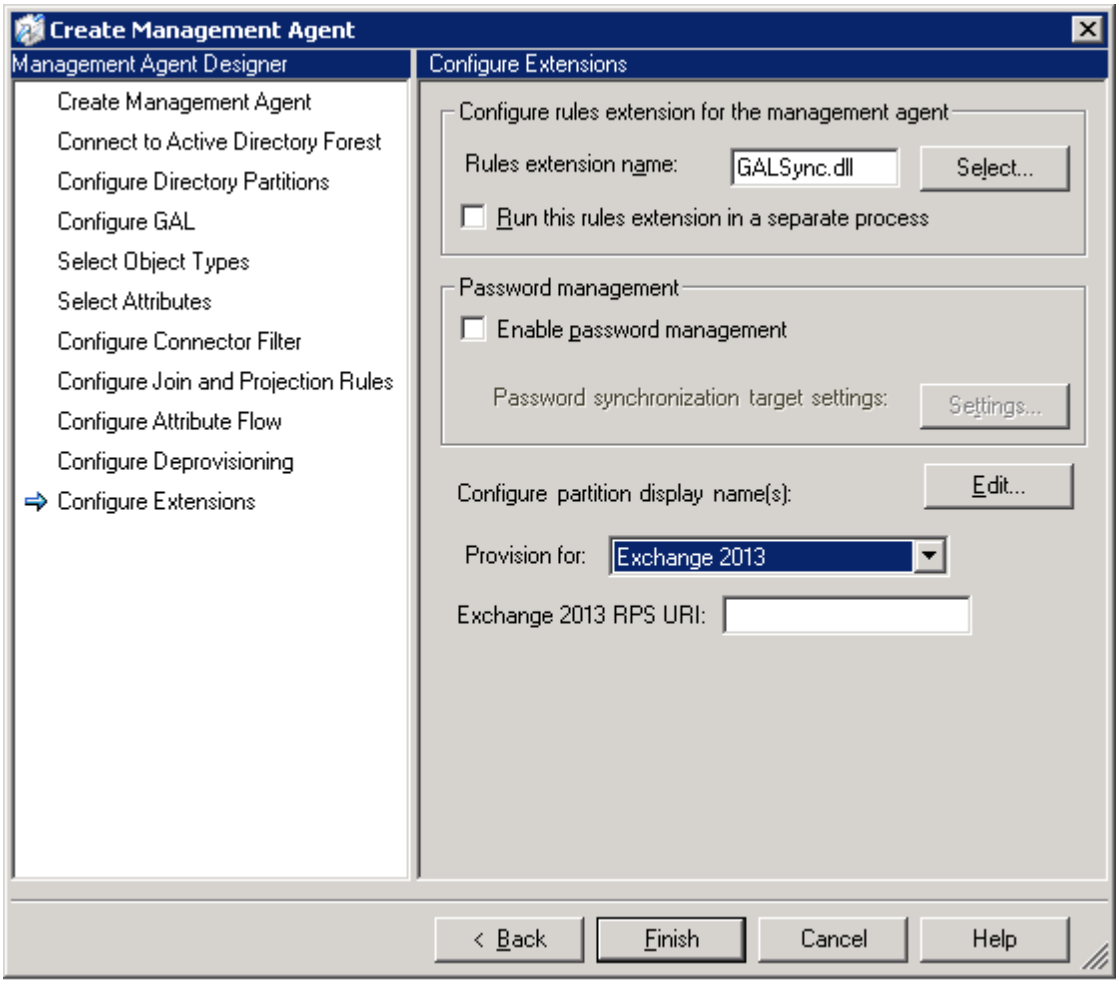
4. On the ILM Identity Manager Console, go to **File** > **Import Server Configuration** to import the ILM server configuration from the folder ILMServerConfig. This action will import two Active Directory Management Agents along with the Metaverse schema and the provisioning rule.

Note:

During the import, you must provide the forest name and credentials and match the partitions of the imported Active Directory Management Agent (ADMA) to the partition name in your configuration for both the source and target ADMAs.

5. For the ADMA to support the Exchange 2013 target forest, on the **Create Management Agent** page, on the **Configure Extensions** pane, select **Exchange 2013** in the **Provision for** drop-down and then enter the remote Windows PowerShell URI of an Exchange 2010 Client Access server in **Exchange 2013 RPS URI**.

Create Management Agent page



6. On the ILM Identity Manager Console on the **Create Management Agent** pane, open the **Properties** for the Source Forest Management Agent. Select the **Configure Directory Partitions** wizard, and then click **Containers** to select the container that will contain the mailboxes you will be moving to the target forest. Clear the selections for all other containers, that is, scope the management agent to only manage this one container. Similarly, for the target forest MA, select the container to which mail-enabled users will be provisioned, that is, the TargetOU specified in step 2.

Note:

If you're using this sample together with ILM GalSync 2007, exclude both of these containers from the list of containers managed by GalSync 2007.

7. Perform an initial Full Import (stage only) on the target MAs so that ILM can discover the TargetOU specified in step 2.

Step 2: Create mail user in target Exchange forest

Now that you've installed the sample code, use the following procedure to create the required mail user in the target Exchange forest so that **New-MoveRequest** can be run to perform an online mailbox move.

1. In the source forest, use the Exchange admin center to create mailbox users in the container selected in step 4 of "Install the ILM sample code". You can also use Active Directory Users and Computers to move existing mailbox users to the container.
2. Perform Delta Import and Delta Sync run on the source MA to discover the mailboxes added to

the source container, and provision mail users to the target MA.
3. Perform Export run on the target MA to export the mail users provisioned in step 1 to the target Active Directory.
4. Perform Delta Import on the target MA to confirm the changes exported in step 2.
5. In the target forest, open the Exchange Management Shell and use the **New-MoveRequest** cmdlet to move mailboxes from the source forest.

How do you know this worked?

To verify that you have successfully completed your migration, do the following:

- From the target forest, verify that the users that you moved from the source forest are present in the target forest.

Enable the MRS Proxy endpoint for remote moves

Mailbox and Client Access servers > Mailbox server > Mailbox moves in Exchange 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-07-02*

The Mailbox Replication Service Proxy (MRS Proxy) facilitates cross-forest mailbox moves and remote move migrations between your on-premises Exchange organization and Exchange Online. In Exchange 2013, MRS Proxy is included in the Mailbox server role (also called the *Mailbox server*). During cross-forest and remote move migrations, a Client Access server acts as a proxy for incoming move requests for the Mailbox server. The ability of a Client Access server to accept these requests is disabled by default. To allow the Client Access server to accept incoming move requests, you have to enable the MRS Proxy endpoint.

The Client Access server on which to enable the MRS Proxy endpoint depends on the type and direction of the mailbox move.

- **Cross-forest enterprise moves** For cross-forest moves that are initiated from the target environment (known as a *pull* move type), you have to enable the MRS Proxy endpoint on Client Access servers in the source environment. For cross-forest moves that are initiated from the source environment (known as a *push* move type), you have to enable the MRS Proxy endpoint on Client Access servers in the target environment.
- **Remote move migrations between an on-premises Exchange organization and Exchange Online** For both onboarding and offboarding remote move migrations, you have to enable the MRS Proxy endpoint on Client Access servers in your on-premises organization.

Note:

If you use the EAC to move mailboxes, cross-forest moves and onboarding remote move migrations are pull move types because the request is initiated from the target environment. Offboarding remote move migrations are a push move type because the request is initiated from the source environment.

What do you need to know before you begin?

- Estimated time to complete: 2 minutes per server.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange Web Services permissions" section in the Clients and mobile devices permissions topic.
- If you've deployed more than one Client Access server in your Exchange organization, you should enable the MRS Proxy endpoint on each one. If you add additional Client Access servers, be sure to enable the MRS Proxy endpoint on the new servers. Cross-forest moves and remote move migrations can fail if the MRS Proxy endpoint isn't enabled on all Client Access servers.
- If you don't perform cross-forest moves or remote move migrations, keep MRS Proxy endpoints disabled to reduce the attack surface of your organization.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to enable the MRS Proxy endpoint

1. In the EAC, navigate to **Recipients** > **Servers** > **Virtual Directories**.
2. In the **Select server** drop-down list, select the name of the Client Access server on which you want to enable the MRS Proxy endpoint. Or select **All servers** to display the virtual directories on all Client Access servers in your organization.
3. In the **Select type** drop-down list, select **EWS** to display the Exchange Web Service (EWS) virtual directory for the selected server.
4. In the list of virtual directories, click **EWS (Default Web Site)** for the Client Access server that you want to configure, and then click **Edit** .
5. On the **EWS (Default Web Site)** properties page, select the **Enable MRS Proxy endpoint** check box, and then click **Save**.

Use the Shell to enable the MRS Proxy endpoint

The following command enables the MRS Proxy endpoint on a Client Access server named EXCH-SRV-01.

```
Set-WebServicesVirtualDirectory -Identity "EXCH-SRV-01\EWS
(Default Web Site)" -MRSProxyEnabled $true
```

The following command enables the MRS Proxy endpoint on all Client Access servers in your Exchange organization.

```
Get-WebServicesVirtualDirectory | Set-
WebServicesVirtualDirectory -MRSProxyEnabled $true
```

Important:
As previously stated, the MRS Proxy endpoint should be enabled on each Client Access server in your organization. Run the preceding command after adding a new Client Access server to your organization.

How do you know this worked?

To verify that you've successfully enabled the MRS Proxy endpoint, do one of the following:

1. In the EAC, navigate to **Recipients** > **Servers** > **Virtual Directories**.
2. In the list of virtual directories, click **EWS (Default Web Site)** and verify in the details pane that the MRS Proxy endpoint is enabled.

Alternatively, you can click **Edit** to view the **EWS (Default Web Site)** properties page and verify that the **Enable MRS Proxy endpoint** check box is selected.

Or

Run the following command in the Shell:

```
Get-WebServicesVirtualDirectory | FL
Identity,MRSProxyEnabled
```

Verify that the *MRSProxyEnabled* parameter is set to `True`.

Another way to verify that the MRS Proxy endpoint is enabled is to use the **Test-MigrationServerAvailability** cmdlet to test the ability to communicate with the remote server that hosts the mailboxes that you want to move, or in the case of offboarding Exchange Online mailboxes to your on-premises organization, a server in your on-premises organization. For more information, see Test-MigrationServerAvailability.

The following example tests the connection to a server in the corp.contoso.com forest.

```
$Credentials = Get-Credential

Test-MigrationServerAvailability -ExchangeRemoteMove -
Autodiscover -EmailAddress administrator@corp.contoso.com -
Credentials $Credentials
```

To run this command successfully, the MRS Proxy endpoint must be enabled.

CSV files for mailbox migration

Mailbox and Client Access servers > Mailbox server > Mailbox moves in Exchange 2013 >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-04-25*

You can use a CSV file to bulk migrate a large number of user mailboxes. You can specify a CSV file when you use the Exchange admin center (EAC) or the **New-MigrationBatch** cmdlet in the Exchange Management Shell to create a migration batch. Using a CSV to specify multiple users to migrate in a migration batch is supported in the following migration scenarios:

- **Moves in on-premises Exchange organizations**
  - **Local move:** A local move is where you move mailboxes from one mailbox database to another. A local move occurs within a single forest.
  - **Cross-forest enterprise move:** In a cross-forest enterprise move, mailboxes are moved to a different forest. Cross-forest moves are initiated either from the target forest, which is the forest that you want to move the mailboxes to, or from the source forest, which is the forest that currently hosts the mailboxes.
- **Onboarding and offboarding in Exchange Online**
  - **Onboarding remote move migration:** In an Exchange hybrid deployment, you can move mailboxes from an on-premises Exchange organization to Exchange Online. This is also known as an *onboarding* remote move migration because you onboard mailboxes to Exchange Online.
  - **Offboarding remote move migration:** You can also perform an *offboarding* remote move migration, where you migrate Exchange Online mailboxes to your on-premises Exchange organization.

Note:
Both onboarding and offboarding remote move migrations are initiated from your Exchange Online organization.

  - **Staged Exchange migration:** You can also migrate a subset of mailboxes from an on-premises Exchange organization to Exchange Online. This is another type of onboarding migration. You can migrate only Exchange 2003 and Exchange 2007 mailboxes using a staged Exchange migration. Migrating Exchange 2010 and Exchange 2013 mailboxes isn't supported using a staged migration. Prior to running a staged migration, you have to use directory synchronization or some other method to provision mail users in your Exchange Online organization.
  - **IMAP migration:** This onboarding migration type migrates mailbox data from an IMAP server (including Exchange) to Exchange Online. For an IMAP migration, you must provision mailboxes in Exchange Online before you can migrate mailbox data.

Note:

A cutover Exchange migration doesn't support a using a CSV file because all on-premises user mailboxes are migrated to Exchange Online in a single batch.

Supported attributes for CSV files for bulk moves or migrations

The first row, or header row, of a CSV file used for migrating users lists the names of the attributes, or fields, specified on the rows that follow. Each attribute name is separated by a comma. Each row under the header row represents an individual user and supplies the information required for the migration. The attributes in each individual user row must be in the same order as the attribute names in the header row. Each attribute value is separated by a comma. If the attribute value for a particular record is null, don't type anything for that attribute. However, make sure that you include the comma to separate the null value from the next attribute.

Attribute values in the CSV file override the value of the corresponding parameter when that same parameter is used when creating a migration batch with the EAC or the Shell. For more information and examples, see the section Attribute values in the CSV file override the values for the migration batch.

Tip:

You can use any text editor to create the CSV file, but using an application like Microsoft Excel will make it easier to import data and configure and organize CSV files. Be sure to save CSV files as a .csv or .txt file.

The following sections describe the supported attributes for the header row of a CSV file for each migration type. Each section includes a table that lists each supported attribute, whether it's required, an example of a value to use for the attribute, and a description.

Note:

In the following sections, *source environment* denotes the current location of a user mailbox or a database. *Target environment* denotes the location that the mailbox will be migrated to or the database that the mailbox will be moved to.

Local moves

The following table describes the supported attributes for a CSV file for local moves. For more information, see Manage on-premises moves.

Attribute	Required or optional	Accepted values	Description
EmailAddress	Required	SMTP address for the user	Specifies the user that will be moved.

TargetDatabase	Optional	Database name	Specifies the mailbox database that the user's primary mailbox will be moved to. You can specify a different database in the different rows of the CSV file, which lets you move mailboxes in the same migration batch to different databases.
TargetArchiveDatabase	Optional	Database name	Specifies the mailbox database that the user's archive mailbox will be moved to. You can specify a different database in the different rows of the CSV file, which lets you move archive mailboxes in the same migration batch to different databases. **Note:** If you specify a specific archive database, the archive mailbox (if it exists) will be moved to the same database as the primary mailbox.
BadItemLimit	Optional	`Unlimited` or a non-negative integer from `0` (the default) to a maximum value of	Specifies the number of bad items to skip if the migration service

		2147483647	encounters a corrupted item in the mailbox. If you include this attribute in the CSV file, it will override the default value or a value you specify if you include the *BadItemLimit* parameter when creating the migration batch using the EAC or the Shell. **Tip:** We recommend that you use the default value of 0 and only increase the bad item limit for a particular user if the move or migration for that user fails.
MailboxType	Optional	Use one of the following values: • `PrimaryOnly` • `ArchiveOnly` • `PrimaryAndArchive` (the default value)	Specifies whether to move the user's primary mailbox, archive mailbox, or both.

Onboarding remote move migrations in a hybrid deployment

In a hybrid deployment, you can move mailboxes from an on-premises Exchange organization to Exchange Online. When onboarding mailboxes, the migration batch is created in the Exchange Online organization and initiated by an Exchange Online administrator. For more information, see **Move mailboxes between on-premises and Exchange Online organizations in 2013 hybrid deployments**.

The following table describes the supported attributes for a CSV file for onboarding remote move migrations.

Attribute	Required or optional	Accepted values	Description
EmailAddress	Required	SMTP address for the user	Specifies the email address for the mail-enabled user in the Exchange Online organization that corresponds to the on-premises user mailbox that will be migrated.
BadItemLimit	Optional	`Unlimited` or a non-negative integer from `0` (the default) to a maximum value of `2147483647`	Specifies the number of bad items to skip if the migration service encounters a corrupted item in the mailbox. If you include this attribute in the CSV file, it will override the default value or the value you specify if you include the *BadItemLimit* parameter when creating the migration batch using the EAC or the Shell. **Tip:** We recommend that you use the default value of 0 and only increase the bad item limit for a particular user if the move or

			migration for that user fails.
LargeItemLimit	Optional	`Unlimited` or a non-negative integer from `0` (the default) to a maximum value.	Specifies the number of large items in the user's mailbox that will be skipped. When the number of large items exceeds this value, the migration for the mailbox fails. The default value is 0, which means that the migration fails if the mailbox contains any large items. When onboarding mailboxes to Exchange Online, items up to 35 MB are migrated.
MailboxType	Optional	Use one of the following values: • `PrimaryOnly` • `ArchiveOnly` • `PrimaryAndArchive` (the default value)	Specifies whether to move the user's primary mailbox, archive mailbox, or both.

Cross-forest enterprise moves and offboarding remote move migrations in a hybrid deployment

As previously stated, cross-forest moves are initiated either from the target forest or from the source forest. Offboarding remote move migrations are initiated from your Exchange Online organization. For more information, see:

- Prepare mailboxes for cross-forest move requests
- **Move mailboxes between on-premises and Exchange Online organizations in 2013 hybrid**

deployments

The following table describes the supported attributes for a CSV file for cross-forest enterprise moves and for offboarding remote move migrations in an Exchange hybrid deployment.

Attribute	Required or optional	Accepted values	Description
EmailAddress	Required	SMTP address for the user	For cross-forest enterprise moves, this specifies the mailbox or mail-enabled user in the source forest. For offboarding remote move migrations, it specifies the Exchange Online mailbox.
TargetDatabase	Required for offboarding remote move migrations and for cross-forest enterprise moves that are initiated from the source forest. Alternatively, this attribute can be specified when creating the migration batch in the EAC or using the Shell. This attribute is optional for cross-forest enterprise moves that are initiated from the	Database name	Specifies the mailbox database in the target forest that the user's primary mailbox will be moved to. You can specify a different database in the different rows of the CSV file, which lets you move mailboxes in the same migration batch to different databases.

	target forest.		
TargetArchiveDatabase	Optional	Database name	Specifies the mailbox database in the target forest that the user's archive mailbox will be moved to. You can specify a different database in the different rows of the CSV file, which lets you move archive mailboxes in the same migration batch to different databases.
BadItemLimit	Optional	`Unlimited` or a non-negative integer from `0` (the default) to a maximum value of `2147483647`	Specifies the number of bad items to skip if the migration service encounters a corrupted item in the mailbox. If you include this attribute in the CSV file, it will override the default value or the value you specify if you include the *BadItemLimit* parameter when creating the migration batch using the EAC or the Shell. **Tip:** We recommend that

			you use the default value of 0 and only increase the bad item limit for a particular user if the move or migration for that user fails.
LargeItemLimit	Optional	`Unlimited` or a non-negative integer from `0` (the default) to a maximum value.	Specifies the number of large items in the user's mailbox that will be skipped. When the number of large items exceeds this value, the migration for the mailbox fails. The default value is 0, which means that the migration fails if the mailbox contains any large items. When onboarding mailboxes to Exchange Online, items up to 35 MB are migrated.
MailboxType	Optional	Use one of the following values: • `PrimaryOnly` • `ArchiveOnly` • `PrimaryAndArchive` (the default value)	Specifies whether to move the user's primary mailbox, archive mailbox, or both.

Staged Exchange migrations

You have to use a CSV file to identify the group of users for a migration batch when you want to use a staged Exchange migration to migrate Exchange 2003 and Exchange 2007 on-premises mailboxes to Exchange Online. There isn't a limit for the number of mailboxes that you can migrate

to the cloud using a staged Exchange migration. However, the CSV file for a migration batch can contain a maximum of 1,000 rows. To migrate more than 1,000 mailboxes, you have to create additional CSV files, and then use each one to create a new migration batch. For more information about staged Exchange migrations, see **Migrate mailboxes to Exchange Online with a staged migration**.

The following table describes the supported attributes for a CSV file for a staged Exchange migration.

Attribute	Required or optional	Accepted values	Description
EmailAddress	Required	SMTP address for the user	Specifies the email address for the mail-enabled user (or a mailbox if you're retrying the migration) in Exchange Online that corresponds to the on-premises user mailbox that will be migrated. Mail-enabled users are created in Exchange Online as a result of directory synchronization or another provisioning process. The email address of the mail-enabled user must match the *WindowsEmailAddress* property for the corresponding on-premises mailbox.
Password	Optional	A password has to have a minimum length of eight	This password is set on the user account when the corresponding mail-

		characters, and satisfy any password restrictions that are applied to your Office 365 organization.	enabled user in Exchange Online is converted to a mailbox during the migration.
ForceChangePassword	Optional	`True` or `False`	Specifies whether a user must change the password the first time they sign in to their Exchange Online mailbox. **Note:** If you've implemented a single sign-on solution by deploying Active Directory Federation Services 2.0 (AD FS 2.0) in your on-premises organization, you must use `False` for the value of this attribute.

IMAP migrations

A CSV file for an IMAP migration batch can have maximum of 50,000 rows. But it's a good idea to migrate users in several smaller batches. For more information about IMAP migrations, see the following topics:

- **Migrate Email from an IMAP Server to Exchange Online Mailboxes**
- **CSV files for IMAP migration batches**

The following table describes the supported attributes for a CSV file for an IMAP migration.

Attribute	Required or optional	Accepted values	Description
EmailAddress	Required	SMTP address for the user.	Specifies the user ID for the user's Exchange Online mailbox
UserName	Required	String that identifies the user on the IMAP	Specifies the logon name for the user's

		messaging system, in a format supported by the IMAP server.	account in the IMAP messaging system (the source environment). In addition to the user name, you can use the credentials of an account that has been assigned the necessary permissions to access mailboxes on the IMAP server. For more information, see **CSV files for IMAP migration batches**.
Password	Required	Password string.	Specifies the password for the user account specified by the UserName attribute.

Attribute values in the CSV file override the values for the migration batch

Attribute values in the CSV file override the value of the corresponding parameter when that same parameter is used when creating a migration batch with the EAC or the Shell. If you want the migration batch value to be applied to a user, you would leave that cell blank in the CSV file. This lets you mix and match certain attribute values for selected users in one migration batch.

For example, let's say you create a batch in the Shell for a cross-forest enterprise move to move users' primary and archive mailboxes to the target forest with the following Shell command.

```
New-MigrationBatch -Name CrossForestBatch1 -SourceEndpoint
ForestEndpoint1 -TargetDeliveryDomain forest2.contoso.com -
TargetDatabases @(EXCH-MBX-02,EXCH-MBX-03) -
TargetArchiveDatabases @(EXCH-MBX-A02,EXCH-MBX-A03) -
CSVData ([System.IO.File]::ReadAllBytes("C:\Users
```

```
\Administrator\Desktop\CrossForestBatch1.csv")) -AutoStart
```

Note:

Because the default is to move primary and archive mailboxes, you don't have to explicitly specify it in the Shell command.

A portion of the CrossForestBatch1.csv file for this migration batch looks like this:

```
EmailAddress,TargetDatabase,TargetArchiveDatabase
user1@contoso.com,EXCH-MBX-01,EXCH-MBX-A01
user2@contoso.com,,
user3@contoso.com,EXCH-MBX-01,
...
```

Because the values in the CSV file override the values for the migration batch, the primary and archive mailboxes for user1 are moved to EXCH-MBX-01 and EXCH-MBX-A01, respectively, in the target forest. The primary and archive mailboxes for user2 are moved to either EXCH-MBX-02 or EXCH-MBX-03. The primary mailbox for user3 is moved to EXCH-MBX-01 and the archive mailbox is moved to either EXCH-MBX-A02 or EXCH-MBX-A03.

In another example, let's say you create a batch for an onboarding remote move migration in a hybrid deployment to move archive mailboxes to Exchange Online with the following command.

```
New-MigrationBatch -Name OnBoarding1 -SourceEndpoint
RemoteEndpoint1 -TargetDeliveryDomain cloud.contoso.com -
CSVData ([System.IO.File]::ReadAllBytes("C:\Users
\Administrator\Desktop\OnBoarding1.csv")) -MailboxType
ArchiveOnly -AutoStart
```

But you also want to move the primary mailboxes for selected users, so a portion of the OnBoarding1.csv file for this migration batch would look like this:

```
EmailAddress,MailboxType
user1@contoso.com,
user2@contoso.com,
user3@cloud.contoso.com,PrimaryAndArchive
user4@cloud.contoso.com,PrimaryAndArchive
...
```

Because the value for mailbox type in the CSV file overrides the values for the *MailboxType* parameter in the command to create the batch, only the archive mailbox for user1 and user2 is migrated to Exchange Online. But the primary and archive mailboxes for user3 and user4 are moved to Exchange Online.

Mailbox import and export requests

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-10*

Using the **MailboxImportRequest** or **MailboxExportRequest** cmdlet sets in the Exchange Management Shell, you can import data from or export data to .pst files. After you initiate a mailbox import or export request, the process is completed asynchronously by the Microsoft Exchange Mailbox Replication service (MRS). MRS resides on all Exchange 2010 Client Access servers and is the service responsible for moving mailboxes and importing and exporting .pst files.

Contents

Reasons to import or export mailbox data

You may want to import or export mailbox data for the following reasons:

- **Satisfy compliance requirements** You can export mailbox content to a .pst file for legal discovery purposes. After the export is complete, you can import the content to a mailbox used specifically for compliance purposes.
- **Create a point-in-time snapshot of a mailbox** By creating a snapshot of specific mailboxes, you avoid having to retain an entire backup set for a mailbox database.
- **Move a user's .pst file into his or her mailbox or personal archive** Microsoft Outlook users can save their email locally as .pst files. Using the New-MailboxImportRequest cmdlet, you can move data from a user's .pst file to his or her mailbox or personal archive. This is an easy method for transferring email from a user's local computer to Exchange servers.

Advantages to using import and export requests

Advantages to using import and export requests in Exchange 2013 include the following:

- A .pst provider is included in Exchange 2013 that can read and write .pst files.
- Import and export requests are asynchronous. The process is performed by MRS, which takes advantage of the queuing and throttling frameworks.
- The .pst files can be imported directly to a user's personal archive.

- Multiple .pst files can be imported or exported at the same time.
- The .pst files can reside on any shared network drive accessible by your Exchange servers.
- Exchange 2013 supports these .pst files: Unicode files created by Office Outlook 2007 and Outlook 2010

Considerations

Before you import or export mailbox data, consider the following:

- To import or export mailbox data, a network shared folder accessible by your Exchange servers must be set up. You must also grant read/write permissions to the Exchange Trusted Subsystem group so that the group can access the network share where you import and export mailbox data. If you don't grant this permission, you will receive an error message stating that Exchange is unable to establish a connection to the target mailbox.
- The maximum .pst file size supported by Outlook is 50 gigabytes (GB). Therefore, we recommend that you don't import a .pst file larger than 50 GB. You can create multiple .pst files for mailboxes larger than 50 GB by specifying specific folders to include or exclude or by using a content filter.
- Import and export requests are performed by MRS, which also processes move requests and mailbox restore requests. All requests are queued and throttled by MRS.
- Importing and exporting mailbox data may take several hours depending on file size, network bandwidth, and MRS throttling.
- Data can't be imported to a public folder or public folder database.

Importing mailbox data

Use the **MailboxImportRequest** cmdlet set to import data from a .pst file to a mailbox or personal archive. The following is a list of options you can specify when importing mailbox data from a .pst file:

Note:
The mailbox to which you import the data must exist. You can't import data to a user account that doesn't have a mailbox.

- You can import data to a different user account than the one from which it was exported. For example, you can export data from john@contoso.com and import it to legaldiscovery@contoso.com.
- You can import items to only the user's personal archive by specifying the *IsArchive* parameter.
- If associated folder messages exist in the .pst file, you can import them using the *AssociatedMessagesCopyOption* parameter. Associated messages contain hidden data with information about rules, views, and forms. If they exist in the .pst file, all messages from the safety net are imported.
- You can include or exclude specific folders using the *IncludeFolders* or *ExcludeFolders* parameter.
- You can exclude the Recoverable Items folder using the *ExcludeDumpster* parameter. By default, an import request includes the user's Recoverable Items folder if it's present in the .pst file.

Mailbox import request cmdlets

Use the following cmdlets for mailbox import requests.

Cmdlet	Description
New-MailboxImportRequest	Starts the process of importing a .pst file to a mailbox or personal archive. You can create more than one import request per mailbox. Each request must have a unique name.
Set-MailboxImportRequest	Changes import request options after the request is created or recover from a failed request.
Suspend-MailboxImportRequest	Suspends an import request any time after the request is created but before the request reaches the status of Completed.
Resume-MailboxImportRequest	Resumes an import request that's suspended or failed.
Remove-MailboxImportRequest	Removes fully or partially completed import requests. Completed import requests aren't automatically cleared. You must use this cmdlet to remove them.
Get-MailboxImportRequest	View general information about an import request.
Get-MailboxImportRequestStatistics	View detailed information about an import request.

Exporting mailbox data

Use the **MailboxExportRequest** cmdlet set to export mailbox data to a .pst file. You can export one mailbox or several mailboxes, but only one request is written to each .pst file at a time. The following is a list of options you can specify when exporting mailbox data to a .pst file:

- You can export personal archive data using the *IsArchive* parameter.

- You can filter the messages that are exported using the *ContentFilter* parameter. You can filter by message content, attachment, senders, recipients, Inbox category, importance, message type, message size, and when the message was sent, received, or expired.
- You can specify folders to include or exclude using the *IncludeFolders* or *ExcludeFolders* parameter. If exporting data from an Exchange 2013 mailbox, you can also exclude the Recoverable Items folder using the *ExcludeDumpster* parameter.
- You can export associated messages using the *AssociatedMessagesCopyOption* parameter. Associated messages contain hidden data with information about rules, views, and forms. By default, associated items aren't copied to the .pst file.

Mailbox export request cmdlets

Use the following cmdlets for mailbox export requests.

Cmdlet	Description
New-MailboxExportRequest	Starts the process of exporting data from a primary mailbox or personal archive to a .pst file. You can create more than one export request per mailbox. Each request must have a unique name.
Set-MailboxExportRequest	Changes export request options after the request is created or recover from a failed request.
Suspend-MailboxExportRequest	Suspends an export request any time after the request is created but before the request reaches the status of Completed.
Resume-MailboxExportRequest	Resumes an export request that's suspended or failed.
Remove-MailboxExportRequest	Removes fully or partially completed export requests. Completed export requests aren't automatically cleared. You must use this cmdlet to remove them.
Get-MailboxExportRequest	View general information about an export request.

Get-MailboxExportRequestStatistics	View detailed information about an export request.

Recoverable Items folder

Exchange Server 2013 > Mailbox and Client Access servers > Mailbox server >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-12*

To protect from accidental or malicious deletion and to facilitate discovery efforts commonly undertaken before or during litigation or investigations, Microsoft Exchange Server 2013 uses the Recoverable Items folder. The Recoverable Items folder replaces the feature known as the dumpster in Exchange Server 2007. The Recoverable Items folder is used by the following Exchange features:

- Deleted item retention
- Single item recovery
- In-Place Hold
- Litigation hold
- Mailbox audit logging
- Calendar logging

Contents

Terminology

Knowledge of the following terms will help you understand the content in this topic.

Delete

Describes when an item is deleted from any folder and placed in the Deleted Items default folder.

Soft delete

Describes when an item is deleted from the Deleted Items default folder and placed in the Recoverable Items folder. Additionally describes when a Microsoft Outlook user deletes an item by pressing Shift+Delete, which bypasses the Deleted Items folder and places the item directly in the Recoverable Items folder.

Hard delete

Describes when an item is marked to be purged from the mailbox database. This is also known as

a *store hard delete*.

Recoverable Items folder

To better meet customers' legal compliance requirements, the dumpster is reinvented as the Recoverable Items folder in Exchange 2010. The Recoverable Items folder resides in the non-IPM subtree of each mailbox. The non-IPM subtree is a storage area within the mailbox that contains operational data about the mailbox. This subtree isn't visible to users using Outlook, Microsoft Office Outlook Web App, or other e-mail clients.

This architectural change provides the following key benefits:

- When a mailbox is moved to another mailbox database, the Recoverable Items folder moves with it.
- The Recoverable Items folder is indexed by Exchange Search and can be discovered using In-Place eDiscovery.
- The Recoverable Items folder has its own storage quota.
- Exchange can prevent data from being purged from the Recoverable Items folder.
- Exchange can track edits of certain content.

The Recoverable Items folder contains the following subfolders:

- **Deletions** This subfolder contains all items deleted from the Deleted Items folder. (In Outlook, you can soft delete an item by pressing Shift+Delete.) This subfolder is exposed to users through the Recover Deleted Items feature in Outlook and Outlook Web App.
- **Versions** If In-Place Hold, litigation hold, or single item recovery is enabled, this subfolder contains the original and modified copies of the deleted items. This folder isn't visible to end users.
- **Purges** If either litigation hold or single item recovery is enabled, this subfolder contains all items that are hard deleted. This folder isn't visible to end users.
- **DiscoveryHolds** If In-Place Hold is enabled, this subfolder contains all items that meet the hold query parameters and are hard deleted.
- **Audits** If mailbox audit logging is enabled for a mailbox, this subfolder contains the audit log entries. To learn more about mailbox audit logging, see Mailbox audit logging.
- **Calendar Logging** This subfolder contains calendar changes that occur within a mailbox. This folder isn't available to users.

Deleted item retention

An item is considered to be soft deleted in the following cases:

- A user deletes an item or empties all items from the Deleted Items folder.
- A user presses Shift+Delete to delete an item from any other mailbox folder.

Soft-deleted items are moved to the Deletions subfolder of the Recoverable Items folder. This provides an additional layer of protection so users can recover deleted items without requiring Help desk intervention. Users can use the Recover Deleted Items feature in Outlook or Outlook Web

App to recover a deleted item. Users can also use this feature to permanently delete an item.

Items remain in the Deletions subfolder until the deleted item retention period is reached. The default deleted item retention period for a mailbox database is 14 days. You can modify this period for a mailbox database or for a specific mailbox. In addition to a deleted item retention period, the Recoverable Items folder is also subject to quotas. To learn more, see Recoverable Items Mailbox Quotas later in this topic.

After the deleted item retention period elapses, the item is moved to the Purges folder and is no longer visible to the user. When the Managed Folder Assistant processes the mailbox, items in the Purges subfolder are purged from the mailbox database.

Single item recovery

If an item is removed from the Deletions subfolder, either using the Recover Deleted Items feature or by an automated process such as the Managed Folder Assistant, the item can't be recovered by the user. In previous versions of Exchange, recovering these items required the administrator to restore the mailbox database or a mailbox from backup copies. This process generally delayed recovery by minutes or hours, depending on the backup mechanism used.

In Exchange 2013, you can use *single item recovery* to recover items without having to restore the mailbox databases using backup media. This results in considerably shorter recovery periods. When the Managed Folder Assistant processes the Recoverable Items folder for a mailbox that has single item recovery enabled, any item in the Purges subfolder isn't purged if the deleted item retention period hasn't elapsed for that item.

The following table lists the contents of and actions that can be performed in the Recoverable Items folder if single item recovery is enabled.

Recoverable Items folder and single item recovery

State of single item recovery	Recoverable Items folder contains soft-deleted items	Recoverable Items folder contains hard-deleted items	Users can purge items from the Recoverable Items folder	Managed Folder Assistant automatically purges items from the Recoverable Items folder
Enabled	Yes	Yes	No	Yes. By default, all items are purged after 14 days, with the exception of calendar items, which are purged

				after 120 days.
Disabled	Yes	No	Yes	Yes. By default, all items are purged after 14 days, with the exception of calendar items, which are purged after 120 days. If the Recoverable Items warning quota is reached before the deleted item retention period elapses, messages are deleted in first in, first out (FIFO) order.

Single item recovery isn't enabled by default for new mailboxes or mailboxes moved from a previous version of Exchange. You must use the Exchange Management Shell to enable single item recovery for a mailbox, and then configure or modify the deleted item retention period. For details about how to perform a single item recovery, see Perform single item recovery.

In-Place Hold and litigation hold

In Exchange 2013, discovery managers can use In-Place eDiscovery with delegated Discovery Management permissions to perform eDiscovery searches of mailbox content. Exchange 2013 also introduces In-Place Hold, which you can use to preserve mailbox items that match query parameters and protect the items from deletion by users or automated processes. You can also use litigation hold, the preservation feature introduced in Exchange 2010, to preserve all items in user mailboxes and protect the items from deletion by users or automated processes.

Placing a mailbox on In-Place Hold or litigation hold stops the Managed Folder Assistant from automatically purging messages from the DiscoveryHolds and Purges subfolders. Additionally, copy-on-write page protection is also enabled for the mailbox. Copy-on-write page protection creates a copy of the original item before any modifications are written to the Exchange store.

After the mailbox is removed from litigation hold, the Managed Folder Assistant resumes automated purging.

The following table lists the contents of and actions that can be performed in the Recoverable Items folder if litigation hold is enabled.

Recoverable Items folder and holds

State of litigation hold	Recoverable Items folder contains soft-deleted items	Recoverable Items folder contains modified and hard-deleted items	Users can purge items from the Recoverable Items folder	Managed Folder Assistant automatically purges items from the Recoverable Items folder
Enabled	Yes	Yes	No	No
Disabled	Yes	No	Yes	Yes

To learn more about In-Place eDiscovery, In-Place Hold, and litigation hold, see the following topics:

- In-Place eDiscovery
- In-Place Hold

Copy-on-write page protection and modified items

If a user who is placed on In-Place Hold or litigation hold modifies specific properties of a mailbox item, a copy of the original mailbox item is created before the changed item is written. The original copy is saved in the Versions subfolder. This process is known as copy-on-write page protection. Copy-on-write page protection applies to items residing in any mailbox folder. The Versions subfolder isn't visible to users.

The following table lists the message properties that trigger copy-on-write page protection.

Properties that trigger copy-on-write page protection

Item type	Properties that trigger copy-on-write page protection
Messages (IPM.Note*) Posts (IPM.Post*)	• Subject • Body • Attachments • Senders and recipients • Sent and received dates
Items other than messages and posts	Any change to a visible property, except the

	following: • Item location (when an item is moved between folders) • Item status change (read or unread) • Changes to a retention tag applied to an item
Items in the Drafts default folder	None. Items in the Drafts folder are exempt from copy-on-write page protection.

Important:
Copy-on-write page protection doesn't save a version of the meeting when a meeting organizer receives responses from attendees and the meeting's tracking information is updated. Also, changes to RSS feeds aren't captured by copy-on-write page protection.

When a mailbox is no longer on In-Place Hold or litigation hold, copies of modified items stored in the Versions folder are removed.

Recoverable Items mailbox quotas

When an item is moved to the Recoverable Items folder, its size is deducted from the mailbox quota and added to the size of the Recoverable Items folder. In Exchange 2010, mailbox databases have a configurable Recoverable Items warning quota (*soft limit*) of 20 gigabytes (GB) and a Recoverable Items quota (*hard limit*) of 30 GB. By default, these limits are inherited by all mailboxes in the database. However, you can configure individual mailboxes with different quotas. To learn more, see Configure Deleted Item retention and Recoverable Items quotas.

When the Recoverable Items folder for a mailbox reaches the Recoverable Items quota, no more items can be stored in the folder. This impacts mailbox functionality in the following ways:

- Mailbox users can't delete items.
- The Managed Folder Assistant can't delete items based on retention tag or managed folder settings.
- For mailboxes that have single item recovery, In-Place Hold or litigation hold enabled, the copy-on-write page protection process can't maintain versions of items edited by the user.
- For mailboxes that have mailbox audit logging enabled, no mailbox audit log entries can be saved in the Audits subfolder.

For mailboxes that aren't placed on In-Place Hold or litigation hold, the Managed Folder Assistant automatically purges items from the Recoverable Items folder when the deleted item retention period elapses. If the folder reaches the Recoverable Items warning quota, the assistant automatically purges items in FIFO order.

When the Recoverable Items folder reaches the soft and hard limit defaults, you are notified by means of an event log and a Microsoft System Center Operations Manager alert. This alert fires

when the Recoverable Items folder first reaches the soft and hard limit defaults, and then once daily afterward.

The following table lists the events logged when the Recoverable Items folder reaches the soft and hard limit defaults.

Recoverable Items quota warnings and errors

Event ID	Type	Source	Message
10024	Warning	MSExchangeIS Mailbox Store	The mailbox for *<mailbox user>* (*GUID*) has exceeded the Recoverable Items Warning Quota. Please remove items from Recoverable Items or increase the Recoverable Items Warning Quota and Recoverable Items Quota. If the Recoverable Items Quota is exceeded, the user will be unable to delete items from the mailbox.
10023	Error	MSExchangeIS Mailbox Store	The mailbox for *<mailbox user>* (*GUID*) has exceeded the maximum Recoverable Items Quota. Items cannot be deleted from this mailbox. The mailbox owner should be notified about the condition of the

			mailbox as soon as possible. Please remove items from Recoverable Items or increase the Recoverable Items Quota to restore functionality.
10023	Warning	MSExchangeMailboxAssistants	The mailbox:<*mailbox user*> Recoverable Items size has exceeded the warning quota limit. Items were deleted from Recoverable Items folders to prevent mailbox outage. Recoverable Items Warning Quota: 20 GB (21,474,836,480 bytes) Original Recoverable Items size: 21475005311 Current Recoverable Items size: 21474823820 Folder stats: - Folders processed: RecoverableItemsRoot, RecoverableItemsVersions, RecoverableItemsPurges,

			RecoverableItemsDeletions - Original folder sizes: 21391661934, 55190914, 1987247, 26157788 (item counts: 276828, 400, 84, 646) - Current folder sizes: 21391480443, 55190914, 1987247, 26157788 (item counts: 276817, 400, 84, 646)

If the mailbox is placed on In-Place Hold or litigation hold, copy-on-write page protection can't maintain versions of modified items. To maintain versions of modified items, you must reduce the size of the Recoverable Items folder. You can use the Search-Mailbox cmdlet to copy messages from the Recoverable Items folder of a mailbox to a discovery mailbox, and then delete the items from the mailbox. Alternatively, you can also raise the Recoverable Items quota for the mailbox. For details, see Clean up the Recoverable Items folder.

More Info

- In Exchange Online and Exchange 2013, copy-on-write is only enabled when a mailbox is on In-Place Hold or Litigation Hold.

Clean up the Recoverable Items folder

Mailbox and Client Access servers > Mailbox server > Recoverable Items folder >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-11*

The Recoverable Items folder (known in earlier versions of Exchange as the dumpster) exists to protect from accidental or malicious deletions and to facilitate discovery efforts commonly undertaken before or during litigation or investigations. To learn more about the Recoverable Items folder, see Recoverable Items folder.

How you clean up a mailbox's Recoverable Items folder depends on whether the mailbox is placed on In-Place Hold or litigation hold, or had single item recovery enabled:

- If a mailbox isn't placed on In-Place Hold or litigation hold or doesn't have single item recovery enabled, you can simply delete items from the Recoverable Items folder. After being deleted, you can't use single item recovery to recover the items.
- If the mailbox is placed on In-Place Hold or litigation hold or has single item recovery enabled, it's important to preserve the mailbox data until the hold is removed or single item recovery is disabled. In this case, you need to perform more detailed steps to clean up the Recoverable Items folder.

To learn more about In-Place Hold and litigation hold, see In-Place Hold. To learn more about single item recovery, see "Single Item Recovery" in Recoverable Items folder.

To learn more about the Recoverable Items folder, see Recoverable Items folder.

What do you need to know before you begin?

- Estimated time to complete this procedure: 30 minutes. This may vary depending on the size of the Recoverable Items folder
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Delete mailbox content" entry in the Messaging policy and compliance permissions topic.
- Because incorrectly cleaning up the Recoverable Items folder can result in data loss, it’s important that you're familiar with the Recoverable Items folder and the impact of removing its contents. Before performing this procedure, we recommend that you review the information in Recoverable Items folder.
- You can't use the Exchange Administration Center (EAC) to perform these procedures. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to delete items from the Recoverable Items folder for mailboxes that aren't placed on In-Place Hold or litigation hold or don't have single item recovery enabled

This example permanently deletes items from Gurinder Singh's Recoverable Items folder and also copies the items to the GurinderSingh-RecoverableItems folder in the Discovery Search Mailbox (a discovery mailbox created by Exchange Setup).

```
Search-Mailbox -Identity "Gurinder Singh" -
SearchDumpsterOnly -TargetMailbox "Discovery Search
Mailbox" -TargetFolder "GurinderSingh-RecoverableItems" -
DeleteContent
```

Note:

To delete items from the mailbox without copying them to another mailbox, use the preceding command without the *TargetMailbox* and *TargetFolder* parameters.

For detailed syntax and parameter information, see Search-Mailbox.

Use the Shell to clean up the Recoverable Items folder for mailboxes that are placed on In-Place Hold or litigation hold or have single item recovery enabled

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Delete mailbox content" entry in the Messaging policy and compliance permissions topic.

If a mailbox reaches its Recoverable Items quota, we recommend that you raise the quota and not delete items from the folder. You can also monitor events in the Application log related to the Recoverable Items warning quota and take necessary actions (such as raising the quota or investigating dumpster growth for mailboxes that reach the warning quota).

If storage constraints or similar issues prevent you from raising the Recoverable Items quota, and you need to delete messages from the Recoverable Items folder of a mailbox on In-Place Hold or litigation hold or has single item recovery enabled, we recommend that you first copy data from the user's Recoverable Items folder to another mailbox. If you're deleting items due to storage constraints on one volume, you can copy items to a mailbox located on a volume that has adequate storage.

This procedure copies items from Gurinder Singh's Recoverable Items folder to the GurinderSingh-RecoverableItems folder in the Discovery Search Mailbox. Before you copy and delete items from the Recoverable Items folder, you must first perform several steps to make sure items aren't deleted from the Recoverable Items folder. After you copy items to a discovery or backup mailbox and clean up the folder, you can revert to the mailbox's previous settings.

1. Retrieve the following quota settings. Be sure to note the values so you can revert to these settings after cleaning up the Recoverable Items folder:
   - *RecoverableItemsQuota*
   - *RecoverableItemsWarningQuota*
   - *ProhibitSendQuota*
   - *ProhibitSendReceiveQuota*

- *UseDatabaseQuotaDefaults*
- *RetainDeletedItemsFor*
- *UseDatabaseRetentionDefaults*

Note:

If the *UseDatabaseQuotaDefaults* parameter is set to `$true`, the previous quota settings aren't applied. The corresponding quota settings configured on the mailbox database are applied, even if individual mailbox settings are populated.

```
Get-Mailbox "Gurinder Singh" | Format-List
RecoverableItemsQuota, RecoverableItemsWarningQuota,
ProhibitSendQuota, ProhibitSendReceiveQuota,
UseDatabaseQuotaDefaults, RetainDeletedItemsFor,
UseDatabaseRetentionDefaults
```

2. Retrieve the mailbox access settings for the mailbox. Be sure to note these settings for later.

```
Get-CASMailbox "Gurinder Singh" | Format-List EwsEnabled,
ActiveSyncEnabled, MAPIEnabled, OWAEnabled, ImapEnabled,
PopEnabled
```

3. Retrieve the current size of the Recoverable Items folder. Note the size so you can raise the quotas in Step 6.

```
Get-MailboxFolderStatistics "Gurinder Singh" -FolderScope
RecoverableItems | Format-List Name,FolderAndSubfolderSize
```

4. Retrieve the current Managed Folder Assistant work cycle configuration. Be sure to note the setting for later.

```
Get-MailboxServer "My Mailbox Server" | Format-List
Name,ManagedFolderWorkCycle
```

5. Disable client access to the mailbox to make sure no changes can be made to mailbox data for the duration of this procedure.

```
Set-CASMailbox "Gurinder Singh" -EwsEnabled $false -
ActiveSyncEnabled $false -MAPIEnabled $false -OWAEnabled
$false -ImapEnabled $false -PopEnabled $false
```

6. To make sure no items are deleted from the Recoverable Items folder, increase the Recoverable Items quota, increase the Recoverable Items warning quota, and set the deleted item retention period to a value higher than the current size of the user's Recoverable Items folder. This is particularly important for preserving messages for mailboxes placed on In-Place Hold or litigation hold. We recommend raising these settings to twice their current size.

```
Set-Mailbox "Gurinder Singh" -RecoverableItemsQuota 50Gb -
RecoverableItemsWarningQuota 50Gb -RetainDeletedItemsFor
3650 -ProhibitSendQuota 50Gb -ProhibitSendRecieveQuota 50Gb
-UseDatabaseQuotaDefaults $false -
UseDatabaseRetentionDefaults $false
```

7. Disable the Managed Folder Assistant on the Mailbox server.

```
Set-MailboxServer MyMailboxServer -ManagedFolderWorkCycle
$null
```

Important:

If the mailbox resides on a mailbox database in a database availability group (DAG), you must disable the Managed Folder Assistant on each DAG member that hosts a copy of the database. If the database fails over to another server, this prevents the Managed Folder Assistant on that server from deleting mailbox data.

8. Disable single item recovery and remove the mailbox from litigation hold.

```
Set-Mailbox "Gurinder Singh" -SingleItemRecoveryEnabled
$false -LitigationHoldEnabled $false
```

Important:

After you run this command, it may take up to one hour to disable single item recovery or litigation hold. We recommend that you perform the next step only after this period has elapsed.

9. Copy items from the Recoverable Items folder to a folder in the Discovery Search Mailbox and delete the contents from the source mailbox.

```
Search-Mailbox -Identity "Gurinder Singh" -
SearchDumpsterOnly -TargetMailbox "Discovery Search
Mailbox" -TargetFolder "GurinderSingh-RecoverableItems" -
DeleteContent
```

If you need to delete only messages that match specified conditions, use the *SearchQuery* parameter to specify the conditions. This example deletes messages that have the string "Your bank statement" in the **Subject** field.

```
Search-Mailbox -Identity "Gurinder Singh" -SearchQuery
"Subject:'Your bank statement'" -SearchDumpsterOnly -
TargetMailbox "Discovery Search Mailbox" -TargetFolder
"GurinderSingh-RecoverableItems" -DeleteContent
```

Note:

It isn't required to copy items to the Discovery Search Mailbox. You can copy messages to any

mailbox. However, to prevent access to potentially sensitive mailbox data, we recommend copying messages to a mailbox that has access restricted to authorized records managers. By default, access to the default Discovery Search Mailbox is restricted to members of the Discovery Management role group. For details, see In-Place eDiscovery.

10.If the mailbox was placed on litigation hold or had single item recovery enabled earlier, enable these features again.

```
Set-Mailbox "Gurinder Singh" -SingleItemRecoveryEnabled $true -LitigationHoldEnabled $true
```

Important:

After you run this command, it may take up to one hour to enable single item recovery or litigation hold. We recommend that you enable the Managed Folder Assistant and allow client access (Steps 11 and 12) only after this period has elapsed.

11.Revert the following quotas to the values noted in Step 1:

- *RecoverableItemsQuota*
- *RecoverableItemsWarningQuota*
- *ProhibitSendQuota*
- *ProhibitSendReceiveQuota*
- *UseDatabaseQuotaDefaults*
- *RetainDeletedItemsFor*
- *UseDatabaseRetentionDefaults*

In this example, the mailbox is removed from retention hold, the deleted item retention period is reset to the default value of 14 days, and the Recoverable Items quota is configured to use the same value as the mailbox database. If the values you noted in Step 1 are different, you must use the preceding parameters to specify each value and set the *UseDatabaseQuotaDefaults* parameter to `$false`. If the *RetainDeletedItemsFor and UseDatabaseRetentionDefaults* parameters were previously set to a different value, you must also revert them to the values noted in Step 1.

```
Set-Mailbox "Gurinder Singh" -RetentionHoldEnabled $false -RetainDeletedItemsFor 14 -RecoverableItemsQuota unlimited -UseDatabaseQuotaDefaults $true
```

12.Enable the Managed Folder Assistant by setting the work cycle back to the value you noted in Step 4. This example sets the work cycle to one day.

```
Set-MailboxServer MyMailboxServer -ManagedFolderWorkCycle 1
```

13.Enable client access.

```
Set-CASMailbox -ActiveSyncEnabled $true -EwsEnabled $true -MAPIEnabled $true -OWAEnabled $true -ImapEnabled $true -PopEnabled $true
```

For detailed syntax and parameter information, see the following topics:

- Get-Mailbox
- Get-CASMailbox
- Get-MailboxFolderStatistics
- Get-MailboxServer
- Set-CASMailbox
- Set-Mailbox
- Set-MailboxServer
- Search-Mailbox

How do you know this worked?

To verify that you have successfully cleaned up the Recoverable Items folder of a mailbox, use Get-MailboxFolderStatistics cmdlet the check the size of the Recoverable Items folder.

This example retrieves the size of the Recoverable Items folder and its subfolders and an item count in the folder and each subfolder.

```
Get-MailboxFolderStatistics -Identity "Gurinder Singh" -
FolderScope RecoverableItems | Format-Table
Name,FolderAndSubfolderSize,ItemsInFolderAndSubfolders -
Auto
```

Configure Deleted Item retention and Recoverable Items quotas

Mailbox and Client Access servers > Mailbox server > Recoverable Items folder >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2012-12-11*

When a user deletes items from the Deleted Items default folder by using the Delete, Shift+Delete, or **Empty Deleted Items Folder** actions, the items are moved to the **Recoverable Items\Deletions** folder. The duration that deleted items remain in this folder is based on the deleted item retention settings configured for the mailbox database or the mailbox. By default, a mailbox database is configured to retain deleted items for 14 days, and the recoverable items warning quota and recoverable items quota are set to 20 gigabytes (GB) and 30 GB respectively.

Note:
Before the retention time for deleted items elapses, Microsoft Outlook and Microsoft Office

Outlook Web App users can recover deleted items by using the Recover Deleted Items feature. To learn more about these features, see the "Recover deleted items" topic for Outlook or Outlook Web App.

You can use the Shell to configure deleted item retention settings and recoverable items quotas for a mailbox or mailbox database. Deleted item retention settings are ignored when a mailbox is placed on In-Place Hold or litigation hold.

To learn more about deleted item retention, the Recoverable Items folder, In-Place Hold, and litigation hold, see Recoverable Items folder.

What do you need to know before you begin?

- Estimated time to completion: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Configure deleted item retention for a mailbox

Use the EAC to configure deleted item retention for a mailbox

1. Navigate to **Recipients** > **Mailboxes**.
2. In the list view, select a mailbox, and then click **Edit** .
3. On the mailbox property page, click **Mailbox usage**, click **More options**, and then select one of the following
   - **Use the default retention settings from the mailbox database** Use this setting to use the deleted item retention setting that's configured for the mailbox database.
   - **Customize the settings for this mailbox** Use this setting to configure deleted item retention settings for the mailbox.

Keep deleted items for (days) This box displays the length of time that deleted items are retained before they're permanently deleted and can't be recovered by the user. When the mailbox is created, this value is based on the deleted item retention settings configured for the mailbox database. By default, a mailbox database is configured to retain deleted items for 14 days. The value range for this property is from 0 through 24,855 days.

   - **Don't permanently delete items until the database is backed up** Select this check box to prevent mailboxes and email messages from being deleted until after the mailbox database

on which the mailbox is located has been backed up.

Use the Shell to configure deleted item retention for a mailbox

This example configures April Stewart's mailbox to retain deleted items for 30 days.

```
Set-Mailbox -Identity - "April Stewart" -
RetainDeletedItemsFor 30
```

For detailed syntax and parameter information, see Set-Mailbox.

Use the Shell to configure recoverable items quotas for a mailbox

Note:
You can't use the EAC to configure recoverable items quotas for a mailbox.

This example configures a recoverable items warning quota of 12 GB and a recoverable items quota of 15 GB for April Stewart's mailbox.

```
Set-Mailbox -Identity "April Stewar"t -
RecoverableItemsWarningQuota 12GB -RecoverableItemsQuota
15GB -UseDatabaseQuotaDefaults $false
```

Note:
To configure a mailbox to use different recoverable items quotas than the mailbox database in which it resides, you must set the *UseDatabaseQuotaDefaults* parameter to `$false`.

For detailed syntax and parameter information, see Set-Mailbox.

Use the Shell to configure deleted item retention for a mailbox database

Note:
You can't use the EAC to configure deleted item retention for a mailbox database.

This example configures a deleted item retention period of 10 days for the mailbox database MDB2.

```
Set-MailboxDatabase -Identity MDB2 -DeletedItemRetention 10
```

For detailed syntax and parameter information, see Set-MailboxDatabase.

Use the Shell to configure recoverable items quotas for a mailbox database

Note:
You can't use the EAC to configure recoverable items quotas for a mailbox database

This example configures a recoverable items warning quota of 15 GB and a recoverable items quota of 20 GB on mailbox database MDB2.

```
Set-MailboxDatabase -Identity MDB2 -
RecoverableItemsWarningQuota 15GB -RecoverableItemsQuota
20GB
```

For detailed syntax and parameter information, see Set-MailboxDatabase.

Perform single item recovery

Mailbox and Client Access servers > Mailbox server > Recoverable Items folder >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2012-12-11*

Single item recovery provides an additional layer of protection so that you can recover items that were accidentally deleted by a user or by automated processes such as the Managed Folder Assistant. Single item recovery simplifies recovery and reduces recovery time because you can recover items without recovering an entire mailbox or mailbox database from backup media. To learn more, see "Single item recovery" in Recoverable Items folder.

Note:
In addition to using this procedure to search for and recover deleted items (which are moved to the Recoverable Items\Purges folder if either single item recovery or litigation hold is enabled), you can also use this procedure to search for items residing in other folders in the mailbox and to delete items from the source mailbox (also known as *search and destroy*).

This topic shows you how to use the **Search-Mailbox** cmdlet in the Shell to search for and recover missing items. If the mailbox has a standard CAL, this is the only method you can use. If you use this cmdlet, you can search only one mailbox at a time. If you want to search multiple mailboxes simultaneously, you can use the In-Place eDiscovery feature in the Exchange Administration Center (EAC) or the New-MailboxSearch cmdlet in the Shell.

What you need to know before you begin?

- Estimated time to complete: 15-30 minutes.
- Procedures in this topic require specific permissions. See each procedure for its permissions information.
- Single item recovery must be enabled for a mailbox before the item you want to recover is deleted.
- To search for and recover items, you must have the following information:
  - **Source mailbox** This is the mailbox being searched.
  - **Target mailbox** This is the discovery mailbox in which messages will be recovered. Exchange Setup creates a default discovery mailbox. If required, you can create additional discovery mailboxes. For details, see Create a discovery mailbox.

Note:

When using the **Search-Mailbox** cmdlet, you can also specify a target mailbox that isn't a discovery mailbox. However, you can't specify the same mailbox as the source and target mailbox.

  - **Search criteria** Criteria include sender or recipient, or keywords (words or phrases) in the message.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Step 1: Search for and recover missing items

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.

Note:

You can use In-Place eDiscovery in the EAC to search for missing items. However, when using the EAC, you can’t restrict the search to the Recoverable Items folder. Messages matching your search parameters will be returned even if they’re not deleted. After they’re recovered to the specified discovery mailbox, you may need to review the search results and remove unnecessary messages before recovering the remaining messages to the user’s mailbox or exporting them to a .pst file.

The first step in the recovery process is to search for messages in the source mailbox. Use one of the following methods to search a user mailbox and copy messages to a discovery mailbox.

Use the Shell

This example searches for messages in April Stewart's mailbox that meet the following criteria:

- Sender: Ken Kwok
- Keyword: Seattle

```
Search-Mailbox "April Stewart" -SearchQuery "from:'Ken Kwok' AND seattle" -TargetMailbox "Discovery Search Mailbox" -TargetFolder "April Stewart Recovery" -LogLevel Full
```

Note:

When using the **Search-Mailbox** cmdlet, you can scope the search by using the *SearchQuery* parameter to specify a query formatted using Keyword Query Language (KQL). You can also use the *SearchDumpsterOnly* switch to search only items in the Recoverable Items folder.

For detailed syntax and parameter information, see Search-Mailbox.

Use the EAC to perform an In-Place eDiscovery search

For details about how to use the EAC to perform an In-Place eDiscovery search, see Create an In-Place eDiscovery search.

How do you know this worked?

To verify that you have successfully searched the messages you want to recover, log on to the discovery mailbox you selected as the target mailbox and review the search results.

Step 2: Restore recovered items

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.

Note:

You can't use the EAC to restore recovered items.

After messages have been recovered to a discovery mailbox, you can restore them to the user's mailbox by using the **Search-Mailbox** cmdlet. You can also use the **New-MailboxExportRequest** and **New-MailboxImportRequest** cmdlets to export the messages to or import the messages from a .pst file.

Use the Shell to restore messages

This example restores messages to April Stewart's mailbox and deletes them from the Discovery Search Mailbox.

```
Search-Mailbox "Discovery Search Mailbox" -SearchQuery
"from:'Ken Kwok' AND seattle" -TargetMailbox "April
Stewart" -TargetFolder "Recovered Messages" -LogLevel Full
-DeleteContent
```

For detailed syntax and parameter information, see Search-Mailbox.

How do you know this worked?

To verify that you have successfully recovered messages to the user's mailbox, have the user review messages in the target folder you specified in the above command.

Use the Shell to export and import messages from a .pst file

You can export contents from a mailbox to a .pst file and import the contents of a .pst file to a mailbox. To learn more about mailbox import and export, see Mailbox import and export requests.

This example uses the following settings to export messages from the folder April Stewart Recovery in the Discovery Search Mailbox to a .pst file:

- **Mailbox** Discovery Search Mailbox
- **Source folder** April Stewart Recovery
- **ContentFilter** April travel plans
- **PST file path** \\MYSERVER\HelpDeskPst\AprilStewartRecovery.pst

```
New-MailboxExportRequest -Mailbox "Discovery Search
Mailbox" -SourceRootFolder "April Stewart Recovery" -
ContentFilter {Subject -eq "April travel plans"} -FilePath
\\MYSERVER\HelpDeskPst\AprilStewartRecovery.pst
```

For detailed syntax and parameter information, see New-MailboxExportRequest.

This example uses the following settings to import messages from a .pst file to the folder Recovered By Helpdesk in April Stewart's mailbox:

- **Mailbox** April Stewart
- **Target folder** Recovered By Helpdesk
- **PST file path** \\MYSERVER\HelpDeskPst\AprilStewartRecovery.pst

```
New-MailboxImportRequest -Mailbox "April Stewart" -
TargetRootFolder "Recovered By Helpdesk" -FilePath \
\MYSERVER\HelpDeskPst\AprilStewartRecovery.pst
```

For detailed syntax and parameter information, see New-MailboxImportRequest.

How do you know this worked?

To verify that you have successfully exported messages to a .pst file, use Outlook to open the .pst file and inspect its contents. To verify that you have successfully imported messages from the .pst file, have the user inspect the contents of the target folder you specified in the above command.

Get Recoverable Items folder statistics

Mailbox and Client Access servers > Mailbox server > Recoverable Items folder >

Applies to: *Exchange Server 2010 Service Pack 2 (SP2)*

Topic Last Modified: *2013-02-22*

The Recoverable Items folder contains items deleted by Microsoft Outlook and Microsoft Office Outlook Web App users or by the Mailbox Assistant. The duration that deleted items remain in this folder is based on the deleted item retention settings configured for the mailbox database or the mailbox. By default, a mailbox database is configured to retain deleted items for 14 days, and the Recoverable Items warning quota and Recoverable Items quota are set to 20 gigabytes (GB) and 30 GB respectively. However, if In-Place Hold or litigation hold is enabled for the mailbox, the Recoverable Items folder can accumulate deleted items beyond the specified retention period and can also maintain different versions of modified mailbox items.

When the Recoverable Items folder reaches the Recoverable Items warning quota, a warning event is logged in the Application event log. If the mailbox isn't on litigation hold, items are then removed on a first in, first out (FIFO) basis. However, if the mailbox is on litigation hold, the mailbox is never emptied and upon reaching the Recoverable Items quota, mailbox functionality is impacted.

Therefore, it's important to monitor the event log for alerts generated when mailboxes reach the Recoverable Items quotas. You can also use this procedure to report statistics for the Recoverable Items folder, particularly for mailboxes placed on litigation hold.

To learn more, see the following topics:

- Recoverable Items folder
- In-Place Hold

What do you need to know before you begin?

- Estimated time to complete: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox folders" entry in the Recipients Permissions topic.
- You can't use the Exchange admin center (EAC) to get Recoverable Items folder statistics for a

mailbox.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Get Recoverable Items folder statistics for a mailbox

This example gets folder statistics for Soumya Singhi's Recoverable Items folder and displays the output in a list format.

```
Get-MailboxFolderStatistics -Identity "Soumya Singhi" -
FolderScope RecoverableItems | Format-List
```

This example gets folder statistics for Soumya Singhi's Recoverable Items folder and displays the folder name, folder path, number of items in the folder, and folder size in a table format.

```
Get-MailboxFolderStatistics -Identity "Soumya Singhi" -
FolderScope RecoverableItems | Format-Table
Name,FolderPath,ItemsInFolder,FolderAndSubfolderSize
```

For detailed syntax and parameter information, see Get-MailboxFolderStatistics.

Get Recoverable Items folder statistics for all mailboxes on litigation hold

This example retrieves a list of all mailboxes placed on litigation hold and retrieves mailbox folder statistics for the Recoverable Items folder and its subfolders for each mailbox. The **Identity** (mailbox folder identity) and the **FolderAndSubfolderSize** properties are displayed in a table format.

```
Get-Mailbox -ResultSize Unlimited -Filter
{LitigationHoldEnabled -eq $true} | Get-
MailboxFolderStatistics | Format-Table
Identity,FolderAndSubfolderSize
```

For detailed syntax and parameter information, see Get-Mailbox and Get-MailboxFolderStatistics.

Client Access server

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-02*

For Microsoft Exchange 2013, there have been major architectural changes to the Exchange server roles. Instead of the five server roles that were present in Exchange 2010 and Exchange 2007, in Exchange 2013, the number of server roles has been reduced to three: the Client Access server and the Mailbox server, and with Service Pack 1, the Edge Transport server role.

Note:

Exchange 2013 can also work with the Exchange 2010 Edge Transport server role.

The Exchange 2013 Mailbox server includes all many of the server components found in Exchange 2010: client access protocols, transport services, mailbox databases, and Unified Messaging services (the Client Access server redirects SIP traffic generated from incoming calls to the Mailbox server). For more information about the Exchange 2013 Mailbox server, see Mailbox server.

The Client Access server provides authentication, proxy, and limited redirection services, and offers all the usual client access protocols: HTTP, POP, IMAP, and SMTP. The Client Access server is a thin and stateless server that doesn't do any data rendering. There's never anything queued or stored on the Client Access server. For more information about the new Exchange 2013 architecture, see What's new in Exchange 2013.

Warning:

Client Access servers are not supported in perimeter networks, and they must be deployed within your internal Active Directory environment. Every Active Directory site that contains a Mailbox server must also contain a Client Access server.

As a result of these architectural changes, there have been some changes to client connectivity. First, RPC/TCP is no longer a supported direct access protocol. This means that all Outlook connectivity must occur using RPC over HTTPS (also known as Outlook Anywhere), or with Exchange 2013 SP1 and Outlook 2013 SP1, MAPI over HTTP. As a result of these changes, there's no need to have the RPC Client Access service on the Client Access server. In addition, fewer namespaces are required for a site-resilient solution than were required in Exchange 2010, and it's no longer necessary to provide affinity for the RPC Client Access service. Also, Outlook clients no longer connect to a server fully qualified domain name (FQDN) as they've done in all previous versions of Exchange. Using Autodiscover, Outlook finds a new connection point made up of the user's mailbox GUID + @ + the domain portion of the user's primary SMTP address. This change makes it much less likely that users will see the dreaded message "Your administrator has made a change to your mailbox." Only Outlook 2007 and later versions are supported with Exchange 2013.

Client Access server functionality

The Client Access server in Exchange 2013 functions much like a front door, admitting all client requests and routing them to the correct active Mailbox database. The Client Access server provides network security functionality such as Secure Sockets Layer (SSL) and client authentication, and manages client connections through redirection and proxy functionality. The Client Access server authenticates client connections and, in most cases, will proxy a request to the Mailbox server that houses the currently active copy of the database that contains the user's mailbox. In some cases, the Client Access server might redirect the request to a more suitable Client Access server, either in a different location or running a more recent version of Exchange Server.

The Client Access server has the following features:

- **Stateless server** In previous versions of Exchange, many of the Client Access protocols required session affinity. For example, Outlook Web App required that all requests from a particular client be handled by a specific Client Access server within a load balanced array of Client Access servers. In Exchange 2013, the Client Access server is stateless. In other words, because all processing for the mailbox happens on the Mailbox server, it doesn't matter which Client Access server in an array of Client Access servers receives each individual client request. This change in functionality means that session affinity is no longer required at the load balancer level. This allows inbound connections to Client Access servers to be balanced using simple techniques provided by load balancing technology such as DNS round-robin. It also allows hardware load balancing devices to support significantly more concurrent connections.
- **Connection pooling** The Client Access servers handle client authentication and send the AuthN data to the Mailbox server. The account used by the Client Access servers to connect to the Mailbox servers is a privileged account that's a member of the Exchange Servers group. This allows the Client Access servers to pool connections to the Mailbox servers effectively. An array of Client Access servers can handle millions of client connections from the Internet, but far fewer connections are used to proxy the requests to the Mailbox servers than in previous releases of Exchange. This improves processing efficiency and end-to-end latency.

Management tasks on the Client Access server

In Exchange 2013, there are several key tasks that can be performed on the Client Access server. The management of digital certificates is primarily performed on the Client Access server and some of the client protocol management for Exchange ActiveSync, POP3, and IMAP4 is also handled on the Client Access server.

Exchange Remote Connectivity

Analyzer

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-28*

The Microsoft Exchange Remote Connectivity Analyzer (ExRCA) helps you make sure that connectivity for your Exchange servers is set up correctly. If you're having problems, it can also help you find and fix these problems. The ExRCA website can run tests to check for Microsoft Exchange ActiveSync, Exchange Web Services, Microsoft Outlook, and Internet email connectivity.

Remote Connectivity Analyzer tests

You can perform several tests with the ExRCA. The following tests work on Exchange 2007 and later versions:

- Exchange ActiveSync
- Exchange Web Services
- Outlook
- Internet email

Exchange ActiveSync tests

You can run the following tests for Exchange ActiveSync:

- **Exchange ActiveSync:** This test simulates the steps that a mobile device uses to connect to an Exchange server using Exchange ActiveSync.
- **Exchange ActiveSync Autodiscover:** This walks through the steps an Exchange ActiveSync device uses to obtain settings from the Autodiscover service.

Exchange Web Services connectivity tests

The Exchange Web Services tests check the settings for many of the Exchange Web Services. You can run the following tests for ExchangeWeb Services:

- **Synchronization, Notification, Availability, and Automatic Replies:** These tests walk through many basic Exchange Web Services tasks to confirm that they're working. This is useful for IT administrators who want to troubleshoot external access using Entourage EWS or other Web Services clients.
- **Service Account Access (Developers):** This test verifies a service account's ability to access a specified mailbox, create and delete items in it, and access it via Exchange impersonation. This test is primarily used by application developers to test the ability to access mailboxes with alternate credentials.

Microsoft Office Outlook Connectivity tests

You can run the following tests for Outlook connectivity:

- **Outlook Anywhere (RPC over HTTP):** This test walks through the steps Outlook uses to connect via Outlook Anywhere (RPC over HTTP).
- **Outlook Autodiscover:** This test walks through the steps Outlook uses to obtain settings from the Autodiscover service. This test doesn't actually connect to a mailbox.

Internet email tests

You can run the following tests for Internet email:

- **Inbound SMTP E-Mail:** This test walks through the steps an Internet email server uses to send inbound SMTP email to your domain.
- **Outbound SMTP E-Mail:** This test checks your outbound IP address for certain requirements. This includes Reverse DNS, Sender ID, and RBL checks.
- **POP Email:** This test walks through the steps an email client uses to connect to a mailbox using POP3.
- **IMAP Email:** This test walks through the steps an email client uses to connect to a mailbox using IMAP.

Help Identify My Issue with Sending/ Receiving Email on a Mobile Device (Automatic Checks)

Mailbox and Client Access servers > Client Access server > Exchange Remote Connectivity Analyzer >

Topic Last Modified: *2014-07-25*

The check on this page will help identify some of the most common connectivity issues. You can use the automatic check below to validate your connectivity or help determine root cause.

You'll be asked to install an application on your machine and agree to the end user licensing agreement. Depending on the test, you'll be asked for a user account or domain name when the check runs.

Prerequisites

We'll check to see if you have Microsoft .NET Framework 4.5 installed. To obtain Microsoft .NET

Framework 4.5, go to the Microsoft Download Center. See additional perquisite details here.

Mobile Email Check

Application	Symptom	Check
Mobile Email	Cannot send or receive email on my mobile device	Run this check

Help Identify My Issue with Sending/ Receiving Email in Office Outlook (Automatic Checks)

Mailbox and Client Access servers > Client Access server > Exchange Remote Connectivity Analyzer >

Topic Last Modified: *2014-07-25*

The check on this page will help identify some of the most common connectivity issues. You can use the automatic check below to validate your connectivity or help determine root cause.

You'll be asked to install an application on your machine and agree to the end user licensing agreement. Depending on the test, you'll be asked for a user account or domain name when the check runs.

Prerequisites

We'll check to see if you have Microsoft .NET Framework 4.5 installed. To obtain Microsoft .NET Framework 4.5, go to the Microsoft Download Center. See additional perquisite details here.

Office Outlook Check

Application	Symptom	Check
Outlook	Cannot send or receive email in Outlook	Run this check

Outlook	I think my PC might not be configured correctly for Outlook	1. Sign in to Office 365 2. Select Tools 3. Select **Check your on=premises PC with the Office 365 Best Practices Analyzer**.

Azure: Help Identify My issue with Automatic Checks - DNS Records

Mailbox and Client Access servers > Client Access server > Exchange Remote Connectivity Analyzer >

Topic Last Modified: *2014-07-31*

One of the most common configuration setup issues is incorrectly configuring DNS records. You can use the automatic checks listed below to validate your configuration and help you update your environment.

If you already have an Office 365 user account, select Sign In. You don't need an Azure ID account. You might be asked for a user account again when the checks run. If so, your user account is in the format of username@youroffice365login.domain and your password.

Prerequisites

We'll check to see if you have Windows Azure Active Directory Sign-in Assistant and the Windows Azure Active Directory Module for Windows PowerShell installed.

The Windows Azure Active Directory Sign-in Assistant comes in two versions: 32 bit and 64 bit.

The Windows Azure Active Directory Module for Windows PowerShell comes in two versions: 32 bit and 64 bit.

DNS Records Checks

Application	Symptom	Check
Domains	My custom domain (e.g. contoso.com) doesn't seem to	Run this check

	be configured with Office 365	
Domains	My custom domain (e.g. contoso.com) doesn't seem to be configured with Office 365 (I used a CNAME record)	Run this check
Domains	My custom domain (e.g. contoso.com) doesn't seem to be configured with Office 365 (I used a TXT record)	Run this check
Instant Messaging	My users are having trouble getting their Lync client to work	Run this check
Instant Messaging	My users are having trouble getting their Lync client to work with other organizations	Run this check
Mail	I can't get Outlook to automatically configure with Office 365	Run this check
Mail	My email doesn't seem to be routing to Office 365	Run this check
Mail	My organization is getting a good deal of SPAM	Run this check
Mail	I can't seem to get Exchange Hybrid working	Run this check

Help Identify My Issues with Automatic Checks - 3rd Party Tools

Mailbox and Client Access servers > Client Access server > Exchange Remote Connectivity Analyzer >

Topic Last Modified: *2014-07-28*

The check(s) on this page will help identify some of the most common configuration challenges. You can use the automatic check(s) below to validate your configuration and help you update your environment.

You'll be asked to sign-in on a new page. If you already have an Office 365 user account select **sign-in**. You won't need an Azure ID account. You might be asked for a user account again when the checks run. Use your username@youroffice365login.domain and your password.

Prerequisites

We'll check to see if you have the necessary pre-requisites on your Windows 7 x64 or Windows Server 2008 R2 computer automatically.

3rd Party Tools Check

Application	Symptom	Check
3rd Party Tools	Im not sure what 3rd party integrations/applications I have installed in my messaging environment	Run this check

Help Identify My Issue with Automatic Checks - Active Directory

Mailbox and Client Access servers > Client Access server > Exchange Remote Connectivity Analyzer >

Topic Last Modified: *2014-07-31*

The check(s) on this page will help identify some of the most common configuration challenges. You can use the automatic check(s) below to validate your configuration and help you update your environment..

If you already have an Office 365 user account, select **Sign In**. You don’t need an Azure ID account.

You might be asked for a user account again when the checks run. If so, your user account is in the format of username@youroffice365login.domain and your password.

Prerequisites

We'll check to see if you have Windows Azure Active Directory Sign-in Assistant and the Windows Azure Active Directory Module for Windows PowerShell installed.

The Windows Azure Active Directory Sign-in Assistant comes in two versions: 32 bit and 64 bit.

The Windows Azure Active Directory Module for Windows PowerShell comes in two versions: 32 bit and 64 bit.

Active Directory Checks

Application	Symptom	Check
Exchange Version	I'm not sure if I have Exchange Server installed on-premises or what version I have.	Run this check
Credentials	I'm not sure I have the right credentials to move to Office 365	Run this check
3rd Party Tools	I'm not sure what 3rd party integrations/applications I have installed in my messaging environment.	Run this check
Shared Mailboxes	I'd like to determine who is managing mailboxes for others (delegates).	Run this check

Help Idenfity My Issue with Automatic Checks - Adding Domains

Topic Last Modified: *2014-07-31*

The check(s) on this page will help identify some of the most common configuration challenges. You can use the automatic check(s) below to validate your configuration and help you update your environment.

If you already have an Office 365 user account, select **Sign In**. You don't need an Azure ID account. You might be asked for a user account again when the checks run. If so, your user account is in the format of username@youroffice365login.domain and your password.

Prerequisites

We'll check to see if you have Windows Azure Active Directory Sign-in Assistant and the Windows Azure Active Directory Module for Windows PowerShell installed.

The Windows Azure Active Directory Sign-in Assistant comes in two versions: 32 bit and 64 bit.

The Windows Azure Active Directory Module for Windows PowerShell comes in two versions: 32 bit and 64 bit.

Adding Domains Checks

Application	Symptom	Check
Domains	I'd like to determine what domains I have on-premises/I am not sure what domains I own.	Run this check
Domains	I'm not sure if I added and verified the right domains for my tenant.	Run this check
Domains	I need help making sure all of my DNS records are correct for Office 365.	1. Sign in to Office 365 2. Select Tools 3. Select **Check your on-premises PC with the Office 365 Best Practices Analyzer**

Help Identify My Issue with Automatic Checks - SSO

Topic Last Modified: *2014-07-31*

The check(s) on this page will help identify some of the most common configuration challenges. You can use the automatic check(s) below to validate your configuration and help you update your environment.

If you already have an Office 365 user account, select **Sign In**. You don't need an Azure ID account. You might be asked for a user account again when the checks run. If so, your user account is in the format of username@youroffice365login.domain and your password.

Prerequisites

We'll check to see if you have Windows Azure Active Directory Sign-in Assistant and the Windows Azure Active Directory Module for Windows PowerShell installed.

The Windows Azure Active Directory Sign-in Assistant comes in two versions: 32 bit and 64 bit.

The Windows Azure Active Directory Module for Windows PowerShell comes in two versions: 32 bit and 64 bit.

SSO Checks

Application	Symptom	Check
SSO	I'd like to determine what domains I have on-premises/ I'm not sure what domains I own.	Run this check
SSO	I'd like to determine if my domain joined computers meet the requirements for SSO.	Run this check
SSO	I'd like to confirm I have a	Run this check

	directory that supports SSO.	
SSO	I'm not sure if my domains meet the requirements for SSO.	Run this check

Help Identify My Issue with Automatic Checks - Quota

Mailbox and Client Access servers > Client Access server > Exchange Remote Connectivity Analyzer >

Topic Last Modified: *2014-07-28*

The check(s) on this page will help identify some of the most common configuration challenges. You can use the automatic check(s) below to validate your configuration and help you update your environment.

If you already have an Office 365 user account, select **Sign In**. You don't need an Azure ID account. You might be asked for a user account again when the checks run. If so, your user account is in the format of username@youroffice365login.domain and your password.

Prerequisites

We'll check to see if you have Windows Azure Active Directory Sign-in Assistant and the Windows Azure Active Directory Module for Windows PowerShell installed.

The Windows Azure Active Directory Sign-in Assistant comes in two versions: 32 bit and 64 bit.

The Windows Azure Active Directory Module for Windows PowerShell comes in two versions: 32 bit and 64 bit.

Hybrid Wizard Checks

Application	Symptom	Check
Schema	I'd like to determine if my schema is ready for Exchange Hybrid.	Run this check

Help Identify My Issue with Automatic Checks - Deploying Office

Mailbox and Client Access servers > Client Access server > Exchange Remote Connectivity Analyzer >

Topic Last Modified: *2014-07-31*

The check(s) on this page will help identify some of the most common configuration challenges. You can use the automatic check(s) below to validate your configuration and help you update your environment.

If you already have an Office 365 user account, select **Sign In**. You don't need an Azure ID account. You might be asked for a user account again when the checks run. If so, your user account is in the format of username@youroffice365login.domain and your password.

Prerequisites

We'll check to see if you have Windows Azure Active Directory Sign-in Assistant and the Windows Azure Active Directory Module for Windows PowerShell installed.

The Windows Azure Active Directory Sign-in Assistant comes in two versions: 32 bit and 64 bit.

The Windows Azure Active Directory Module for Windows PowerShell comes in two versions: 32 bit and 64 bit.

Hybrid Wizard Checks

Application	Symptom	Check
Client	I'm not sure if my domain computers are ready for Office.	Run this check

Help Identify My Issue with Automatic Checks - Migration

Mailbox and Client Access servers > Client Access server > Exchange Remote Connectivity Analyzer >

Topic Last Modified: *2014-07-28*

The check(s) on this page will help identify some of the most common configuration challenges. You can use the automatic check(s) below to validate your configuration and help you update your environment.

If you already have an Office 365 user account, select **Sign In**. You don't need an Azure ID account. You might be asked for a user account again when the checks run. If so, your user account is in the format of username@youroffice365login.domain and your password.

Prerequisites

We'll check to see if you have Windows Azure Active Directory Sign-in Assistant and the Windows Azure Active Directory Module for Windows PowerShell installed.

The Windows Azure Active Directory Sign-in Assistant comes in two versions: 32 bit and 64 bit.

The Windows Azure Active Directory Module for Windows PowerShell comes in two versions: 32 bit and 64 bit.

Adding Domains Checks

Application	Symptom	Check
Mailbox Sizes	I'd like to determine if my users are using my organization's default mailbox sizes so I can best plan my migration.	Run this check
Single Forest	I'm trying to evaluate if I can migrate my users' accounts with Directory Synchronization.	Run this check
Domain Functional Mode	I'm not sure if I'm ready to integrate ADFS 2.0/Single Sign-On	Run this check
Public Folders	I'm not sure if I have public folders to migrate to Office 365.	Run this check

Help Idenfity My Issue with Automatic Checks - Directory Synchronization

Topic Last Modified: *2014-07-28*

You can use the automatic checks below to validate your configuration and help you update your environment.

If you already have an Office 365 user account, select Sign In. You don't need an Azure ID account. You might be asked for a user account again when the checks run. If so, your user account is in the format of username@youroffice365login.domain and your password.

Prerequisites

We'll check to see if you have Windows Azure Active Directory Sign-in Assistant and the Windows Azure Active Directory Module for Windows PowerShell installed.

The Windows Azure Active Directory Sign-in Assistant comes in two versions: 32 bit and 64 bit.

The Windows Azure Active Directory Module for Windows PowerShell comes in two versions: 32 bit and 64 bit.

Directory Synchronization Checks

Application	Symptom	Check
Directory Sync	I'm not sure if all of my Active Directory user accounts meet the requirements for directory synchronization.	Run this check
Directory Sync	I'm not sure if my Active Directory domain functional levels are correctly set to Windows Server 2003 or above.	Run this check

Directory Sync	I'm not sure if directory synchronization occurred in the last three hours.	Run this check
Directory Sync	I'm not sure if my Active Directory has any duplicate user attributes that will block directory synchronization.	Run this check
Directory Sync	I'm not sure if directory synchronization is enabled in Office 365.	Run this check
Directory Sync	I'm not sure if I can deploy Office 365 quote limitations.	Run this check
Directory Sync	I'm not sure if I can deploy Office 365 and use the default directory synchronization setup.	Run this check
Directory Sync	I'm not sure if I use Active Directory and can support directory synchronization.	Run this check
Directory Sync	I'm not sure if all of my Active Directory groups meet the requirements for directory synchronization.	Run this check

Exchange 2013 Client Access server configuration

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-29*

After you've installed the Exchange 2013 Client Access server, there are a variety of configuration tasks that you can perform. Although the Client Access server in Exchange 2013 doesn't handle processing for the client protocols, several settings need to be applied to the Client Access server, including virtual directory settings and certificate settings.

Configuring server certificates

In Exchange 2013, you can use the Certificate Wizard to request a digital certificate from a certification authority. After you've requested a digital certificate, you'll need to install it on the Client Access server.

You don't need to install digital certificates on the Mailbox servers in your organization. A self-signed certificate is installed by default on the Mailbox servers, and it doesn't need to be replaced. The Client Access servers in your organization implicitly trust the self-signed certificate on the Mailbox servers. For more information, see Exchange 2013 certificate management UI.

Configuring virtual directories

There are several settings that you can configure on the virtual directories for the Offline Address Book (OAB), Exchange Web Services, Exchange ActiveSync, Outlook Web App, and the Exchange Administration Center. For additional information about virtual directory management, see Virtual directory management. You can configure the virtual directories using the following commands.

- Exchange 2013 provides two sets of HTTP connectivity settings for Outlook Anywhere configuration so that administrators may configure both an internal and external endpoint.

To configure Outlook Anywhere with a single URL for connectivity, you must provide the host name, indicate whether SSL is required, and specify an authpackage using the following command in the Exchange Management Shell:

```
Get-OutlookAnywhere | Set-OutlookAnywhere -InternalHostname
"internalServer.contoso.com" -
InternalClientAuthenticationMethod Ntlm -
InternalClientsRequireSsl $true -IISAuthenticationMethods
Negotiate,NTLM,Basic
```

You may also specify an externally reachable endpoint by using the following command in the Exchange Management Shell:

```
Get-OutlookAnywhere | Set-OutlookAnywhere -InternalHostname
"internalServer.contoso.com" -
InternalClientAuthenticationMethod Ntlm -
```

```
InternalClientsRequireSsl $true -ExternalHostname
"externalServer.company.com" -
ExternalClientAuthenticationMethod Basic -
ExternalClientsRequireSsl $true -IISAuthenticationMethods
Negotiate,NTLM,Basic
```

Tip:

While Exchange 2013 supports Negotiate for Outlook Anywhere HTTP authentication, this should only be used when all servers in the environment are running Exchange 2013.

- To configure Exchange ActiveSync, run the following command.

```
Set-ActiveSyncVirtualDirectory -Identity "<CAS2013>
\Microsoft-Server-ActiveSync (Default Web Site)" -
ExternalUrl "https://mail.contoso.com/Microsoft-Server-
ActiveSync"
```

- To configure the Exchange Web Services virtual directory, run the following command.

```
Set-WebServicesVirtualDirectory -Identity "<CAS2013>\EWS
(Default Web Site)" -ExternalUrl https://mail.contoso.com/
EWS/Exchange.asmx
```

- To configure the Offline Address Book, run the following command.

```
Set-OABVirtualDirectory -Identity "<CAS2013>\OAB (Default
Web Site)" -ExternalUrl "https://mail.contoso.com/OAB"
```

- To configure the Service Connection Point, run the following command.

```
Set-ClientAccessServer -Identity <CAS2013> -
AutoDiscoverServiceInternalURI https://
autodiscover.contoso.com/AutoDiscover/AutoDiscover.xml
```

Upgrade from Exchange 2007 and 2010 Client Access

Use this section to help you configure external access to protocols on the Exchange 2013 Client Access server. Run the Exchange Management Shell commands in the configuring virtual directories section above, as well as the commands below.

You'll have to run the following commands to configure the virtual directories for Exchange 2013.

1. To configure an external URL for Outlook Web App, run the following command in Exchange Management Shell.

```
Set-OwaVirtualDirectory "<CAS2013>\OWA (Default Web Site)"
```

```
-ExternalUrl https://mail.contoso.com/OWA
```

Run the following commands at a command prompt after you set the Outlook Web App virtual directory.

```
Net stop IISAdmin /y

Net start W3SVC
```

2. To configure external EAC access, run the following command in Exchange Management Shell.

```
Set-EcpVirtualDirectory "<CAS2013>\ECP (Default Web Site)"
-ExternalUrl https://mail.contoso.com/ECP -InternalURL
https://mail.contoso.com/ECP
```

3. To configure the Availability service, run the following command in Exchange Management Shell.

```
Set-WebServicesVirtualDirectory -Identity "<CAS2013>\EWS
(Default Web Site)" -ExternalURL https://mail.contoso.com/
EWS/Exchange.asmx
```

To verify that the external URL has been configured correctly for Exchange ActiveSync or Outlook Web App, you can use the Exchange Remote Connectivity Analyzer (ExRCA), a free Web-based tool provided by Microsoft. You can find ExRCA at: https://www.testexchangeconnectivity.com

To verify that authentication has been configured correctly for Exchange ActiveSync or Outlook Web App, you can also use ExRCA.

To verify that direct file access has been configured correctly for Outlook Web App, log on as a user to Outlook Web App using the public computer option and then try to access and save a file attached to an email message.

Configure protocols on the Exchange 2007 Client Access servers

You'll have to run the following commands to configure the virtual directories for Exchange 2007.

- To configure the external URL on the Exchange ActiveSync virtual directory, run the following command in Exchange Management Shell.

```
Set-ActiveSyncVirtualDirectory -Identity "<CAS2007>
\Microsoft-Server-ActiveSync (Default Web Site)" -
ExternalUrl https://mail.contoso.com/Microsoft-Server-
ActiveSync
```

- To configure the external URL on the Outlook Web App virtual directory, run the following

command in Exchange Management Shell.

```
Set-OwaVirtualDirectory -Identity "<CAS2007>\owa (Default
Web Site)" -ExternalUrl https://legacy.contoso.com/owa
```

Autodiscover service

Exchange Server 2013 > Mailbox and Client Access servers > Client Access server >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-07*

Microsoft Exchange 2013 includes a service named the Autodiscover service. This topic gives an overview of the service and explains how it works, how it configures Outlook clients, and what options there are for deploying the Autodiscover service in your messaging environment.

The Autodiscover service does the following:

- Automatically configures user profile settings for clients running Microsoft Office Outlook 2007, Outlook 2010, or Outlook 2013, as well as supported mobile phones. Phones running Windows Mobile 6.1 or a later version are supported. If your phone isn't a Windows Mobile phone, check your mobile phone documentation to see if it's supported.
- Provides access to Exchange features for Outlook 2007, Outlook 2010, or Outlook 2013 clients that are connected to your Exchange messaging environment.
- Uses a user's email address and password to provide profile settings to Outlook 2007, Outlook 2010, or Outlook 2013 clients and supported mobile phones. If the Outlook client is joined to a domain, the user's domain account is used.

Contents

Overview of the Autodiscover service

The Autodiscover service makes it easier to configure Outlook 2007, Outlook 2010, or Outlook 2013 and some mobile phones. You can't use the Autodiscover service with versions of Outlook earlier than Office Outlook 2007. In earlier versions of Microsoft Exchange (Exchange 2003 SP2 or earlier) and Outlook (Outlook 2003 or earlier), you had to configure all user profiles manually to access Exchange. Extra work was required to manage these profiles if changes occurred to the messaging environment. Otherwise, the Outlook clients would stop functioning correctly.

Through the Autodiscover service, Outlook finds a new connection point made up of the user's mailbox GUID + @ + the domain portion of the user's primary SMTP address. The Autodiscover service returns the following information to the client:

- The user's display name
- Separate connection settings for internal and external connectivity
- The location of the user's Mailbox server
- The URLs for various Outlook features that govern functionality such as free/busy information, Unified Messaging, and the offline address book
- Outlook Anywhere server settings

When a user's Exchange information is changed, Outlook automatically reconfigures the user's profile using the Autodiscover service. For example, if a user's mailbox is moved or the client can't connect to the user's mailbox or to available Exchange features, Outlook will contact the Autodiscover service and automatically update the user's profile to include the information that's required to connect to the mailbox and Exchange features.

Return to top

How the Autodiscover service works

When you install a Client Access server in Exchange 2013, a default virtual directory named Autodiscover is created under the default website in Internet Information Services (IIS). This virtual directory handles Autodiscover service requests from Outlook 2007, Outlook 2010, and Outlook 2013 clients and supported mobile phones under the following circumstances:

- When a user account is configured or updated
- When an Outlook client periodically checks for changes to the Exchange Web Services URLs
- When underlying network connection changes occur in your Exchange messaging environment

Additionally, a new Active Directory object named the service connection point (SCP) is created on the server where you install the Client Access server.

The SCP object contains the authoritative list of Autodiscover service URLs for the forest. You can use the **Set-ClientAccessServer** cmdlet to update the SCP object. For more information, see Set-ClientAccessServer.

Important:

Before you run the **Set-ClientAccessServer** cmdlet, make sure the Authenticated Users account on the Client Access server has Read permissions for the SCP object. If users don't have the correct permissions, they can't search for and read items.

For more information about SCP objects, see Publishing with Service Connection Points.

For external access, or using DNS, the client locates the Autodiscover service on the Internet by using the primary SMTP domain address from the user's email address.

Note:

You must provide a host service (SRV) resource record in DNS for Outlook clients to discover

the Autodiscover service by using DNS. For more information, see your Windows documentation for configuring DNS and also see the White Paper: Exchange 2007 Autodiscover Service.

Depending on whether you've configured the Autodiscover service on a separate site, the Autodiscover service URL will be either https://<*smtp-address-domain*>/autodiscover/autodiscover.xml or https://autodiscover.<*smtp-address-domain*>/autodiscover/autodiscover.xml, where ://<*smtp-address-domain*> is the primary SMTP domain address. For example, if the user's email address is tony@contoso.com, the primary SMTP domain address is contoso.com. When the client connects to Active Directory, the client looks for the SCP object created during Setup. In deployments that include multiple Client Access servers, an Autodiscover SCP object is created for each Client Access server. The SCP object contains the *ServiceBindingInfo* attribute with the fully qualified domain name (FQDN) of the Client Access server in the form https://CAS01/autodiscover/autodiscover.xml, where CAS01 is the FQDN for the Client Access server. Using the user credentials, the Outlook 2007, Outlook 2010, or Outlook 2013 client authenticates to Active Directory and searches for the Autodiscover SCP objects. After the client obtains and enumerates the instances of the Autodiscover service, the client connects to the first Client Access server in the enumerated list and obtains the profile information in the form of XML data that's needed to connect to the user's mailbox and available Exchange features.

Return to top

Deployment options for the Autodiscover service

The Autodiscover service must be deployed and configured correctly for Outlook 2007, Outlook 2010, and Outlook 2013 clients to automatically connect to Exchange features such as the offline address book, the Availability service, and Unified Messaging (UM). Deploying the Autodiscover service is only one step in making sure your Exchange services, such as the Availability service, can be accessed by Outlook 2007, Outlook 2010, or Outlook 2013 clients.

Configuring Autodiscover for cross-forest moves

The Autodiscover service can provide user profile information to connecting Outlook clients for mailboxes that have been moved from one Exchange forest to another. For this to happen, you must configure a mail-enabled user in both the original forest where the user's mailbox resided and in the target forest using the **New-MailUser** cmdlet. In the source forest, you should use the *ExternalEmailAddress* parameter in the cmdlet to specify the new email address of the mailbox in the target forest. For more information, see New-MailUser.

When you configure a mail-enabled user, the Autodiscover service in the original forest will redirect the authenticating user to the new email address in the target forest. The connecting Outlook client will then be redirected to the Client Access server in the target forest where the mailbox has been moved.

Return to top

Load balancing

Topic Last Modified: *2014-02-10*

Load balancing is a way to manage which of your servers receive traffic. Load balancing helps distribute incoming client connections over a variety of endpoints—for example, Client Access servers—to ensure that no one endpoint takes on a disproportional share of the load. Load balancing can also provide failover redundancy in case one or more endpoints fails. By using load balancing with Exchange Server 2013, you ensure that your users continue to receive Exchange service in case of a computer failure. Load balancing also enables your deployment to handle more traffic than one server can process while offering a single host name for your clients.

Load balancing serves two primary purposes. It reduces the impact of a single Client Access server failure within one of your Active Directory sites. In addition, load balancing ensures that the load on each of your Client Access servers is evenly distributed.

Exchange 2013 also includes the following solutions for switchover and failover redundancy:

- **High availability** Exchange 2013 uses database availability groups (DAGs) to keep multiple copies of your mailboxes on different servers synchronized. That way, if a mailbox database fails on one server, users can connect to a synchronized copy of the database on another server.
- **Site resilience** You can deploy two Active Directory sites in separate geographic locations, keep the mailbox data synchronized between the two, and have one of the sites take on the entire load if the other fails.
- **Online mailbox moves** In an online mailbox move, users can access their email accounts during the move. Users are locked out of their accounts for only a brief time at the end of the process, when the final synchronization occurs. You can perform online mailbox moves across forests or in the same forest.
- **Shadow redundancy** Shadow redundancy protects the availability and recoverability of messages while they're in transit. With shadow redundancy, the deletion of a message from the transport databases is delayed until the transport server verifies that all the next hops for that message have completed. If any of the next hops fail before reporting successful delivery, the message is resubmitted for delivery to the hop that didn't complete.

Architectural changes in load balancing for Exchange Server 2013

In Exchange Server 2010, client connections and processing were handled by the Client Access

server role. This required that both external and internal Outlook connections, as well as mobile device and third-party client connections, be load balanced across the array of Client Access servers in a deployment to achieve fault tolerance and efficient utilization of servers. Many Exchange 2010 Client Access protocols required affinity—a relationship between the client and a particular Client Access server. In particular, Outlook Web App, the Exchange Control Panel, Exchange Web Services, Outlook Anywhere, Outlook TCP/IP MAPI connections, Exchange ActiveSync, the Exchange Address Book Service, and Remote PowerShell either required or benefited from client-to-Client Access server affinity. Load balancing options in Exchange 2010 included the following:

- Windows Network Load Balancing with source IP affinity
- Hardware load balancing

Because of the differing needs of client protocols in Exchange 2010, we recommended using a Layer 7 load balancing solution. Layer 7, also known as application-level load balancing, allowed the load balancing solution to use complex rules to determine how to balance each request entering the system, given that the entire conversation between client and server would be available to the load balancer logic. These complex rules ensured that all requests from a specific client went to the same Client Access server endpoint. In Exchange 2010, if all requests from a specific client did not go to the same endpoint for protocols that required affinity, the user experience would be negatively affected. For more information about Exchange 2010 load balancing options, see Understanding Load Balancing in Exchange 2010.

In Exchange Server 2013, there are two primary types of servers—the Client Access server and the Mailbox server. The Client Access servers in Exchange 2013 serve as lightweight, stateless proxy servers, allowing clients to connect to Exchange 2013 Mailbox servers. Exchange 2013 Client Access servers provide a unified namespace and authentication. In addition, Exchange 2013 Client Access servers:

- Support proxy and redirection logic for client protocols.
- Support the use of Layer 4 load balancing.

With session affinity and Layer 7 load balancing, all requests between the client and the server are sent to the same endpoint, as required by various protocols. Requests are distributed at the application layer. With Layer 4 load balancing, the requests are distributed at the transport layer. The load balancing solution distributes requests from the client, which is aware of a single IP address (sometimes called the virtual IP address or VIP), to a set of servers that perform the work. The connection between the client and server must be established before the content of the request is determined, so the load balancer selects a server to receive the request before examining the content of the request. The selection of the target server can be made in various ways such as "round-robin," in which each inbound connection goes to the next target server in a circular list, or "least connections," in which the load balancer sends each new connection to the server that has the fewest established connections at that time. Now that session affinity isn't required, you have more flexibility, choice, and simplicity with respect to the load balancing architecture you deploy. Load balancing without session affinity lets you increase the capacity and utilization of the load balancer, because processing isn't used to maintain more involved affinity options such as cookie-based load balancing or Secure Sockets Layer (SSL) session ID.

Client Access server arrays and Exchange 2013

In Exchange 2010, we introduced the concept of a Client Access array. After a Client Access array was configured for an Active Directory site, all Client Access servers in the site automatically became members of the array. In current builds of Exchange 2013, no configuration of a Client Access array is required, because the deployment of a load balanced and highly available service is much simpler.

Load balancing solutions

The use of hardware load balancers is still supported for Exchange 2013. For information about the hardware load balancing solutions that have completed solution testing with Exchange 2010 and will likely work just as well with Exchange 2013, see Exchange Server 2010 load balancer deployment. Keep in mind that this page shows the more complex Layer-7 configuration of hardware load balancers with Exchange 2010. Load balancing Exchange 2013 traffic can be much simpler, given the architectural changes discussed earlier in this topic. Rather than configuring session affinity for each of the Exchange protocols, inbound connections to Exchange 2013 Client Access Servers can be directed to an available server by the load balancer with no additional affinity processing necessary. The hardware load balancer still has an important role in providing high availability of the Exchange service because it can detect when a specific Client Access server has become unavailable and remove it from the set of servers that will handle inbound connections.

Windows Network Load Balancing

Windows Network Load Balancing (WNLB) is a common software load balancer used for Exchange servers. There are several limitations associated with deploying WNLB with Microsoft Exchange.

- WNLB can't be used on Exchange servers where mailbox DAGs are also being used because WNLB is incompatible with Windows failover clustering. If you're using an Exchange 2013 DAG and you want to use WNLB, you need to have the Client Access server role and the Mailbox server role running on separate servers.
- WNLB doesn't detect service outages. WNLB only detects server outages by IP address. This means that if a particular web service, such as Outlook Web App, fails, but the server is still functioning, WNLB won’t detect the failure and will still route requests to that Client Access server. Manual intervention is required to remove the Client Access server experiencing the outage from the load balancing pool.
- Using WNLB can result in port flooding, which can overwhelm networks.
- Because WNLB only performs client affinity using the source IP address, it's not an effective solution when the source IP pool is small. This can occur when the source IP pool is from a remote network subnet or when your organization is using network address translation.

Configuring Kerberos authentication for load-balanced Client Access servers

Applies to: *Exchange Server 2010 SP2*

Topic Last Modified: *2014-06-12*

To use Kerberos authentication with a load-balanced array of Client Access servers, you'll first need to complete several configuration steps. For more information about how to use Kerberos with a Client Access server array or a load-balancing solution, see **Using Kerberos with a Client Access Server Array or a Load-Balancing Solution**.

Contents

Create the Alternate Service Account Credential in Active Directory

All computers within the Client Access server array must share a service account. In addition, any Client Access servers that may be called on in a fallback scenario must also share the same service account. In general, it's sufficient to have a single service account per forest. This account is referred to as the alternate service account credential (ASA credential).

Note:
If your deployment is complex and extends beyond a single-site family, has multiple-site families configured, has administrator delegation issues, or has multiple forest segments on different Exchange deployment schedules, you may have to create additional accounts. The **RollAlternateServiceAccountCredential**.ps1 script must be run separately for every account created.

Credential Type

You can create a computer account or a user account for the alternate service account. Because a computer account doesn't allow interactive logon, it may have simpler security policies than a user account and therefore be preferable as an ASA credential. If you create a computer account, the password doesn't actually expire, but we still recommend updating the password periodically. Local group policy can specify a maximum account age for computer accounts and there might be scripts scheduled to periodically delete computer accounts that do not meet current policies. Periodically updating the password for computer accounts will ensure that your computer accounts aren't deleted for not meeting local policy. Your local security policy will determine when the password needs to be changed.

Credential Name

There are no particular requirements for the name of the ASA credential. You can use any name that conforms to your naming scheme.

Groups and Roles

The ASA credential doesn't need special security privileges. For enhanced security, it's recommended that you remove this account from as many security groups as possible and give it the lowest possible set of privileges and roles.

Password

The password you provide when you create the account will never actually be used. Instead, the script will reset the password. So when you create the account, you can use any password that conforms to your organization's password requirements.

Cross-Forest Scenarios

If you have a cross-forest or resource-forest deployment, and users are outside the Active Directory forest that contains Exchange, you must configure cross-forest trusts.

If your Exchange servers are in Forest A, and your users are in Forest B, the Forest A inbound trust UPN suffixes that match the name suffixes you'll be handling from Forest B must be explicitly set for the Forest A inbound trust. This can be done using the Active Directory Domains and Trusts snap-in. Forest B must also be updated to ensure that the name-suffix routing rules are correctly configured under Incoming Trusts.

In addition, if there are users in Forest A that must authenticate to services in Forest B, the name-suffix routing rules for Forest A incoming trusts should also be configured to enable all name suffixes used in forest.

Convert the Offline Address Book Virtual Directory to an Application

Out of the box, the Offline Address Book virtual directory isn't a Web application, so it isn't controlled by the Microsoft Exchange Service Host service. Therefore, Kerberos authentication requests to the Offline Address Book virtual directory can't be decrypted by the ASA credential.

To convert the Offline Address Book virtual directory to a Web application, run the ConvertOABDir.ps1 script on each Client Access server member. The script will also create a new application pool for the Offline Address Book virtual directory. The script is located in the Exchange 2010 SP2 Scripts directory, or you can download the script here.

Running the Script

As soon as the ASA credential for Kerberos has been configured, you can run the AlternateServiceAccount credential script in the Exchange Management Shell. For more information, see **Using the RollAlternateserviceAccountPassword.ps1 Script in the Shell**. When the script has run, we recommend that you verify that all of the targeted servers have been properly updated.

Note:
The script is provided in English only.

For help troubleshooting script errors, see **Troubleshooting the RollAlternateServiceAccountPassword.ps1 Script**.

Associate Service Principal Names with the Alternate Service Account

After you create the alternate service account, you must associate Exchange Service Principal Names (SPNs) with the service account. The list of Exchange SPNs may vary with your configuration, but should include at least the following:

- **http/** For Exchange Web Services and the Autodiscover service
- **exchangeMDB/** For RPC Client Access
- **exchangeRFR/** For the Address Book service
- **exchangeAB/** For the Address Book service

The SPNs must be formed to match the service name at the network load balancer rather than at individual servers. For example, you might associate the following SPNs with the ASA account:

- http/autodiscover.contoso.com
- http/mail.contoso.com

- exchangeMDB/mail.contoso.com
- exchangeRFR/mail.contoso.com
- exchangeAB/mail.contoso.com

Before you configure the SPNs, verify that the target SPNs aren't already configured on a different account in the forest. The ASA credential must be the only account in the forest with these SPNs associated with it. You can verify no other account in the forest has the SPNs associated with it by running the following command from the command line.

```
Setspn -q
```

Run this command once for every target SPN. The command should return nothing. If it returns something, another account is already associated with the SPN.

Run the following command from the command line to set the SPNs on the shared ASA credential. This command must be run once for every target SPN.

```
Setspn -S exchangeMDB/mail.contoso.com
newSharedServiceAccountName
```

When you've set the SPNs, verify that they've been added by using the following command.

```
Setspn -L newSharedServiceAccountName$
```

Note:

We recommend that SPNs for the alternate service account include records for the main site as well as any failover sites that are active and might be accessed by a client by name. Add the SPNs as previously described, but with the FQDN of the failover site. For more information about active/active and active/passive setup for Client Access server arrays, see Server Cluster Overview.

Validating Exchange Client Kerberos Authentication

After you've successfully configured Kerberos and deployed the RollAlternateServiceAccountCredential.ps1 script, verify that clients can authenticate successfully.

Validating Authentication from Outlook

To ensure that Outlook is able to connect to the Client Access servers with Kerberos authentication, follow these steps:

1. Confirm that Outlook is configured to point to the correct load-balanced Client Access server array.
2. Configure the e-mail account server security settings to use logon network security **Kerberos Password Authentication** instead of **Negotiate Authentication**.
3. Confirm that Outlook Anywhere isn't enabled for the client. If Outlook fails to authenticate by

using Kerberos, it will try to fall back to Outlook Anywhere, so Outlook Anywhere should be disabled for this test.

4. If Outlook connects successfully, Kerberos has been configured correctly.

Validating Using the Test-OutlookConnectivity Cmdlet

To test whether Kerberos is working, use the **Test-OutlookConnectivity** cmdlet. This is the best way to see if TCP connectivity can be established. By default, the cmdlet will use Negotiate authentication for a TCP connection. So if Kerberos is configured, it will be used. The file KLIST.exe can be used to view the Kerberos tickets on the computer. This can be run from the Client Access server itself, as well as from an automated monitoring tool such as SCOM. When using the **Test-OutlookConnectivity** cmdlet, be sure that the Mailbox database has the RPCClientAccessServer property set to the Client Access server array name. Otherwise the cmdlet will not test the shared ASA credential functionality.

```
Test-OutlookConnectivity -Identity administrator -
MailboxCredential $c -Protocol tcp
```

To make sure that the connection is made using Kerberos, view KLIST.exe to see if there are Kerberos tickets associated with the new SPNs that were added.

Validating Kerberos from the Client Access Server

To confirm that Kerberos is working correctly from the Client Access server, you can examine the protocol logs to verify successful Kerberos connections. You can use these logs in addition to the other validations to confirm that Kerberos is being used.

- On the Client Access server, examine the Address Book protocol logs. These logs are typically located at the following path: C:\Program Files\Microsoft\Exchange server\v14\Logging \AddressBook Service
- Examine the latest log file and look for the word Kerberos after the script has been run. If you see Kerberos traffic, a connection has been made successfully. The line in the log file should look something like the following.

```
2010-06-11T22:58:49.799Z,9,0,/o=First Organization/
ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
cn=Recipients/
cn=Administrator,,2001:4898:f0:3031:99f:ce35:750a:8b09,EXCH
-A-363,ncacn_ip_tcp,Bind,,6,,,Kerberos,
```

If you see the word Kerberos, then the server is successfully creating Kerberos authenticated connections. For more information about the Address Book service log, see **Understanding the Address Book Service**.

Troubleshooting Authentication Failures

There are several common problems that may occur when you're configuring Kerberos authentication.

Outlook Clients Configured to use Kerberos Authentication Only Can't Connect

If your Outlook client that's configured to use only Kerberos authentication can't connect, follow these troubleshooting steps:

1. Configure Outlook to use NTLM authentication only, and then verify connectivity. If a connection can't be made, verify that the Client Access server array is available or that network connectivity is stable.

If NTLM connectivity is successful, but Kerberos is not, verify that the SPNs aren't registered on any other account besides the alternate service account. Make sure that the Exchange SPNs are registered on the account used by the shared alternate service account by using the setSPN query command as described earlier in this topic.

2. Make sure that the password is coordinated between all Client Access servers and Active Directory. To do this, run the script in attended mode and have it generate a new password.
3. Make sure that the Microsoft Exchange Address Book service is running on your Client Access servers.
4. If authentication still isn't successful, make sure that the virtual directories for the services you want to access with Kerberos have Integrated Windows authentication enabled. You can use the Get-VirtualDirectory cmdlets to verify the authentication methods. For more information on virtual directories, see **Understanding Outlook Web App Virtual Directories** and **Understanding Exchange Web Services Virtual Directories**.

Autodiscover Service Failures

If you notice the following Autodiscover service failure, it's probably because the Autodiscover service request header contains a large Kerberos authentication ticket that caused the header size to exceed the limit configured by the Internet Information Services (IIS) server. The error will be similar to the following.

```
HTTP/1.1 400 Bad Request
Content-Type: text/html; charset=us-ascii
Server: Microsoft-HTTPAPI/2.0
Date: Tue, 09 Mar 2010 18:06:18 GMT
Connection: close
Content-Length: 346
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN""http://
www.w3.org/TR/html4/strict.dtd">
<HTML><HEAD><TITLE>Bad Request</TITLE>
<META HTTP-EQUIV="Content-Type" Content="text/html;
charset=us-ascii"></HEAD>
<BODY><h2>Bad Request - Request Too Long</h2>
<hr><p>HTTP Error 400. The size of the request headers is
too long.</p>
</BODY></HTML>
```

To fix this error, increase the IIS header size limit. For more information, see IIS documentation.

Ongoing Maintenance of the ASA Credential

If your local password on the shared ASA credential must be refreshed periodically, see **Using the RollAlternateserviceAccountPassword.ps1 Script in the Shell** for instructions for setting up a scheduled task to perform regular password maintenance. Be sure to monitor the scheduled task to ensure timely password rollovers and prevent possible authentication outages.

Turning Kerberos Authentication Off

To revert your Client Access server array so that it doesn't use Kerberos, remove the SPNs from the shared service account. If the SPNs are removed, Kerberos authentication won't be attempted by your clients, and clients configured to use Negotiate authentication will use NTLM. Clients configured to use only Kerberos will be unable to connect. Once the SPNs are removed you should also delete the shared service account. You can use the maintenance script to clean out credentials from all Client Access server array members by using the *toEntireForest* parameter and using the -copy from server parameter to specify a server that does not have any Kerberos credentials.

Digital certificates and SSL

Exchange Server 2013 > Mailbox and Client Access servers > Client Access server >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-08-26*

Secure Sockets Layer (SSL) is a method for securing communications between a client and a server. For Exchange Server 2013, SSL is used to help secure communications between the server and clients. Clients include mobile phones, computers inside an organization's network, and computers outside an organization's network.

By default, when you install Exchange 2013, client communications are encrypted using SSL when you use Outlook Web App, Exchange ActiveSync, and Outlook Anywhere.

SSL requires you to use digital certificates. This topic summarizes the different types of digital certificates and information about how to configure Exchange 2013 to use these types of digital certificates.

Contents

Overview of digital certificates

Digital certificates are electronic files that work like an online password to verify the identity of a user or a computer. They're used to create the SSL encrypted channel that's used for client communications. A certificate is a digital statement that's issued by a certification authority (CA) that vouches for the identity of the certificate holder and enables the parties to communicate in a secure manner using encryption.

Digital certificates do the following:

- They authenticate that their holders—people, websites, and even network resources such as routers—are truly who or what they claim to be.
- They protect data that's exchanged online from theft or tampering.

Digital certificates can be issued by a trusted third-party CA or a Windows public key infrastructure (PKI) using Certificate Services, or they can be self-signed. Each type of certificate has advantages and disadvantages. Each type of digital certificate is tamper-proof and can't be forged.

Certificates can be issued for several uses. These uses include web user authentication, web server authentication, Secure/Multipurpose Internet Mail Extensions (S/MIME), Internet Protocol security (IPsec), Transport Layer Security (TLS), and code signing.

A certificate contains a public key and attaches that public key to the identity of a person, computer, or service that holds the corresponding private key. The public and private keys are used by the client and the server to encrypt the data before it's transmitted. For Windows-based users, computers, and services, trust in a CA is established when there's a copy of the root certificate in the trusted root certificate store and the certificate contains a valid certification path. For the certificate to be valid, the certificate must not have been revoked and the validity period must not have expired.

Types of certificates

There are three primary types of digital certificates: self-signed certificates, Windows PKI-generated

certificates, and third-party certificates.

Self-signed certificates

When you install Exchange 2013, a self-signed certificate is automatically configured on the Mailbox servers. A self-signed certificate is signed by the application that created it. The subject and the name of the certificate match. The issuer and the subject are defined on the certificate. This self-signed certificate is used to encrypt communications between the Client Access server and the Mailbox server. The Client Access server trusts the self-signed certificate on the Mailbox server automatically, so no third-party certificate is needed on the Mailbox server. When you install Exchange 2013, a self-signed certificate is also created on the Client Access server. This self-signed certificate will allow some client protocols to use SSL for their communications. Exchange ActiveSync and Outlook Web App can establish an SSL connection by using a self-signed certificate. Outlook Anywhere won't work with a self-signed certificate on the Client Access server. Self-signed certificates must be manually copied to the trusted root certificate store on the client computer or mobile device. When a client connects to a server over SSL and the server presents a self-signed certificate, the client will be prompted to verify that the certificate was issued by a trusted authority. The client must explicitly trust the issuing authority. If the client confirms the trust, then SSL communications can continue.

Note:

By default, the digital certificate installed on the Mailbox server or servers is a self-signed certificate. You don’t need to replace the self-signed certificate on the Mailbox servers in your organization with a trusted third-party certificate. The Client Access server automatically trusts the self-signed certificate on the Mailbox server and no other configuration is needed for certificates on the Mailbox server.

Frequently, small organizations decide not to use a third-party certificate or not to install their own PKI to issue their own certificates. They might make this decision because those solutions are too expensive, because their administrators lack the experience and knowledge to create their own certificate hierarchy, or for both reasons. The cost is minimal and the setup is simple when you use self-signed certificates. However, it's much more difficult to establish an infrastructure for certificate life-cycle management, renewal, trust management, and revocation when you use self-signed certificates.

Windows public key infrastructure certificates

The second type of certificate is a Windows PKI-generated certificate. A PKI is a system of digital certificates, certification authorities, and registration authorities (RAs) that verify and authenticate the validity of each party that's involved in an electronic transaction by using public key cryptography. When you implement a PKI in an organization that uses Active Directory, you provide an infrastructure for certificate life-cycle management, renewal, trust management, and revocation. However, there is some additional cost involved in deploying servers and infrastructure to create and manage Windows PKI-generated certificates.

Certificate Services are required to deploy a Windows PKI and can be installed through **Add Or**

Remove Programs in Control Panel. You can install Certificate Services on any server in the domain.

If you obtain certificates from a domain-joined Windows CA, you can use the CA to request or sign certificates to issue to your own servers or computers on your network. This enables you to use a PKI that resembles a third-party certificate vendor, but is less expensive. These PKI certificates can't be deployed publicly, as other types of certificates can be. However, when a PKI CA signs the requestor's certificate by using the private key, the requestor is verified. The public key of this CA is part of the certificate. A server that has this certificate in the trusted root certificate store can use that public key to decrypt the requestor's certificate and authenticate the requestor.

The steps for deploying a PKI-generated certificate resemble those required for deploying a self-signed certificate. You must still install a copy of the trusted root certificate from the PKI to the trusted root certificate store of the computers or mobile devices that you want to be able to establish an SSL connection to Microsoft Exchange.

A Windows PKI enables organizations to publish their own certificates. Clients can request and receive certificates from a Windows PKI on the internal network. The Windows PKI can renew or revoke certificates.

Trusted third-party certificates

Third-party or commercial certificates are certificates that are generated by a third-party or commercial CA and then purchased for you to use on your network servers. One problem with self-signed and PKI-based certificates is that, because the certificate is not automatically trusted by the client computer or mobile device, you must make sure that you import the certificate into the trusted root certificate store on client computers and devices. Third-party or commercial certificates do not have this problem. Most commercial CA certificates are already trusted because the certificate already resides in the trusted root certificate store. Because the issuer is trusted, the certificate is also trusted. Using third-party certificates greatly simplifies deployment.

For larger organizations or organizations that must publicly deploy certificates, the best solution is to use a third-party or commercial certificate, even though there is a cost associated with the certificate. Commercial certificates may not be the best solution for small and medium-size organizations, and you might decide to use one of the other certificate options that are available.

Return to top

Choosing a certificate type

When you choose the type of certificate to install, there are several things to consider. A certificate must be signed to be valid. It can be self-signed or signed by a CA. A self-signed certificate has limitations. For example, not all mobile devices let a user install a digital certificate in the trusted root certificate store. The ability to install certificates on a mobile device depends on the mobile device manufacturer and the mobile service provider. Some manufacturers and mobile service

providers disable access to the trusted root certificate store. In this case, neither a self-signed certificate nor a certificate from a Windows PKI CA can be installed on the mobile device.

Default Exchange certificates

By default, Exchange installs a self-signed certificate on both the Client Access server and the Mailbox server so that all network communication is encrypted. Encrypting all network communication requires that every Exchange server have an X.509 certificate that it can use. You should replace this self-signed certificate on the Client Access server with one that is automatically trusted by your clients.

“Self-signed” means that a certificate was created and signed only by the Exchange server itself. Because it wasn't created and signed by a generally trusted CA, the default self-signed certificate won't be trusted by any software except other Exchange servers in the same organization. The default certificate is enabled for all Exchange services. It has a subject alternative name (SAN) that corresponds to the server name of the Exchange server that it's installed on. It also has a list of SANs that include both the server name and the fully qualified domain name (FQDN) of the server.

Although other Exchange servers in your Exchange organization trust this certificate automatically, clients like web browsers, Outlook clients, mobile phones, and other email clients in addition to external email servers won't automatically trust it. Therefore, consider replacing this certificate with a trusted third-party certificate on your Exchange Client Access servers. If you have your own internal PKI, and all your clients trust that entity, you can also use certificates that you issue yourself.

Certificate requirements by service

Certificates are used for several things in Exchange. Most customers also use certificates on more than one Exchange server. In general, the fewer certificates you have, the easier certificate management becomes.

IIS

All the following Exchange services use the same certificate on a given Exchange Client Access server:

- Outlook Web App
- Exchange Administration Center (EAC)
- Exchange Web Services
- Exchange ActiveSync
- Outlook Anywhere
- Autodiscover
- Outlook Address Book distribution

Because only a single certificate can be associated with a website, and because all these services are offered under a single website by default, all the names that clients of these services use must

be in the certificate (or fall under a wildcard name in the certificate).

POP/IMAP

Certificates that are used for POP or IMAP can be specified separately from the certificate that's used for IIS. However, to simplify administration, we recommend that you include the POP or IMAP service name in your IIS certificate and use a single certificate for all these services.

SMTP

A separate certificate can be used for each receive connector that you configure. The certificate must include the name that SMTP clients (or other SMTP servers) use to reach that connector. To simplify certificate management, consider including all names for which you have to support TLS traffic in a single certificate.

Digital certificates and proxying

Proxying is the method by which one server sends client connections to another server. In the case of Exchange 2013, this happens when the Client Access server proxies an incoming client request to the Mailbox server that contains the active copy of the client's mailbox.

When Client Access servers proxy requests, SSL is used for encryption but not for authentication. The self-signed certificate on the Mailbox server encrypts the traffic between the Client Access server and the Mailbox server.

Reverse proxies and certificates

Many Exchange deployments use reverse proxies to publish Exchange services on the Internet. Reverse proxies can be configured to terminate SSL encryption, examine the traffic in the clear on the server, and then open a new SSL encryption channel from the reverse proxy servers to the Exchange servers behind them. This is known as SSL bridging. Another way to configure the reverse proxy servers is to let the SSL connections pass straight through to the Exchange servers behind the reverse proxy servers. With either deployment model, the clients on the Internet connect to the reverse proxy server using a host name for Exchange access, such as mail.contoso.com. Then the reverse proxy server connects to Exchange using a different host name, such as the machine name of the Exchange Client Access server. You don't have to include the machine name of the Exchange Client Access server on your certificate because most common reverse proxy servers are able to match the original host name that's used by the client to the internal host name of the Exchange Client Access server.

SSL and split DNS

Split DNS is a technology that allows you to configure different IP addresses for the same host name, depending on where the originating DNS request came from. This is also known as split-

horizon DNS, split-view DNS, or split-brain DNS. Split DNS can help you reduce the number of host names that you must manage for Exchange by allowing your clients to connect to Exchange through the same host name whether they're connecting from the Internet or from the intranet. Split DNS allows requests that originate from the intranet to receive a different IP address than requests that originate from the Internet.

Split DNS is usually unnecessary in a small Exchange deployment because users can access the same DNS endpoint whether they're coming from the intranet or the Internet. However, with larger deployments, this configuration will result in too high of a load on your outgoing Internet proxy server and your reverse proxy server. For larger deployments, configure split DNS so that, for example, external users access mail.contoso.com and internal users access internal.contoso.com. Using split DNS for this configuration ensures that your users won't have to remember to use different host names depending on where they're located.

Remote Windows PowerShell

Kerberos authentication and Kerberos encryption are used for remote Windows PowerShell access, from both the Exchange Administration Center (EAC) and the Exchange Management Shell. Therefore, you won't have to configure your SSL certificates for use with remote Windows PowerShell.

Return to top

Digital certificates best practices

Although the configuration of your organization's digital certificates will vary based on its specific needs, information about best practices has been included to help you choose the digital certificate configuration that's right for you.

Best practice: Use a trusted third-party certificate

To prevent clients from receiving errors regarding untrusted certificates, the certificate that's used by your Exchange server must be issued by someone that the client trusts. Although most clients can be configured to trust any certificate or certificate issuer, it's simpler to use a trusted third-party certificate on your Exchange server. This is because most clients already trust their root certificates. There are several third-party certificate issuers that offer certificates configured specifically for Exchange. You can use the EAC to generate certificate requests that work with most certificate issuers.

How to select a certification authority

A certification authority (CA) is a company that issues and ensures the validity of certificates. Client software (for example, browsers such as Microsoft Internet Explorer, or operating systems such as Windows or Mac OS) have a built-in list of CAs they trust. This list can usually be configured to add

and remove CAs, but that configuration is often cumbersome. Use the following criteria when you select a CA to buy your certificates from:

- Ensure the CA is trusted by the client software (operating systems, browsers, and mobile phones) that will connect to your Exchange servers.
- Choose a CA that says it supports "Unified Communications certificates" for use with Exchange server.
- Make sure that the CA supports the kinds of certificates that you'll use. Consider using subject alternative name (SAN) certificates. Not all CAs support SAN certificates, and other CAs don't support as many host names as you might need.
- Make sure that the license you buy for the certificates allows you to put the certificate on the number of servers that you intend to use. Some CAs only allow you to put a certificate on one server.
- Compare certificate prices between CAs.

Best practice: Use SAN certificates

Depending on how you configure the service names in your Exchange deployment, your Exchange server may require a certificate that can represent multiple domain names. Although a wildcard certificate, such as one for *.contoso.com, can resolve this problem, many customers are uncomfortable with the security implications of maintaining a certificate that can be used for any subdomain. A more secure alternative is to list each of the required domains as SANs in the certificate. By default, this approach is used when certificate requests are generated by Exchange.

Best practice: Use the Exchange certificate wizard to request certificates

There are many services in Exchange that use certificates. A common error when requesting certificates is to make the request without including the correct set of service names. The certificate wizard in the Exchange Administration Center will help you include the correct list of names in the certificate request. The wizard lets you specify which services the certificate has to work with and, based on the services selected, includes the names that you must have in the certificate so that it can be used with those services. Run the certificate wizard when you've deployed your initial set of Exchange 2013 servers and determined which host names to use for the different services for your deployment. Ideally you'll only have to run the certificate wizard one time for each Active Directory site where you deploy Exchange.

Instead of worrying about forgetting a host name in the SAN list of the certificate that you purchase, you can use a certification authority that offers, at no charge, a grace period during which you can return a certificate and request the same new certificate with a few additional host names.

Best practice: Use as few host names as possible

In addition to using as few certificates as possible, you should also use as few host names as possible. This practice can save money. Many certificate providers charge a fee based on the number of host names you add to your certificate.

The most important step you can take to reduce the number of host names that you must have and, therefore, the complexity of your certificate management, is not to include individual server host names in your certificate's subject alternative names.

The host names you must include in your Exchange certificates are the host names used by client applications to connect to Exchange. The following is a list of typical host names that would be required for a company named Contoso:

- **Mail.contoso.com** This host name covers most connections to Exchange, including Microsoft Outlook, Outlook Web App, Outlook Anywhere, the Offline Address Book, Exchange Web Services, POP3, IMAP4, SMTP, Exchange Control Panel, and ActiveSync.
- **Autodiscover.contoso.com** This host name is used by clients that support Autodiscover, including Microsoft Office Outlook 2007 and later versions, Exchange ActiveSync, and Exchange Web Services clients.
- **Legacy.contoso.com** This host name is required in a coexistence scenario with Exchange 2007 and Exchange 2013. If you'll have clients with mailboxes on Exchange 2007 and Exchange 2013, configuring a legacy host name prevents your users from having to learn a second URL during the upgrade process.

Understanding wildcard certificates

A wildcard certificate is designed to support a domain and multiple subdomains. For example, configuring a wildcard certificate for *.contoso.com results in a certificate that will work for mail.contoso.com, web.contoso.com, and autodiscover.contoso.com.

Return to top

Create a digital certificate request

Mailbox and Client Access servers > Client Access server > Digital certificates and SSL >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-21*

In Exchange Server 2013, you can manage certificates using the EAC or the Shell. The EAC includes a new certificate management user interface. Through this new UI, you can create a new certificate, edit an existing certificate, or remove a certificate.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes plus time for the certification authority response.

- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Client Access server security" entry in the Clients and mobile devices permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to create a new certificate request

1. In the EAC, navigate to **Servers** > **Certificates**.
2. In the **Select server** list, select the server for which you want to create a certificate, and then click **Add +**.
3. In the **New Exchange certificate** wizard, choose either **Create a request for a certificate from a certification authority** or **Create a self-signed certificate**, and then select **Next**.
4. Enter a friendly name for the certificate and select **Next**.
5. If you didn't choose a self-signed certificate and you want a wildcard certificate, select the box marked **Request a wildcard certificate**, enter the root domain, for example *.contoso.com, and then select **Next**. If you chose a self-signed certificate, skip this step.
6. Select the servers that you want to apply this certificate to and select **Next**.
7. Specify the domains you want to be included in your certificate and then select **Next**.
8. Verify that the included domains are correct. If you chose a self-signed certificate, select **Finish**. Otherwise select **Next**.
9. Enter your organization name, department name, city or locality, state or province, and country or region, and then select **Next**.
10. Enter a location to save the certificate request and select **Finish**.

If you didn't select a self-signed certificate, you'll need to send the certificate request file to the certification authority for processing.

Use the Shell to create a new certificate request

Run the following commands.

```
$reqfile = New-ExchangeCertificate -GenerateRequest -
SubjectName "C=US,o=Contoso,cn=contosotocert" -DomainName
"contoso.com" -PrivateKeyExportable $true

$reqfile | out-file c:\certreq.txt
```

How do you know this worked?

If you created a self-signed certificate, the newly created certificate will appear in the certificate management UI. If you created a certificate request from a certification authority, the certificate request file will be in the location you specified. Send this file to the certification authority.

Exchange 2013 certificate management UI

Mailbox and Client Access servers > Client Access server > Digital certificates and SSL >

Topic Last Modified: *2014-04-21*

Managing certificates in an Exchange Server deployment is one of the most important administrative tasks. Ensuring that certificates are appropriately set up and configured is key to delivering a secure messaging infrastructure for the enterprise. In Exchange 2010, the Exchange Management Console (EMC) was the primary method of managing certificates. In Exchange 2013, certificate management functionality is provided in the Exchange Administration Console (EAC), the new Exchange 2013 administrative user interface. In Exchange 2013, the focus is on minimizing the number of certificates that an administrator must manage, minimizing the interaction the administrator must have with certificates, and allowing management of certificates from a central location.

Client Access server certificates

The Client Access server in Exchange 2013 is a stateless thin server designed to accept incoming client connections and proxy them to the correct Mailbox server. The Exchange Certificate Management UI on the Client Access server can help you with a variety of tasks, including requesting new certificates and renewing expired or soon–to-expire certificates.

Understanding the Certificate Management UI

You can access the Exchange Certificate Management UI through the EAC by selecting **Servers** and then **Certificates**. Using the management UI you can perform the following actions:

- **Create a new certificate** Selecting **Add +** launches the New Exchange Certificate wizard.
- **Edit an existing certificate** Selecting **Edit** on a valid certificate allows you to see the certificate's property page.
- **Delete a certificate** Selecting **Delete** when a certificate is selected launches the delete confirmation dialog box.

- **Perform additional actions** Selecting **More options** ••• allows you to export or import a certificate. If you want to export a certificate, the certificate must be valid.

Certificate expiration

In previous versions of Microsoft Exchange, you couldn't easily see when a digital certificate was nearing expiration. In Exchange 2013, the Notifications center displays warnings when a certificate stored on any Exchange 2013 Client Access server is about to expire.

Mailbox server certificates

One key difference between Exchange 2010 and Exchange 2013 is that the certificates that are used on the Exchange 2013 Mailbox server are self-signed certificates. Because all clients connect to an Exchange 2013 Mailbox server through an Exchange 2013 Client Access server, the only certificates that you need to manage are those on the Client Access server. The Client Access server automatically trusts the self-signed certificate on the Mailbox server, so clients will not receive warnings about a self-signed certificate not being trusted, provided that the Client Access server has a non-self-signed certificate from either a Windows certification authority (CA) or a trusted third party. There are no tools or cmdlets available to manage self-signed certificates on the Mailbox server. After the server has been properly installed, you should never need to worry about the certificates on the Mailbox server.

Self-signed certificate expiration

By default, the self-signed certificate that is installed on the Exchange 2013 Mailbox server will expire five years from the date of installation.

Certificate cmdlets

You can use the following cmdlets to manage digital certificates on an Exchange Client Access server:

- **Import-ExchangeCertificate** This cmdlet is used to import certificates to a server. You can import a CA-signed certificate (to complete a pending certificate signing request (CSR)) or a certificate with a private key (PKCS #12 files, generally with a .pfx extension, previously exported from a server along with the private key).
- **Remove-ExchangeCertificate** This cmdlet is used to remove certificates from a server.
- **Enable-ExchangeCertificate** This cmdlet is used to assign services to a certificate.
- **Get-ExchangeCertificate** This cmdlet is used to retrieve an Exchange certificate based on a variety of criteria.
- **New-ExchangeCertificate** This cmdlet is used to create a new self-signed certificate or a CSR.

Configure client-specific message size limits

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-20*

In Microsoft Exchange Server 2013, there are several different message size limits that apply to messages as they travel through your Exchange organization. For more information, see Message size limits.

However, there are client-specific message size limits you can configure for Outlook Web App and email clients that use ActiveSync or Exchange Web Services (EWS). If you change the Exchange organization-wide message size limits, you need to verify that the message size limits for Outlook Web App, ActiveSync, and Exchange Web Services are set accordingly. You configure these values in web.config files on Client Access servers and Mailbox servers. These limits are described in the following tables.

ActiveSync

Server role	Configuration file	Keys and default values	Size
Client Access	`%ExchangeInstallPath%FrontEnd\HttpProxy\Sync\web.config`	`maxRequestLength="10240"`	kilobytes
Mailbox	`%ExchangeInstallPath%ClientAccess\Sync\web.config`	`maxRequestLength="10240"`	kilobytes
Mailbox	`%ExchangeInstallPath%ClientAccess\Sync\web.config`	`<add key="MaxDocumentDataSize" value="10240000">`	bytes

Comments on ActiveSync limits

- To change the maximum message size for ActiveSync clients, you need to change these values in both files.

Exchange Web Services

Serve role	Configuration file	Keys and default values	Size
Client Access	`%ExchangeInstallPath%FrontEnd\HttpProxy\ews\web.config`	`maxAllowedContentLength="67108864"`	bytes

Mailbox	%ExchangeInstallPath%ClientAccess\exchweb\ews\web.config	maxAllowedContentLength="67108864"	bytes
Mailbox	%ExchangeInstallPath%ClientAccess\exchweb\ews\web.config	14 instances of maxReceivedMessageSize="67108864"	bytes

Comments on Exchange Web Services limits

- There are 14 separate instances of the value maxReceivedMessageSize="67108864" that correspond to different combinations of bindings (http and https) and authentication methods.
- To change the maximum message size for Exchange Web Services clients, you need to change the value of *maxAllowedContentLength* in both web.config files, and all 14 instances of maxReceivedMessageSize="67108864" in the web.config file on Mailbox servers.
- In the web.config file on Mailbox servers, there are also two instances of the value maxReceivedMessageSize="1048576" for **UMLegacyMessageEncoderSoap11Element** bindings that you don't need to modify.
- *maxRequestLength* is an ASP.NET setting that's present in both web.config files, but is not used by Exchange Web Services, so you don't need to modify it.

Outlook Web App

Server role	Configuration file	Keys and default values	Size
Client Access	%ExchangeInstallPath%FrontEnd\HttpProxy\owa\web.config	maxAllowedContentLength="35000000"	bytes
Client Access	%ExchangeInstallPath%FrontEnd\HttpProxy\owa\web.config	maxRequestLength="35000"	kilobytes
Mailbox	%ExchangeInstallPath%ClientAccess\Owa\web.config	maxAllowedContentLength="35000000"	bytes
Mailbox	%ExchangeInstallPath%ClientAccess\Owa\web.config	maxRequestLength="35000"	kilobytes
Mailbox	%ExchangeInstallPath%ClientAccess\Owa\web.config	2 instances of maxReceivedMessageSize="35000000"	bytes
Mailbox	%ExchangeInstallPath%ClientAccess\Owa\web.config	2 instances of maxStringContentLength="35000000"	bytes

Comments on Outlook Web App limits

- In the web.config file on Mailbox servers, there are two separate instances of the values maxReceivedMessageSize="35000000" and maxStringContentLength="35000000" that correspond to http and https bindings.

- To change the maximum message size for Outlook Web App clients, you need to change all of these values in both files, including both instances of *maxReceivedMessageSize* and *maxStringContentLength* in the `web.config` file on Mailbox servers.
- In the `web.config` file on Mailbox servers, there is also an instance of the value `maxStringContentLength="102400"` for the **MsOnlineShellService** binding that you don't need to modify.

For all message size limits, you need to set values that are larger than the actual sizes you want enforced. This increase in values is necessary to account for the inevitable message size increase that occurs after the message attachments and any other binary data are Base64 encoded. Base64 encoding increases the size of the message by approximately 33%, so the values you specify for any message size limits are approximately 33% larger than the actual usable message sizes. For example, if you specify a maximum message size value of 64 MB, you can expect a realistic maximum message size value of approximately 48 MB.

What do you need to know before you begin?

- Estimated time to complete: 15 minutes
- Exchange permissions don't apply to the procedures in this topic. These procedures are performed in the operating system of the Exchange Server.
- Changes you save to the Web.config configuration file are applied after you restart IIS.
- To allow for a 33% increase in size due to Base64 encoding, multiply your desired new maximum size value in megabytes by 4/3. To convert the value into kilobytes, multiply by 1024. To convert the value into bytes, multiply by 1048756 (1024*1024). Note that the size increase caused by Base64 encoding could be greater than 33%, and depends on several factors, for example, the attachment file size, type, compression, and the email client used to compose and send the message.
- Any customized per-server settings you make in Exchange XML application configuration files, for example, web.config files on Client Access servers or the EdgeTransport.exe.config file on Mailbox servers, will be overwritten when you install an Exchange Cumulative Update (CU). Make sure that you save this information so you can easily re-configure your server after the install. You must re-configure these settings after you install an Exchange CU.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Command Prompt to configure the client-specific message size limit

1. In a Command prompt window, open the appropriate web.config files in Notepad. For example,

to open the web.config files for Exchange Web services clients, run the following commands:

```
Notepad %ExchangeInstallPath%ClientAccess\exchweb\ews
\web.config
Notepad %ExchangeInstallPath%FrontEnd\HttpProxy\ews
\web.config
```

2. Find the relevant keys in the appropriate web.config files as described in the tables earlier in the topic. For example, for Exchange Web services clients, find the *maxAllowedContentLength* key in both files and all 14 instances of the value `maxReceivedMessageSize="67108864"` in the `web.config` file on Mailbox servers.

```
<requestLimits maxAllowedContentLength="67108864" />
...maxReceivedMessageSize="67108864"...
```

For example, to allow a Base64 encoded maximum message size of approximately 64 MB, change all instances of `67108864` to `89478486` (64*4/3*1048756):

```
<requestLimits maxAllowedContentLength="89478486" />
...maxReceivedMessageSize="89478486"...
```

3. When you are finished, save and close the web.config files.
4. Restart IIS by running the following command:

```
IISReset /noforce
```

How do you know this worked?

To verify that you have successfully configured the client-specific message size limit, you need to send a test message to and from a mailbox that's being accessed by the affected client. You can try a few smaller attachments or one large attachment so the test messages are approximately 33% less than the value you configured. For example, a configured value of 85 MB results in a realistic maximum message size of approximately 64 MB.

Availability service in Exchange 2013

Exchange Server 2013 > Mailbox and Client Access servers > Client Access server >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-10-10*

The Exchange 2013 Availability service makes free/busy information available to Microsoft Outlook and Outlook Web App clients. The Availability service improves information workers' calendaring

and meeting scheduling experience by providing secure, consistent, and up-to-date free/busy information.

Outlook and Outlook Web App use the Availability service to perform the following tasks:

- Retrieve current free/busy information for Exchange 2013 mailboxes
- Retrieve current free/busy information from other Exchange 2013 organizations
- Retrieve published free/busy information from public folders for mailboxes on servers that have previous versions of Exchange
- View attendee working hours
- Show meeting time suggestions

Contents

Overview of the Availability service

The Availability service retrieves free/busy information directly from the target mailbox for users on Exchange 2013, Exchange 2010, or Exchange 2007 and can be configured to retrieve free/busy information for users on earlier versions of Exchange. For topologies that have Exchange 2007, Exchange 2010, or Exchange 2013 mailboxes in which all clients are running Outlook 2007 or higher, the Availability service is used to retrieve free/busy information.

Outlook uses the Exchange Autodiscover service to obtain the URL of the Availability service. For more information about the Autodiscover service, see Autodiscover service.

You can use the Exchange Management Shell to configure the Availability service. You can't use the Exchange Administration Center (EAC) to configure the Availability service.

Information about away status

The Availability service also provides access to automatic-reply messages that users send when they are out of the office or away for an extended period of time.

Information workers use the Automatic Replies feature (formerly known as Out of Office) in Outlook and Outlook Web App to alert others when they're unavailable to respond to email messages. This functionality makes it easier to set and manage automatic-reply messages for both information workers and administrators.

Availability service API

The Availability service is part of the Exchange 2013 programming interface. It's available as a web service to let developers write third-party tools for integration purposes.

Availability service Network Load Balancing

Using Network Load Balancing (NLB) on your Client Access servers that are running the Availability service can improve performance and reliability for your users who rely on free/busy information. Outlook discovers the Availability service URL using the Autodiscover service. To use NLB with the Availability service, you must make changes to your configuration.

The internal URL is used from the intranet, and the external URL is used from the Internet. If you want to use the same URL for both internal and external traffic, make sure that DNS is correctly configured to route internal traffic directly to the internal URL. Also, make sure that the URL can be accessed both internally and externally. For the Autodiscover and Availability services to work, DNS must be configured so that mail.<*domain name*>.com and autodiscover.mail.<*domain name*>.com point to the virtual IP (VIP) of your load-balancing solution, where <*domain name*> is the name of your domain.

Note:

For more information, see Network Load Balancing Technical Reference and Network Load Balancing Clusters. You can also search for third-party load-balancing software websites.

Methods used to retrieve free/busy information

The following table lists the different methods used to retrieve free/busy information in different single-forest topologies.

Client	Mailbox retrieving free/busy information is running	Target mailbox is running	Free/busy retrieval method
Outlook 2013	Exchange 2013, Exchange 2010, or Exchange 2007	Exchange 2013, Exchange 2010, or Exchange 2007	The Availability service reads free/busy information from the target mailbox.
Outlook 2007	Exchange 2013, Exchange 2010, or Exchange 2007	Exchange 2010 or Exchange 2007	The Availability service reads free/busy information from the

			target mailbox.
Outlook 2007	Exchange 2010 or Exchange 2007	Exchange 2003	The Availability service makes HTTP connections to the / public virtual directory of the Exchange 2003 mailbox.
Outlook Web App	Exchange 2013, Exchange 2010, or Exchange 2007	Exchange 2013, Exchange 2010, or Exchange 2007	Outlook Web App in Exchange 2010 or Outlook Web Access in Exchange 2007 calls the Availability service API, which reads the free/busy information from the target mailbox.

Configure the Availability service for cross-forest topologies

Mailbox and Client Access servers > Client Access server > Availability service in Exchange 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-22*

The Availability service improves information workers' free/busy information by providing secure, consistent, and up-to-date free/busy information to clients that are running Microsoft Outlook. By default, this service is installed with Exchange Server 2013. In cross-forest topologies where all connecting clients are running Outlook, the Availability service is the only method of retrieving free/busy information. You can use the Shell to configure the Availability service for cross-forest topologies.

Note:

You can't use the EAC to configure the Availability service for cross-forest topologies.

Using the Availability service in trusted and untrusted forests

You can use the Availability service in cross-forest topologies across trusted or untrusted forests. The type of free/busy information that's available depends on if you're using a trusted or untrusted forest.

Trusted forests In trusted forests, you can configure the Availability service to retrieve free/busy information on a per-user basis. When the Availability service is configured to retrieve free/busy information on a per-user basis, the service can make cross-forest requests on behalf of a particular user. This allows a user in a remote forest to retrieve detailed free/busy information for someone who is not in the same forest.

Untrusted forests In untrusted forests, you can only configure the Availability service to retrieve free/busy information on an organization-wide basis. When the Availability service makes free/busy cross-forest requests at the organizational level, free/busy information is returned for each user in the organization. In untrusted forests, it isn't possible to control the level of free/busy information that's returned on a per-user basis.

Configuring Windows for cross-forest topologies

By default, a global address list (GAL) contains mail recipients from a single forest. If you have a cross-forest environment, we recommend using Microsoft Identity Lifecycle Manager (ILM) 2007 Feature Pack 1 (FP1) to ensure that the GAL in any given forest contains mail recipients from other forests. ILM 2007 FP1 creates mail users that represent recipients from other forests, thereby allowing users to view them in the GAL and send mail. For example, users in Forest A appear as a mail user in Forest B and vice versa. Users in the target forest can then select the mail user object that represents a recipient in another forest to send mail.

To enable GAL synchronization, you create management agents that import mail-enabled users, contacts, and groups from designated Active Directory services into a centralized metadirectory. In the metadirectory, mail-enabled objects are represented as mail users. Groups are represented as contacts without any associated membership. The management agents then export these mail users to an organizational unit in the specified target forest.

What do you need to know before you begin?

- Estimated time to complete each procedure: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Availability Service Permissions" entries in the Clients

and mobile devices permissions topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to configure per-user free/busy information in a trusted cross-forest topology

This example configures the Availability service to retrieve per-user free/busy information on a Mailbox server in the target forest.

```
Get-MailboxServer | Add-ADPermission -Accessrights
Extendedright -Extendedrights "ms-Exch-
EPI-Token-Serialization" -User "<Remote Forest Domain>
\Mailbox servers"
```

This example defines the free/busy access method that the Availability service uses on the local Mailbox server in the source forest. The local Mailbox server is configured to access free/busy information from the forest ContosoForest.com on a per-user basis. This example uses the service account to retrieve free/busy information.

```
Add-AvailabilityAddressSpace -Forestname ContosoForest.com
-AccessMethod PerUserFB -UseServiceAccount:$true
```

Note:

To configure bidirectional cross-forest availability, repeat these steps in the target forest.

If you choose to configure cross-forest availability with trust, and also choose to use a service account (instead of specifying organization-wide or per-user credentials), you must extend permissions as shown in the example in the "Use the Shell to configure trusted cross-forest availability with a service account" section. Performing that procedure in the target forest gives Mailbox servers in the source forest permission to serialize the original user context.

Use the Shell to configure trusted cross-forest availability with a service account

This example configures trusted cross-forest availability with a service account.

```
Get-MailboxServer | Add-ADPermission -Accessrights
Extendedright -Extendedright "ms-Exch-EPI-Token-
Serialization" -User "<Remote Forest Domain>\Exchange
servers"
```

For detailed information about syntax and parameters, see the following topics:

- Get-MailboxServer
- Add-ADPermission
- Add-AvailabilityAddressSpace
- Set-AvailabilityConfig

Use the Shell to configure organization-wide free/busy information in an untrusted cross-forest topology

This example sets the organization-wide account on the availability configuration object to configure the access level for free/busy information in the target forest.

```
Set-AvailabilityConfig -OrgWideAccount "Contoso.com\User"
```

This example adds the Availability address space configuration object for the source forest.

```
$a = Get-Credential (Enter the credentials for
organization-wide user in Contoso.com domain)
Add-AvailabilityAddressspace -Forestname Contoso.com -
Accessmethod OrgWideFB -Credential:$a
```

Edge Transport servers

Exchange Server 2013 >

Topic Last Modified: *2014-02-21*

Edge Transport servers minimize the attack surface by handling all Internet-facing mail flow, which provides SMTP (Simple Mail Transfer Protocol) relay and smart host services for your Exchange organization. Agents running on the Edge Transport server provide additional layers of message protection and security. These agents provide protection against viruses and spam and apply transport rules to control mail flow.

Because the Edge Transport server is installed in the perimeter network, it's never a member of your

organization's internal Active Directory forest and doesn't have access to Active Directory information. However, the Edge Transport server requires data that resides in Active Directory—for example, connector information for mail flow and recipient information for antispam recipient lookup tasks. This data is synchronized to the Edge Transport server by the Microsoft Exchange EdgeSync service (EdgeSync). EdgeSync is a collection of processes run on an Exchange 2013 Mailbox server to establish one-way replication of recipient and configuration information from Active Directory to the Active Directory Lightweight Directory Services (AD LDS) instance on the Edge Transport server. EdgeSync copies only the information that's required for the Edge Transport server to perform anti-spam configuration tasks and to enable end-to-end mail flow. EdgeSync performs scheduled updates so the information in AD LDS remains current.

You can install more than one Edge Transport server in the perimeter network. Deploying more than one Edge Transport server provides redundancy and failover capabilities for your inbound message flow. You can load balance the SMTP traffic to your organization among Edge Transport servers by defining more than one MX record with the same priority value for your mail domain. You can achieve consistency in the configuration among multiple Edge Transport servers by using cloned configuration scripts.

The Edge Transport server role lets you manage the following message-processing scenarios.

Internet mail flow

Edge Transport servers accept messages coming into the Exchange organization from the Internet. After the messages are processed by the Edge Transport server, where they are routed next depends on the configuration of your internal Exchange servers:

- If the Client Access server and the Mailbox server are installed on separate computers, mail is routed to the Transport service on the Mailbox server. The Client Access server is bypassed for inbound SMTP mail flow.
- If the Client Access server and the Mailbox server are installed on the same computer, mail is routed to the Front End Transport service on the Client Access server and then to the Transport service on the Mailbox server.

All messages sent to the Internet from inside the organization are routed to Edge Transport servers after the messages are processed by the Transport service on the Mailbox server. You can configure the Edge Transport server to use DNS to resolve MX resource records for external SMTP domains, or you can configure the Edge Transport server to forward messages to a smart host for DNS resolution.

Anti-spam and antivirus protection

In Exchange 2013, anti-spam and antivirus features provide services to block viruses and unsolicited commercial email (spam) at the network perimeter. Most viruses use spam-like tactics to gain access to your organization and to entice users to open an email message. If you can filter out most of your spam, you'll also be more likely to quarantine viruses before they enter your organization.

Spammers use a variety of techniques to send spam into your organization. Edge Transport servers help prevent users from ever receiving spam by providing a collection of agents that work together to provide different layers of spam filtering and protection. Establishing tarpitting intervals on connectors makes email harvesting attempts ineffective.

Edge Transport rules

Edge Transport rules are used to control the flow of messages sent to or received from the Internet. Edge Transport rules are configured on each Edge Transport server to help protect corporate network resources and data by applying an action to messages meeting specified conditions. Edge Transport rule conditions are based on data, such as specific words or text patterns in the message subject, body, header, or from address; the spam confidence level (SCL); or the attachment type. Actions determine how the message is processed when a specified condition is true. Possible actions include quarantining a message, dropping or rejecting a message, appending additional recipients, or logging an event. Optional exceptions exempt particular messages from having an action applied.

Address rewriting

Address rewriting presents a consistent email address appearance to external recipients. You configure address rewriting on Edge Transport servers to modify the SMTP addresses on inbound and outbound messages. Address rewriting is especially useful for newly merged organizations that want to present a consistent email address appearance.

For more information about address rewriting, see Address rewriting on Edge Transport servers.

Edge Subscriptions

Topic Last Modified: *2014-07-31*

Edge Transport servers minimize attack surface by handling all Internet-facing mail flow and providing SMTP relay and smart host services for your Exchange organization. Additional layers of message protection and security are provided by a series of agents running on the Edge Transport server in your organization's perimeter network. These agents support features that provide protection against viruses and spam and apply transport rules to control message flow.

Edge Subscriptions are used to populate the Active Directory Lightweight Directory Services (AD LDS) instance on the Edge Transport server with Active Directory data. Although creating an Edge Subscription is optional, subscribing an Edge Transport server to the Exchange organization

provides a simpler management experience and enhances antispam features. You need to create an Edge Subscription if you plan to use recipient lookup or safelist aggregation, or if you plan to help secure SMTP communications with partner domains by using Mutual Transport Layer Security (MTLS).

Contents

Edge Subscription process

An Edge Transport server doesn't have direct access to Active Directory. The configuration and recipient information the Edge Transport server uses to process messages is stored locally in AD LDS. Creating an Edge Subscription establishes secure, automatic replication of information from Active Directory to AD LDS. The Edge Subscription process provisions the credentials used to establish a secure LDAP connection between Exchange 2013 Mailbox servers and a subscribed Edge Transport server. The Microsoft Exchange EdgeSync service (EdgeSync) that runs on Mailbox servers performs periodic one-way synchronization to transfer up-to-date data to AD LDS. This reduces the administration tasks you perform in the perimeter network by letting you configure the Mailbox server and then synchronize that information to the Edge Transport server.

You subscribe an Edge Transport server to the Active Directory site that contains the Mailbox servers responsible for transferring messages to and from your Edge Transport servers. The Edge Subscription process creates an Active Directory site membership affiliation for the Edge Transport server. The site affiliation enables Mailbox servers in the Exchange organization to relay messages to the Edge Transport server for delivery to the Internet without having to configure explicit Send connectors.

One or more Edge Transport servers can be subscribed to a single Active Directory site. However, an Edge Transport server can't be subscribed to more than one Active Directory site. If you have more than one Edge Transport server deployed, each server can be subscribed to a different Active Directory site. Each Edge Transport server requires an individual Edge Subscription.

To deploy an Edge Transport server and subscribe it to an Active Directory site, follow these steps:

1. Install the Edge Transport server role.
2. Verify that the Mailbox servers and the Edge Transport server can locate one another using DNS name resolution.
3. On the Mailbox Server, configure the objects and settings to be replicated to the Edge Transport server.
4. On the Edge Transport server, create and export an Edge Subscription file.
5. Copy the Edge Subscription file to a Mailbox server or a file share that's accessible from the Active Directory site containing your Mailbox servers.

6. Import the Edge Subscription file to the Active Directory site.

What happens when you create a new Edge Subscription

When you create an Edge Subscription file (by running the **New-EdgeSubscription** cmdlet on the Edge Transport server), the following actions occur:

- An AD LDS account called the EdgeSync bootstrap replication account (ESBRA) is created. These ESBRA credentials are used to authenticate the first EdgeSync connection to the Edge Transport server. This account is configured to expire 24 hours after being created. Therefore, you need to complete the six-step subscription process described in the previous section within 24 hours. If the ESBRA expires before the Edge Subscription process is complete, you will need to run the **New-EdgeSubscription** cmdlet again to create a new Edge Subscription file.
- The ESBRA credentials are retrieved from AD LDS and written to the Edge Subscription file. The public key for the Edge Transport server's self-signed certificate is also exported to the Edge Subscription file. The credentials written to the Edge Subscription file are specific to the server that exported the file.
- Any previously created configuration objects on the Edge Transport server that will now be replicated to AD LDS from Active Directory are deleted from AD LDS, and the Exchange Management Shell cmdlets used to configure those objects are disabled. However, you can still use the **Get-*** cmdlets to view those objects. Running the **New-EdgeSubscription** cmdlet disables the following cmdlets on the Edge Transport server:
  - **Set-SendConnector**
  - **New-SendConnector**
  - **Remove-SendConnector**
  - **New-AcceptedDomain**
  - **Set-AcceptedDomain**
  - **Remove-AcceptedDomain**
  - **New-MessageClassification**
  - **Set-MessageClassification**
  - **Remove-MessageClassification**
  - **New-RemoteDomain**
  - **Set-RemoteDomain**
  - **Remove-RemoteDomain**

When you import the Edge Subscription file on the Mailbox server by running the **New-EdgeSubscription** cmdlet on the Mailbox server:

- The Edge Subscription is created, joining an Edge Transport server to an Exchange organization. EdgeSync will propagate configuration data to this Edge Transport Server, creating an Edge configuration object in Active Directory.
- Each Mailbox server in the Active Directory site receives notification from Active Directory that a new Edge Transport server has been subscribed. The Mailbox server retrieves the ESBRA from the Edge Subscription file. The Mailbox server then encrypts the ESBRA by using the public key of the Edge Transport server's self-signed certificate. The encrypted credentials are then written to the

Edge configuration object.

- Each Mailbox server also encrypts the ESBRA using its own public key and then stores the credentials in its own configuration object.
- EdgeSync replication accounts (ESRAs) are created in Active Directory for each Edge Transport-Mailbox server pair. Each Mailbox server stores its ESRA credentials as an attribute of the Mailbox server configuration object.
- Send connectors are automatically created to relay messages outbound from the Edge Transport server to the Internet, and inbound from the Edge Transport server to the Exchange organization.
- The Microsoft Exchange EdgeSync service that runs on Mailbox servers uses the ESBRA credentials to establish a secure LDAP connection between a Mailbox server and the Edge Transport server, and performs the initial replication of data. The following data is replicated to AD LDS:
  - Topology data
  - Configuration data
  - Recipient data
  - ESRA credentials
- The Microsoft Exchange Credential Service that runs on the Edge Transport server installs the ESRA credentials. These credentials are used to authenticate and secure later synchronization connections.
- The EdgeSync synchronization schedule is established.

The Microsoft Exchange EdgeSync service running on the Mailbox servers in the subscribed Active Directory site then performs one-way replication of data from Active Directory to AD LDS on a regular schedule. You can also use the **Start-EdgeSynchronization** cmdlet to override the EdgeSync synchronization schedule and immediately start synchronization.

For more information about ESRA accounts and how they're used to help secure the EdgeSync synchronization process, see Edge Subscription credentials.

This example subscribes an Edge Transport server to the specified site and automatically creates the Internet Send connector and the Send connector from the Edge Transport server to the Mailbox servers.

```
New-EdgeSubscription -FileData ([byte[]]$(Get-Content -Path
"C:\EdgeSubscriptionInfo.xml" -Encoding Byte -ReadCount 0))
-CreateInternetSendConnector $true -
CreateInboundSendConnector $true -Site "Default-First-Site-
Name"
```

Note:

The default values of the *CreateInternetSendConnector* and *CreateInboundSendConnector* parameters are both `$true`. They are shown here for demonstration only.

This example exports an Edge Subscription file.

```
New-EdgeSubscription -FileName "C:
```

```
\EdgeSubscriptionInfo.xml"
```

Note:

When the **New-EdgeSubscription** cmdlet is run on the Edge Transport server, you receive a prompt to acknowledge the commands that will be disabled and the configuration that will be overwritten on the Edge Transport server. To bypass this confirmation, you must use the *Force* parameter. This parameter is useful when you script the **New-EdgeSubscription** cmdlet. The *Force* parameter is also used to overwrite an existing file with the same name as the file that you're creating when you resubscribe an Edge Transport server.

For detailed syntax and parameter information, see New-EdgeSubscription.

Send connectors created during the Edge Subscription process

By default, when you complete the recommended Edge Subscription process by importing the Edge Subscription file to a Mailbox server, the Send connectors required to enable end-to-end mail flow between the Internet and the Exchange organization are created automatically, and any existing Send connectors on the Edge Transport server are deleted. In some scenarios, you may choose to suppress automatic creation of Send connectors and configure Send connectors manually. For more information about manually configuring Send connectors, see Manually configure Edge Transport server mail flow and Configure Internet mail flow through an Edge Transport server without using EdgeSync.

The Edge Subscription process provisions the following Send connectors:

- A Send connector configured to relay email messages from the Exchange organization to the Internet.
- A Send connector configured to relay email messages from the Edge Transport server to the Exchange organization.

Also, subscribing an Edge Transport server to the Exchange organization allows the Mailbox servers in the subscribed Active Directory site to use the intra-organization Send connector to relay messages to that Edge Transport server.

Automatically create an inbound Send connector to receive messages from the Internet

By default, when you run the **New-EdgeSubscription** cmdlet on the Mailbox server, the Inbound Send Connector parameter *CreateInboundSendConnector* is set to the value `$true`. This creates the Send connector needed to send messages to the Exchange organization. The following table shows the configuration of this Send connector.

Automatic inbound Send connector configuration

Property	Value
Name	EdgeSync - Inbound to *<Site Name>*
AddressSpaces	`SMTP:--;1` The `--` value in the address space represents all authoritative and internal relay accepted domains for the Exchange organization. Any messages the Edge Transport server receives for these accepted domains are routed to this Send connector and relayed to the smart hosts.
SourceTransportServers	*<Edge Subscription name>*
Enabled	True
DNSRoutingEnabled	False
SmartHosts	`--` The `--` value in the list of smart hosts represents all Mailbox servers in the subscribed Active Directory site. Any Mailbox servers you add to the subscribed Active Directory site after you establish the Edge Subscription don't participate in the EdgeSync synchronization process. However, they are automatically added to the list of smart hosts for the automatically created inbound Send connector. If more than one Mailbox server is located in the subscribed Active Directory site, inbound connections will be load balanced across the smart hosts.

You can't modify the address space or list of smart hosts at creation time for the automatically created inbound Send connector. However, you can set the *CreateInboundSendConnector* parameter to the value `$false` when you create an Edge Subscription. This allows you to manually configure a Send connector from the Edge Transport server to the Exchange organization.

Automatically create an outbound Send connector to send messages to the Internet

By default, when you run the **New-EdgeSubscription** cmdlet on the Mailbox server, the Outbound Send Connector parameter *CreateInternetSendConnector* is set to the value `$true`. This creates the Send connector needed to send messages to the Internet. The following table shows the default configuration of this Send connector.

Automatic Internet Send connector configuration

Property	Value
Name	EdgeSync - <*Site Name*> to Internet
AddressSpaces	`SMTP:*;100`
SourceTransportServers	<*Edge Subscription name*> **Note:** The name of the Edge Subscription is the same as the name of the subscribed Edge Transport server.
Enabled	True
DNSRoutingEnabled	True
DomainSecureEnabled	True

If more than one Edge Transport server is subscribed to the same Active Directory site, no additional Send connectors to the Internet are created. Instead, all Edge Subscriptions are added to the same Send connector as the source server. This load balances outbound connections to the Internet across the subscribed Edge Transport servers.

The outbound Send connector is configured to send email messages from the Exchange organization to all remote SMTP domains, using DNS routing to resolve domain names to MX resource records. For details about manually configuring a connector's configuration, see Manually configure Edge Transport server mail flow.

Microsoft Exchange EdgeSync service

After you subscribe an Edge Transport server to an Active Directory site, EdgeSync will replicate configuration and recipient data to the Edge Transport servers. The service replicates the following data from Active Directory to AD LDS:

- Send connector configuration
- Accepted domains
- Remote domains
- Message classifications
- Safe Senders Lists
- Blocked Senders Lists
- Recipients
- List of send and receive domains used in domain secure communications with partners
- List of SMTP servers listed as internal in your organization's transport configuration
- List of Mailbox servers in the subscribed Active Directory site

For details about the data replicated to AD LDS and how it's used, see EdgeSync replication data.

EdgeSync uses a mutually authenticated and authorized secure LDAP channel to transfer data from the Mailbox server to the Edge Transport server.

To replicate data to AD LDS, the Mailbox server binds to a global catalog server to retrieve updated data. EdgeSync initiates a secure LDAP session between a Mailbox server and the subscribed Edge Transport server over the non-standard TCP port 50636.

When you first subscribe an Edge Transport server to an Active Directory site, the initial replication that populates AD LDS with data from Active Directory can take five minutes or more, depending on the quantity of data in the directory service. After initial replication, EdgeSync only synchronizes new and changed objects, and removes any deleted objects.

Synchronization schedule

Different types of data synchronize on different schedules. The EdgeSync synchronization schedule specifies the maximum interval between EdgeSync synchronizations. EdgeSync synchronization occurs at the following intervals:

- Configuration data: 3 minutes.
- Recipient data: 5 minutes.
- Topology data: 5 minutes

If you want to change these intervals, use the **Set-EdgeSyncServiceConfig** cmdlet. Using the **Start-EdgeSynchronization** cmdlet on the Mailbox server to force Edge Subscription synchronization overrides the timer for the next scheduled EdgeSync synchronization, and starts EdgeSync immediately.

Selection of Mailbox servers

Each subscribed Edge Transport server is associated with a particular Active Directory site. If more than one Mailbox server exists in the site, any of these Mailbox servers can replicate data to the subscribed Edge Transport servers. To avoid contention among Mailbox servers when synchronizing, the preferred Mailbox server is selected as follows:

1. The first Mailbox server in the Active Directory site to perform a topology scan and discover the

new Edge Subscription performs the initial replication. Because this discovery is based on the timing of the topology scan, any Mailbox server in the site may perform the initial replication.
2. The Mailbox server performing the initial replication establishes an EdgeSync lease option and sets a lock on the Edge Subscription. The lease option establishes that particular Mailbox server as the preferred server providing synchronization services to that Edge Transport server. The lock prevents EdgeSync running on another Mailbox server from taking over the lease option.
3. The EdgeSync lease option lasts for one hour. During that hour, no other EdgeSync service can take over the option unless a manual synchronization is started before the end of the hour. If the preferred Mailbox server isn't available to provide EdgeSync service at the time manual synchronization is started, after a five-minute wait, the lock is released and another EdgeSync service can take over the lease option and perform synchronization.
4. Unless manual synchronization is started, synchronization occurs based on the EdgeSync synchronization schedule. If the preferred server isn't available when a scheduled synchronization occurs, after a five-minute wait, the lock is released and another EdgeSync service can take over the lease option and perform synchronization.

This method of locking and leasing prevents more than one instance of EdgeSync from pushing data to the same Edge Transport server at the same time.

Note:

If you also have Exchange 2010 or Exchange 2007 Mailbox servers in the subscribed Active Directory site, Exchange 2013 Mailbox servers will always take precedence and perform the replication.

Note:

When you subscribe an Edge Transport server to an Active Directory site, all Mailbox servers installed in that Active Directory site at that time can participate in the EdgeSync synchronization process. If one of those servers is removed, the EdgeSync service that's running on the remaining Mailbox servers will continue the data synchronization process. However, if you ;ater install new Mailbox servers in the Active Directory site, they won't automatically participate in EdgeSync synchronization. If you want to enable those new Mailbox servers to participate in EdgeSync synchronization, you will need to subscribe the Edge Transport server again.

The following table lists the EdgeSync properties related to locking and leasing. You can use the **Set-EdgeSyncServiceConfig** cmdlet to configure these properties.

EdgeSync lease properties

Parameter	Default value	Description
LockDuration	`00:05:00` (5 minutes)	This setting determines how long a particular EdgeSync service will acquire a lock. If the EdgeSync service on the Mailbox server that's holding

		this lock doesn't respond, after five minutes the EdgeSync service on another Mailbox server will take over the lease. Forcing immediate EdgeSync synchronization doesn't override this setting.
OptionDuration	`01:00:00` (1 hour)	This setting determines how long an EdgeSync service can declare a lease option on an Edge Transport server. If the EdgeSync service holding the lease is unavailable and doesn't restart during this option period, no other Exchange EdgeSync service will take over the lease option unless you force EdgeSync synchronization.
LockRenewalDuration	`00:01:00` (1 minute)	This setting determines how frequently the lock field is updated when an EdgeSync service has acquired a lock to an Edge Transport server.

Preparing to run the EdgeSync service

Before you can subscribe your Edge Transport server to your Exchange organization, you need to make sure your infrastructure and your Mailbox servers are prepared for the EdgeSync service. To prepare for EdgeSync synchronization, you need to:

- Verify that the perimeter network firewall separating the Edge Transport server from the Exchange organization is configured to enable communications through the correct ports. The Edge Transport server uses non-standard LDAP ports. If your environment requires specific ports, you can modify the ports used by AD LDS using the ConfigureAdam.ps1 script provided with

Exchange. For more information, see Modify AD LDS configuration. Modify the ports before you create the Edge Subscription. If you modify the ports after you create the Edge Subscription, you will need to remove the Edge Subscription and then create a new Edge subscription. By default, the following LDAP ports are used to access AD LDS:

  - **LDAP** Port 50389/TCP is used locally to bind to the AD LDS instance. This port doesn't have to be open on the perimeter network firewall.
  - **Secure LDAP** Port 50636/TCP is used for directory synchronization from Mailbox servers to AD LDS. This port must be open on the firewall for successful EdgeSync synchronization.
- Verify that DNS host name resolution is successful from the Edge Transport server to the Mailbox servers and from the Mailbox servers to the Edge Transport server.
- License the Edge Transport server. The licensing information for the Edge Transport server is captured when the Edge Subscription is created. Subscribed Edge Transport servers need to be subscribed to the Exchange organization after the license key has been applied on the Edge Transport server. If the license key is applied on the Edge Transport server after you perform the Edge Subscription process, licensing information will not be updated in the Exchange organization, and you will need to resubscribe the Edge Transport server.
- Configure the following transport settings for propagation to the Edge Transport server:
  - **Internal SMTP servers** Use the *InternalSMTPServers* parameter on the **Set-TransportConfig** cmdlet to specify a list of internal SMTP server IP addresses or IP address ranges to be ignored by the Sender ID and Connection Filtering agents on the Edge Transport server.
  - **Accepted domains** Configure all authoritative domains, internal relay domains, and external relay domains.
  - **Remote domains** Configure remote domain settings.

Return to top

Managing Edge Subscriptions

For step-by-step instructions on Edge Subscription management tasks, see Manage Edge Subscriptions.

EdgeSync replication data

Exchange Server 2013 > Edge Transport servers > Edge Subscriptions >

Topic Last Modified: *2014-02-21*

When you deploy an Edge Transport server, it doesn't have access to Active Directory. To perform recipient lookup and safelist aggregation tasks, and to implement domain security by using Mutual Transport Layer Security (MTLS) authentication, the Edge Transport server needs data from Active Directory. This data is replicated to the Edge Transport server using EdgeSync; the Edge Transport

server stores all replicated information in Active Directory Lightweight Directory Services (AD LDS).

This topic focuses on data replicated from Active Directory to AD LDS on an Edge Transport server subscribed to an Active Directory site. To learn more about EdgeSync and Edge Subscriptions, see Edge Subscriptions.

Four types of data are replicated to AD LDS and used by the Edge Transport server:

Edge Subscription information

Configuration information

Recipient information

Topology information

Edge Subscription information

Exchange 2013 extends both the Active Directory and AD LDS schemas to provide attributes on the **ms-Exch-ExchangeServer** object to represent the data needed to control EdgeSync synchronization. These attributes provide three functions to EdgeSync:

- Automatic provisioning and maintenance of the credentials used to help secure the LDAP connection between a Mailbox server and a subscribed Edge Transport server.
- Arbitrating the synchronization lock and lease process, ensuring that only one server at a time will try to synchronize with an individual Edge Transport server. For more information about the lock and lease process, see Edge Subscriptions.
- Optimizing EdgeSync synchronization to maintain a record of current synchronization status. Easily viewing synchronization status helps avoid excessive manual synchronization.

The schema extensions in the following table are specific to Edge Subscriptions. The values assigned to these attributes are maintained by the Edge Subscription and EdgeSync; they are not intended to be manually edited.

Edge Subscription schema extensions

Attribute name	Description
ms-Exch-Server-EKPK-Public-Key	The current public key for the certificate being used by the server. This value is stored by both Edge Transport servers and Mailbox servers. The public key is used to encrypt credentials used to authenticate the server during LDAP and SMTP communication.
ms-Exch-EdgeSync-Credential	The list of credentials EdgeSync uses to establish an authenticated LDAP session with

	AD LDS. On Mailbox servers, this attribute contains only credentials the Mailbox server uses to authenticate the subscribed Edge Transport servers. On Edge Transport servers, this attribute contains the credentials of each Mailbox server in the subscribed Active Directory site that participates in EdgeSync synchronization. This attribute is only present on Mailbox servers running EdgeSync synchronization and on subscribed Edge Transport servers.
ms-Exch-Edge-Sync-Lease	Used to arbitrate between Mailbox servers when more than one Mailbox server tries to replicate to the same Edge Transport server.
ms-Exch-Edge-Sync-Status	Only present in AD LDS on the Edge Transport server object. This attribute tracks the status of replication to an AD LDS instance and includes information about replication.

Return to top

Configuration information

When you subscribe an Edge Transport server to an organization, you can manage the configuration objects common to the Edge Transport server and the Exchange organization from inside the organization. These changes are then replicated to the Edge Transport server using EdgeSync. This process helps maintain a consistent configuration across all servers involved in message processing.

A subset of the configuration data for the Exchange organization must also be maintained on the Edge Transport server. During EdgeSync synchronization, the configuration data the Edge Transport server needs is written to the configuration partition of AD LDS. The configuration data written to AD LDS includes:

- **Mailbox servers** The fully qualified domain name (FQDN) of each Mailbox server in the subscribed Active Directory site is made available to the local AD LDS store on the Edge Transport server. This information is used to derive a list of smart host servers for the inbound Send connector.

- **Accepted domains** All authoritative, internal relay, and external relay domains configured for the Exchange organization are written to AD LDS. Having the accepted domains available to the Edge Transport server enables the Exchange organization to perform domain filtering and reject invalid SMTP traffic into their organization as early as possible. For more information about accepted domains, see Accepted domains.
- **Message classifications** If message classifications are available on the Edge Transport server, transport agents and content conversion can act on message classifications in the perimeter network. For example, the Attachment Filter agent can apply the Attachment Removed classification when it removes an attachment, sending informational text to a Microsoft Outlook user or an Outlook Web App user to tell the recipient what happened. Agents developed for use by third-party applications can use message classifications in a similar manner.
- **Remote domains** All remote domain entries configured for the Exchange organization are written to AD LDS. Remote domain entries control out-of-office message settings and message format settings for a remote domain. For more information about remote domains, see Remote domains.
- **Send connectors** By default, creating an Edge Subscription automatically creates the Send connectors required to enable end-to-end mail flow between the Exchange organization and the Internet at the time the Edge Transport Server is subscribed. Any existing Send connectors on the Edge Transport server are deleted. If you want to configure additional Send connectors, configure the Send connector inside the Exchange organization and then select the Edge Subscription as the source server for the connector. For more information, see Edge Subscriptions.
- **Internal SMTP servers** The value for the *InternalSMTPServers* attribute is stored on the **TransportConfig** object for both the Exchange organization and the local Edge Transport server. During EdgeSync synchronization, the value stored on the local Edge Transport server object is overwritten with the value stored on this object for the Exchange organization. This attribute specifies a list of internal SMTP server IP addresses or IP address ranges that should be ignored by Sender ID and connection filtering.
- **Domain Secure lists** The *TLSReceiveDomainSecureList* and the *TLSSendDomainSecureList* attributes are stored on the **TransportConfig** object for both the Exchange organization and the local Edge Transport server. During EdgeSync synchronization, the value stored on the local Edge Transport server object is overwritten with the value stored on this object for the Exchange organization. These attributes specify the list of remote domains configured for mutual TLS authentication.

Return to top

Recipient information

Recipient information replicated to AD LDS includes only a subset of recipient attributes. Only data required by the Edge Transport to perform certain antispam tasks is replicated. Recipient information replicated to AD LDS includes:

- **Recipients** The list of recipients in the Exchange organization is replicated to AD LDS. Each recipient is identified by assigned Active Directory GUID. If you configure a recipient's account to

deny receipt of mail from outside the organization, that recipient isn't replicated to AD LDS. If you disable or delete a recipient's mailbox, that mailbox is no longer replicated to AD LDS.

- **Proxy addresses** All proxy addresses assigned to each recipient are replicated to AD LDS as hashed data. This is a one-way hash using Secure Hash Algorithm (SHA)-256. SHA-256 generates a 256-bit message digest of the original data. Storing proxy addresses as hashed data helps secure this information in case the Edge Transport server or AD LDS is compromised. Proxy addresses are referenced when the Edge Transport server performs the recipient lookup antispam task.
- **Safe Senders List, Blocked Senders List, and Safe Recipients List** Safe Senders Lists, Blocked Senders Lists and Safe Recipients Lists defined in each recipient's Outlook instance are aggregated and replicated to AD LDS. These settings are stored in the mailbox database where the recipient's mailbox resides. An Outlook user's safelist collection is the combined data from the user's Safe Senders List, Safe Recipients List, Blocked Senders List, and external contacts. Having safelist collection data available in AD LDS enables the Edge Transport server to screen senders appropriately, reducing operational overhead for filtering mail. This information is sent as hashed data.

Important:
Although the safe recipient data is stored in Outlook and can be aggregated into the safelist collection on the AD LDS instance on the Edge Transport server, the content filtering functionality doesn't act on safe recipient data.

- **Per recipient anti spam settings** You can use the **Set-Mailbox** cmdlet to assign anti spam threshold settings per recipient that differ from the organization-wide anti spam settings. If you configure per recipient anti spam settings, these settings override organization-wide settings. By replicating these settings to AD LDS, the per recipient settings can be considered before the message is relayed to the Exchange organization. This information is sent as hashed data.

Return to top

Topology information

The topology information includes notification of newly subscribed Edge Transport servers or removed Edge Subscriptions. This data is refreshed every five minutes.

Return to top

Edge Subscription credentials

Topic Last Modified: *2014-02-21*

This topic explains how the Edge Subscription process provisions credentials used to help secure the EdgeSync synchronization process and how EdgeSync uses those credentials to establish a secure LDAP connection between an Exchange 2013 Mailbox server and an Edge Transport server. To learn more about the Edge Subscription process, see Edge Subscriptions.

Contents

Edge Subscription process

The Edge Transport server is subscribed to an Active Directory site to establish a synchronization relationship between the Mailbox servers in an Active Directory site and the subscribed Edge Transport server. The credentials provisioned during the Edge Subscription process are used to help secure the LDAP connection between a Mailbox server and an Edge Transport server in the perimeter network.

When you run the **New-EdgeSubscription** cmdlet on an Edge Transport server, EdgeSync bootstrap replication account (ESBRA) credentials are created in the Active Directory Lightweight Directory Services (AD LDS) directory on the local server and then written to the Edge Subscription file. These credentials are used only to establish initial synchronization and will expire 24 hours after the Edge Subscription file is created. If the Edge Subscription process isn't completed within 24 hours, you will need to run the **New-EdgeSubscription** cmdlet again to create a new Edge Subscription file. The Edge Subscription XML file stores configuration data for the Edge Subscription.

The Edge Subscription XML file contains the data shown in the following table.

Edge Subscription file contents

Subscription data	Description
EdgeServerName	The NetBIOS name of the Edge Transport server. The Active Directory name of the Edge Subscription will match this name.
EdgeServerFQDN	The fully qualified domain name (FQDN) of the Edge Transport server. Mailbox servers in the subscribed Active Directory site must be able to locate the Edge Transport server by using DNS

	to resolve the FQDN.
EdgeCertificateBlob	The public key of the Edge Transport server's self-signed certificate.
ESRAUsername	The name assigned to the ESBRA. The ESBRA account has the following format: ESRA.*Edge Transport server name*. ESRA means EdgeSync replication account; note the difference between ESBRA (initial bootstrap replication account) and ESRA.
ESRAPassword	The password assigned to the ESBRA. The password is generated using a random number generator and is stored in the Edge Subscription file in clear text.
EffectiveDate	The creation date of the Edge Subscription file.
Duration	The length of time these credentials will be valid before they expire. The ESBRA account is valid for only 24 hours.
AdamSslPort	The secure LDAP port EdgeSync binds to when synchronizing data from Active Directory to AD LDS. By default, this is TCP port 50636.
ProductID	The licensing information for the Edge Transport server. You need to license the Edge Transport server before creating the Edge Subscription.
VersionNumber	The version number of the Edge Subscription file.
SerialNumber	The Exchange version of the Edge Transport server.

Important:
ESBRA credentials are written to the Edge Subscription file in clear text. You need to protect this file throughout the subscription process. After the Edge Subscription file is imported to your Exchange organization, you should immediately delete the Edge Subscription file from the Edge Transport server, from the network share you used to import the file to your Exchange organization, and from any removable media.

Return to top

EdgeSync replication accounts

EdgeSync replication accounts (ESRA) are an important part of EdgeSync security. Authentication and authorization of the ESRA is the mechanism used to help secure the connection between an Edge Transport server and a Mailbox server.

The ESBRA contained in the Edge Subscription file is used to establish a secure LDAP connection during initial synchronization. After the Edge Subscription file is imported to a Mailbox server in the Active Directory site where the Edge Transport server is being subscribed, additional ESRA accounts are created in Active Directory for each Edge Transport-Mailbox server pair. During initial synchronization, the newly created ESRA credentials are replicated to AD LDS. These ESRA credentials are used to help secure later synchronization sessions.

Each EdgeSync replication account is assigned the properties shown in the following table.

Ms-Exch-EdgeSyncCredential properties

Property name	Type	Description
TargetServerFQDN	String	The Edge Transport server accepting these credentials.
SourceServerFQDN	String	The Mailbox server presenting these credentials. This value is empty if the credential is the bootstrap credential.
EffectiveTime	DateTime (UTC)	When to start using this credential.
ExpirationTime	DateTime (UTC)	When to stop using this credential.
UserName	String	The user name used to authenticate.

Password	Byte	The password used to authenticate. The password is encrypted using **ms-Exch-EdgeSync-Certificate**.

The following sections describe how the ESRA credentials are provisioned and used during the EdgeSync synchronization process.

Provision the EdgeSync bootstrap replication account

When the **New-EdgeSubscription** cmdlet is run on the Edge Transport server, the ESBRA is provisioned as follows:

- A self-signed certificate (Edge-Cert) is created on the Edge Transport server. The private key is stored in the local computer store and the public key is written to the Edge Subscription file.
- The ESBRA account is created in AD LDS, and its credentials are written to the Edge Subscription file.
- The Edge Subscription file is exported by copying it to removable media (because the Edge Server is not in your Active Directory, you cannot use a shared folder for exporting the file). The file is now ready to import to a Mailbox server.

Provision EdgeSync replication accounts in Active Directory

When the Edge Subscription file is imported on a Mailbox server, the following steps occur to establish a record of the Edge Subscription in Active Directory and to provision additional ESRA credentials:

1. An Edge Transport server configuration object is created in Active Directory. The Edge-Cert certificate is written to this object as an attribute.
2. Every Mailbox server in the subscribed Active Directory site receives an Active Directory notification that a new Edge Subscription has been registered. As soon as the notification is received, each Mailbox server retrieves the ESRA.edge account and encrypts the account by using the Edge-Cert public key. The encrypted ESRA.edge account is written to the Edge Transport server configuration object.
3. Each Mailbox server creates a self-signed certificate (TransportService-Cert). The private key is stored in the local computer store and the public key is stored in the Mailbox server configuration object in Active Directory.
4. Each Mailbox server encrypts the ESRA.edge account by using the public key of its own TransportService certificate and then stores it in its own configuration object.
5. Each Mailbox server generates an ESRA for each existing Edge Transport server configuration object in Active Directory (ESRA.edge. *Mailboxname.*#).

Example: ESRA.edge.Example.0

The password for ESRA.edge is generated by a random number generator and is encrypted by using the public key of the TransportService-Cert certificate. The generated password has the maximum length allowed for Windows Server.

6. Each ESRA.edge. *Mailboxname.#* account is encrypted by using the public key of the Edge-Cert certificate and is stored on the Edge Transport server configuration object in Active Directory.

The following sections explain how these accounts are used during EdgeSync synchronization.

Return to top

Authenticate initial replication

The initial ESBRA account is used only when establishing initial synchronization. During the first EdgeSync synchronization, the additional ESRA accounts, ESRA.edge.*Mailboxname.#*, are replicated to AD LDS. These accounts are used to authenticate later EdgeSync synchronization sessions.

The Mailbox server that performs the initial replication is determined randomly. The first Mailbox server in the Active Directory site to perform a topology scan and discover the new Edge Subscription performs the initial replication. Because this discovery is based on the timing of the topology scan, any Mailbox server in the site may perform the initial replication.

EdgeSync initiates a secure LDAP session from the Mailbox server to the Edge Transport server. The Edge Transport server presents its self-signed certificate and the Mailbox server verifies that the certificate matches the certificate stored on the Edge Transport server configuration object in Active Directory. After the Edge Transport server's identity is verified, the Mailbox server provides the credentials of the ESRA.edge.*Mailboxname.#* account to the Edge Transport server. The Edge Transport server verifies the credentials against the account stored in AD LDS.

The EdgeSync service on the Mailbox server then pushes the topology, configuration, and recipient data from Active Directory to AD LDS. The change to the Edge Transport server configuration object in Active Directory is replicated to AD LDS. AD LDS receives the newly added ESRA.edge.*Mailboxname.#* entries and the Microsoft Exchange Credential Service creates the corresponding AD LDS account. These accounts are now available to authenticate later scheduled EdgeSync synchronization sessions.

Microsoft Exchange Credential Service

The Microsoft Exchange Credential Service is part of the Edge Subscription process. The Credential Service runs only on the Edge Transport server. This service creates the reciprocal ESRA accounts in AD LDS so a Mailbox server can authenticate to an Edge Transport server to perform EdgeSync synchronization. EdgeSync doesn't communicate directly with the Microsoft Exchange Credential Service. The Microsoft Exchange Credential Service communicates with AD LDS and installs the ESRA credentials whenever the Mailbox server updates them.

Return to top

Authenticate scheduled synchronization sessions

After initial EdgeSync synchronization finishes, the EdgeSync synchronization schedule is established and any Active Directory data that has changed is regularly updated in AD LDS. A Mailbox server initiates a secure LDAP session with the AD LDS instance on the Edge Transport server. AD LDS proves its identity to that Mailbox server by presenting its self-signed certificate. The Mailbox server presents its ESRA.edge credentials to AD LDS. The ESRA.edge password is encrypted using the Mailbox server's self-signed certificate's public key. Only that particular Mailbox server can use those credentials to authenticate to AD LDS.

Return to top

Renew EdgeSync replication accounts

The password for the ESRA account must comply with the local server's password policy. To prevent the password renewal process from causing temporary authentication failure, a second ESRA.edge account is created seven days before the first ESRA.edge account expires, with an effective time three days before the first ESRA expiration time. As soon as the second ESRA.edge account becomes effective, EdgeSync stops using the first account and starts to use the second account. When the expiration time for the first account is reached, those ESRA credentials are deleted. This renewal process will continue until the Edge Subscription is removed.

Return to top

Manage Edge Subscriptions

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-21*

This topic gives detailed information on a variety of Edge Subscription management tasks.

Contents

Verify EdgeSync results

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "EdgeSync" entry and the "Edge Transport servers" section in the Mail flow permissions topic.
- You need to have an Edge server subscribed to your Internet-facing Active Directory site. For more information, see Configure Internet mail flow through a subscribed Edge Transport server.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Subscribe an Edge Transport server

You can subscribe one or more Edge Transport servers to a single Active Directory site. If you deploy additional Edge Transport servers in your perimeter network and subscribe them to the same Active Directory site where an Edge Subscription already exists, the following actions occur:

- A new Edge Subscription object is created in Active Directory.
- Additional ESRA accounts are created for each Mailbox server in the Active Directory site. These accounts are replicated to Active Directory Lightweight Directory Services (AD LDS) and used by the EdgeSync synchronization process during synchronization with the new server.
- The new Edge Subscription is added to the source server list of the automatic Send connector to the Internet. Messages submitted to that connector for processing will be load balanced between the subscribed Edge Transport servers.
- An inbound Send connector is automatically created from the Edge Transport server to the Exchange organization.
- EdgeSync synchronization to the Edge Transport server starts.

Remove an Edge Subscription

You may occasionally want to remove an Edge Subscription from the Exchange organization or from both the Exchange organization and the Edge Transport server. If you plan to later resubscribe the Edge Transport server to the Exchange organization, don't remove the Edge Subscription from the Edge Transport server. When you remove the Edge Subscription from an Edge Transport server, all replicated data is deleted from AD LDS. This can take a long time if you have lots of recipient data.

To completely remove an Edge Subscription, you need to run this procedure on the Edge Transport

server you wish to remove and on an Exchange 2013 Mailbox server in the Active Directory site where the Edge Transport server is subscribed.

After you remove the Edge Subscription, synchronization of information from AD LDS stops. All accounts stored in AD LDS are removed, and the Edge Transport server is removed from the source server list of any Send connector. You will no longer be able to use Edge Transport server features that rely on Active Directory data.

1. To remove the Edge Subscription from the Edge Transport server, use the following syntax.

```
Remove-EdgeSubscription <EdgeTransportServerIdentity>
```

For example, to remove the Edge Subscription on the Edge Transport server named Edge01, run the following command.

```
Remove-EdgeSubscription Edge01
```

2. To remove the Edge Subscription from the Mailbox server, use the following syntax.

```
Remove-EdgeSubscription <MailboxServerIdentity>
```

For example, to remove the Edge Subscription on the Mailbox server named Mailbox01, run the following command.

```
Remove-EdgeSubscription Mailbox01
```

You will need to remove the Edge Subscription if:

- You no longer want the Edge Transport server to participate in EdgeSync synchronization. You will need to remove the Edge Subscription from both the Edge Transport server and from the Exchange organization.
- An Edge Transport server is being decommissioned. In this scenario, you only need to remove the Edge Subscription from the Exchange organization. If you uninstall the Edge Transport server role from the computer, the AD LDS instance and all Active Directory data stored in AD LDS will also be removed.
- You want to change the Active Directory site association for the Edge Subscription. You will only need to remove the Edge Subscription from the Exchange organization. After the Edge Subscription is removed from the Exchange organization, you can resubscribe the Edge Transport server to a different Active Directory site.

When you remove an Edge Subscription from the Exchange organization:

- Synchronization of information from Active Directory to AD LDS stops.
- The ESRA accounts are removed from both Active Directory and AD LDS.
- The Edge Transport server is removed from the *SourceTransportServers* property of any Send connector.
- The automatic inbound Send connector from the Edge Transport server to the Exchange organization is removed from AD LDS.

When you remove the Edge Subscription from an Edge Transport server:

- You can no longer use Edge Transport server features that rely on Active Directory data.
- Replicated data is removed from AD LDS.
- Tasks that were disabled when the Edge Subscription was created are re-enabled to allow for local configuration.

Resubscribe an Edge Transport server

Occasionally you may have to resubscribe an Edge Transport server to an Active Directory site. When the Edge Subscription is re-created, new credentials are generated and you need to follow the complete Edge Subscription process. You will need to resubscribe an Edge Transport server if:

- You add new Mailbox servers in the subscribed Active Directory site, and you want the new Mailbox server to participate in EdgeSync synchronization.
- You applied the license key for the Edge Transport server after creating the Edge Subscription. Licensing information for the Edge Transport server is captured when the Edge Subscription is created. Subscribed Edge Transport servers only appear as licensed if they are subscribed to the Exchange organization after the license key has already been applied on the Edge Transport server. If the license key is applied on the Edge Transport server after you perform the Edge Subscription process, the licensing information won't be updated in the Exchange organization, and you will need to resubscribe the Edge Transport server.
- The ESRA credentials are compromised.

Important:
To resubscribe an Edge Transport server, export a new Edge Subscription file on the Edge Transport server and then import the XML file on a Mailbox server. You will need to resubscribe the Edge Transport server to the same Active Directory site where it was originally subscribed. You don't need to first remove the original Edge Subscription; the resubscription process will overwrite the existing Edge Subscription.

Add or Remove a Mailbox server

If you add a Mailbox server to an Active Directory site that already has an Edge Transport server subscribed, the new Mailbox server doesn't automatically participate in EdgeSync synchronization. To enable a newly deployed Mailbox server to participate in EdgeSync synchronization, you need to resubscribe each Edge Transport server to the Active Directory site.

Removing a Mailbox server from an Active Directory site where an Edge Transport server is subscribed won't affect EdgeSync synchronization unless that Mailbox server is the only Mailbox server in that site. If you remove all Mailbox servers from the Active Directory site where an Edge Transport server is subscribed, that site's subscribed Edge Transport servers are orphaned.

Run EdgeSync manually

You may want to manually run EdgeSync if you've made significant changes to the configuration or recipients in Active Directory and want your changes synchronized immediately. You can run a full

synchronization, or only synchronize changes made since the last replication.

A manual EdgeSync resets the EdgeSync synchronization schedule. The next automatic synchronization is based on when you ran the manual synchronization.

To manually run EdgeSync, use the following syntax.

```
Start-EdgeSynchronization [-Server <MailboxServerIdentity>]
[-TargetServer <EdgeTransportServerIdentity> [-
ForceFullSync]
```

The following example starts EdgeSync with the following options:

- The synchronization is initiated from the Exchange 2013 Mailbox server named Mailbox01.
- All Edge Transport servers are synchronized.
- Only the changes since the last replication are synchronized.

```
Start-EdgeSynchronization -Server Mailbox01
```

This example starts EdgeSync with the following options:

- The synchronization is initiated from the local Mailbox server.
- Only the Edge Transport server named Edge03 is synchronized.
- All recipient and configuration data are fully synchronized.

```
Start-EdgeSynchronization -TargetServer Edge03 -
ForceFullSync
```

Verify EdgeSync results

You can use the **Test-EdgeSynchronization** cmdlet to verify that the Edge synchronization is working. This cmdlet reports synchronization status of subscribed Edge Transport servers.

The output of this cmdlet lets you view objects that have not been synchronized to the Edge Transport server. The task compares data stored in Active Directory against data stored in AD LDS and reports any data inconsistencies.

You can use the *ExcludeRecipientTest* parameter on the **Test-EdgeSynchronization** cmdlet to exclude validation of recipient data synchronization. If you include this parameter, only the synchronization of configuration objects is validated. Validating recipient data will take longer than validating only configuration data.

Verify EdgeSync results for a single recipient

To verify EdgeSync results for a single recipient, use the following syntax on a Mailbox server in the subscribed Active Directory site.

```
Test-EdgeSynchronization -VerifyRecipient <emailaddress>
```

This example verifies EdgeSync results for the user kate@contoso.com.

```
Test-EdgeSynchronization -VerifyRecipient kate@contoso.com
```

Return to top

Modify AD LDS configuration

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-22*

You can use the **ConfigureAdam.ps1** script (located in $env:ExchangeInstallPath\Scripts) to modify the default Active Directory Lightweight Directory Services (AD LDS) configuration on Edge Transport servers before you subscribe the Edge Transport server to your Exchange organization.

Important:
The **ConfigureAdam.ps1** script invokes the **dsdbutil** command to change the registry settings for AD LDS. The **dsdbutil** command is an AD LDS management tool intended for use only by experienced administrators; using **ConfigureAdam.ps1** is the recommended way of changing the AD LDS configuration.

The parameters in the following table are available for the **ConfigureAdam.ps1** script. You can use one, all, or any combination of these parameters to modify AD LDS.

Parameters available for the ConfigureAdam.ps1 script

Parameter	Description
Ldapport	Modifies the port used for LDAP communication. By default, the Edge Transport server uses the nonstandard port 50389.
Sslport	Modifies the port used for secure LDAP communication. By default, the Edge Transport server uses the nonstandard port 50636.
LogPath	Modifies the log file location. By default, the Edge Transport server creates log files in the path %ExchangeInstallPath%Transport Roles

	\Data\adam
DataPath	Modifies the location of the directory database file. By default, the Edge Transport server stores the directory database in the path %ExchangeInstallPath%Transport Roles\Data\adam

What do you need to know before you begin?

- Estimated time to complete: five minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Edge Transport servers" section in Mail flow permissions.
- If you need to make any modifications to the Edge Transport server's AD LDS configuration, do so before subscribing the Edge Transport server to your Exchange organization. If you modify the AD LDS configuration of a subscribed Edge Transport server, you will then need to resubscribe the Edge Transport server to the Exchange organization.
- Always use the script to modify the registry settings. Making manual registry changes to the AD LDS configuration might make the AD LDS instance unavailable.
- If you need to modify the LDAP port or the SSL port used by AD LDS, first verify that the selected port isn't being used by another application. You can use the **netstat** command to view the ports being used on the Edge Transport server.
- You can only use the Shell to perform this procedure.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Modify the AD LDS configuration on an Edge Transport server

This example changes the LDAP port used by AD LDS to 5000. The ampersand (&) is part of the command syntax.

```
& $env:ExchangeInstallPath\Scripts\ConfigureAdam.ps1 -LdapPort:5000
```

This example makes the following changes to the AD LDS configuration. The ampersand (&) is part of the command syntax. Note the colon (:) used between each parameter and its value:

- Changes the LDAP port to 5000

- Changes the SSL port to 500
- Changes the log path to D:\Exchange Server\Data\ADLDS
- Changes the data path to D:\Exchange Server\Data\ADLDS

```
& $env:ExchangeInstallPath\Scripts\ConfigureAdam.ps1 -
LdapPort:5000 -SslPort:5001 -LogPath:"D:\Exchange Server
\Data\ADLDS" -DataPath:"D:\Exchange Server\Data\ADLDS"
```

Manually configure Edge Transport server mail flow

Exchange Server 2013 > Edge Transport servers > Edge Subscriptions >

Topic Last Modified: *2014-02-21*

This topic describes procedures for making manual configuration changes to how an Edge Transport Server manages mail flow. These procedures are intended to address specific scenarios; unless your organization has specific needs for making manual configuration changes, using the default configuration when subscribing Edge Transport servers is preferred.

Contents

Manually configure Send connectors

You can manually modify a Send connector's configuration. For example, if you need to route outbound email through a smart host, you can suppress automatic creation of a Send connector and manually configure a Send connector to the Internet.

Intra-Organization Send connectors

The Intra-Organization Send connector is an implicit and hidden Send connector that's

automatically computed by Exchange and enables the Transport service on Mailbox servers within the same organization to relay messages to each other without using explicit Send connectors. Because a configuration object with an Active Directory site association exists in Active Directory for an Edge Subscription, the intra-organization Send connector will also be used to relay messages to that Edge Transport server.

Only Mailbox servers located in the subscribed Active Directory site can transfer email directly to or from the subscribed Edge Transport server. If you have a multi-site Active Directory forest and Exchange is deployed in more than one site, the Mailbox servers in non-subscribed sites will route outbound email to the subscribed site. A Mailbox server in the subscribed site will route outbound email to the Edge Transport server.

Create additional Send connectors after Edge subscription

After an Edge Transport server is subscribed to an Active Directory site, cmdlets for creating and modifying Send connectors on the Edge Transport server are disabled. If you want to create a Send connector whose source server is the Edge Transport server, you can create the Send connector inside the Exchange organization. You can specify one or more Edge Subscriptions as the source server for a Send connector. You can't specify both Mailbox servers and Edge Subscriptions as source servers for the same Send connector. The Send connector will be replicated to the AD LDS instance on the Edge Transport server that's configured as a source server the next time configuration data is synchronized by EdgeSync. If you list more than one Edge Subscription as a source server, connections to that Send connector will be load balanced between the subscribed Edge Transport servers. Edge Transport servers need to be subscribed to the same Active Directory site for load balancing to occur. If Edge Subscriptions in different Active Directory sites are configured as source servers on the same Send connector, Edge Transport servers will route only to the closest source server.

You will need to manually create Send connectors if:

- You suppressed automatic creation of the Internet or inbound Send connectors.
- You have accepted domains in your organization that are configured as external relay domains.

Reasons to suppress automatic creation of Send connectors

Depending on the topology of your Exchange organization, you may decide to suppress automatic creation of Send connectors. The following examples describe scenarios that require you to suppress automatic creation of Send connectors.

Partition mail flow

If you decide to partition the inbound and outbound mail processing between two Edge Transport

servers, one Edge Transport server is responsible for processing outbound mail flow and a second Edge Transport server is responsible for processing inbound mail flow. To do this, configure the Edge Subscriptions as follows:

- For the outbound Edge Transport server, run the following command on the Mailbox server.

```
New-EdgeSubscription -FileData ([byte[]]$(Get-Content -Path
"C:\EdgeServerSubscription.xml" -Encoding Byte -ReadCount
0)) -Site "Site-A" -CreateInboundSendConnector $false -
CreateInternetSendConnector $true
```

- For the inbound Edge Transport server, run the following command on the Mailbox server.

```
New-EdgeSubscription -FileData ([byte[]]$(Get-Content -Path
"C:\EdgeServerSubscription.xml" -Encoding Byte -ReadCount
0)) -Site "Site-A" -CreateInboundSendConnector $true -
CreateInternetSendConnector $false
```

Route outbound email to a smart host

If your Exchange organization routes all outbound email through a smart host, the automatically created Send connector won't have the correct configuration.

Run the following command on the Mailbox server to suppress automatic creation of the Send connector to the Internet.

```
New-EdgeSubscription -FileData ([byte[]]$(Get-Content -Path
"C:\EdgeServerSubscription.xml" -Encoding Byte -ReadCount
0)) -Site "Site-A" -CreateInternetSendConnector $false
```

After the Edge Subscription process is complete, manually create a Send connector to the Internet. Create the Send connector inside the Exchange organization, and select the Edge Subscription as the source server for the connector. Select the `custom` usage type and configure one or more smart hosts. This new Send connector will be replicated to the AD LDS instance on the Edge Transport server the next time EdgeSync synchronizes configuration data. You can force immediate EdgeSync synchronization by running the **Start-EdgeSynchronization** cmdlet on a Mailbox server.

Example: Using the Shell to configure a Send connector for a subscribed Edge Transport server to route messages for all Internet address spaces through a smart host. Run this task on a Mailbox server inside the Exchange organization, not on the Edge Transport server.

```
New-SendConnector -Name "EdgeSync - Site-A to Internet" -
Usage Custom -AddressSpaces SMTP:*;100 -DNSRoutingEnabled
$false -SmartHosts 192.168.10.1 -SmartHostAuthMechanism
None -SourceTransportServers EdgeSubscriptionName
```

Important:
This example doesn't specify any smart host authentication mechanism. Make sure you configure the correct authentication mechanism and provide all necessary credentials when you create a smart host connector in your own Exchange organization.

Configure Send connectors for external relay domains

If you have accepted domains in your Exchange organization that are configured as external relay domains, you need to manually create a Send connector for those address spaces. Messages being delivered to external relay domains are relayed by the Edge Transport server. The Edge Subscription process doesn't automatically create and configure Send connectors for external relay domains. Therefore, you need to configure Send connectors for those domains and specify one or more Edge Subscriptions as the source server for those Send connectors.

The DNS MX resource record for an external relay domain resolves to your Edge Transport server. You can configure a Send connector that relays email to an external relay domain to use a smart host for routing. Configuring the Send connector for an external relay domain to use DNS routing will create a routing loop. For more information about external relay domains, see Accepted domains.

Return to Top

Configure Internet mail flow through a subscribed Edge Transport server

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-22*

To establish Internet mail through an Edge Transport server, subscribe the Edge Transport server to an Active Directory site. This automatically creates the two Send connectors required for Internet mail flow:

- A Send connector configured to send outbound email to all Internet domains.
- A Send connector configured to send inbound email from the Edge Transport server to an Exchange 2013 Mailbox server.

If you don't want to subscribe the Edge Transport server to an Active Directory site, it's possible to manually create the Send connectors required to establish mail flow between the Mailbox server and the Edge Transport server. For more information, see Configure Internet mail flow through an Edge Transport server without using EdgeSync. However, we recommend you subscribe the Edge Transport server to the Active Directory site whenever possible.

Before you begin

- Estimated time to complete: 20 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "EdgeSync" entry and the "Edge Transport servers" in the Mail flow permissions topic.
- Before you subscribe an Edge Transport server to your organization, you will first need to configure authoritative domains and email address policies for your Exchange organization.
- Enable the secure LDAP port 50636/TCP through the firewall separating your perimeter network from the Exchange organization. The Edge Transport server needs to be able to communicate with all Exchange 2013 Mailbox servers in the subscribed Active Directory site.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Configure Internet mail flow through a subscribed Edge Transport server

1. On the Edge Transport server, create the Edge Subscription file using the following syntax.

```
New-EdgeSubscription -FileName <FileName>.xml [-Force]
```

The following example creates an Edge Subscription file named EdgeSubscriptionInfo.xml in the folder C:\My Documents. The *Force* parameter suppresses prompts confirming commands that will be disabled and warnings that configuration data will be overwritten on the Edge Transport server.

```
New-EdgeSubscription -FileName "C:\My Documents
\EdgeSubscriptionInfo.xml" -Force
```

2. Copy the resulting Edge Subscription file to a Mailbox server in the Active Directory site you're subscribing the Edge Transport server to.
3. On the Mailbox server, to import the Edge Subscription file, use the following syntax.

```
New-EdgeSubscription -FileData ([byte[]]$(Get-Content -Path
"<FileName>.xml" -Encoding Byte -ReadCount 0)) -Site
<SiteName>
```

This example imports the Edge Subscription file named EdgeSubscriptionInfo.xml from the folder D:\Data, and subscribes the Edge Transport server to the Active Directory site named "Default-First-Site-Name".

```
New-EdgeSubscription -FileData ([byte[]]$(Get-Content -Path
"D:\Data\EdgeSubscriptionInfo.xml" -Encoding Byte -
ReadCount 0)) -Site "Default-First-Site-Name"
```

Note:

You can use the *CreateInternetSendConnector* or *CreateInboundSendConnector* parameters to prevent one or both of the required Send connectors from being automatically created. For more information, see Edge Subscriptions.

4. On the Mailbox server, run the following command to start the first EdgeSync synchronization.

```
Start-EdgeSynchronization
```

5. When you're finished, we strongly recommend you delete the Edge Subscription file from both the Edge Transport server and from the Mailbox server. The Edge Subscription file contains information about credentials used during the LDAP communication process.

Configure Internet mail flow through an Edge Transport server without using EdgeSync

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-22*

We recommend you use the Edge Subscription process to establish mail flow between your Exchange organization and an Edge Transport server. However, certain situations may prevent you from subscribing the Edge Transport server to your Exchange organization using the Edge Subscription process. To manually establish mail flow between your Exchange organization and an Edge Transport server, you must create and configure the Send connectors and Receive connectors on the Edge Transport server and on the Mailbox servers in your Exchange organization.

Before you begin

- Estimated time to complete this task: 30 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Send connectors" entry, the "Send connectors - Edge Transport" entry and the "Receive connectors - Edge Transport" entry in the Mail flow permissions topic.

- This procedure uses Basic authentication over Transport Layer Security (TLS) to provide encryption and authentication. When you use Basic authentication over TLS, the receiving server must have an X.509 Secure Sockets Layer (SSL) server certificate installed. The fully qualified domain name (FQDN) value configured on the Receive connector must match the FQDN in the SSL server certificate. By default, the value of the FQDN on the Receive connector is the FQDN of the server that contains the Receive connector.
- You can also use the Externally Secured authentication method. However, if you do so, the communication between the Edge Transport server and Mailbox server isn't authenticated or encrypted by Exchange. We recommend you use the Externally Secured authentication method only when an additional encryption method is also used. The encryption method can be an Internet Protocol security (IPsec) association or a virtual private network (VPN).
- An Edge Transport server is typically *multihomed*. This means that the Edge Transport server has network adapters connected to multiple network segments. Each of these network adapters has a unique IP configuration. The network adapter that's connected to the external, or public, network segment should be configured to use a public Domain Name System (DNS) server for name resolution. This enables the server to resolve SMTP domain names to MX resource records and route mail to the Internet. The network adapter that's connected to the internal, or private, network segment should be configured to use a DNS server in the perimeter network or should have a Hosts file available.
- You need to create a user account in Active Directory and add the account to the universal security group on the Exchange Server computer. This account is used by the Send connector on the Edge Transport server to authenticate to the destination Mailbox server in the Exchange organization.

Important:

This account is granted the permissions associated with the computers running Exchange Server. Make sure you safeguard the account credentials to prevent misuse of the account. You can configure the account to allow logon to specific computers only.

Tip:

Edge Transport Server Procedures

The following connectors are required on the Edge Transport server:

- A Send connector configured to send messages to the Internet
- A Send connector configured to send messages to the Mailbox servers in the Exchange organization
- A Receive connector configured to receive messages only from Mailbox servers in the Exchange organization
- A Receive connector configured to accept messages only from the Internet

By default, a single Receive connector is created during the installation of the Edge Transport server role. This connector can be used for both incoming Internet messages and incoming messages from

the Mailbox servers. Typically, the Edge Subscription process automatically configures the correct permissions and authentication on the default Receive connector. When you don't use the Edge Subscription process, we recommend you modify the default Receive connector on the Edge Transport server to only accept messages from the Internet. You should then create a Receive connector on the Edge Transport server that's configured to only accept messages from internal Mailbox servers.

The following sections walk you through all the configuration steps required to prepare your Edge Transport server to communicate with your Exchange organization.

Note:
You can only use the Shell to perform these procedures on Edge Transport servers.

Step 1: Create a Send connector configured to send messages to the Internet

This Send connector requires the following configuration:

- **Name** To Internet (or any descriptive name)
- **Usage type** Internet
- **Address spaces** "*" (all domains)
- **Network settings** Use DNS MX records to route mail automatically. Depending on your network configuration, you can also route mail through a smart host. The smart host then routes mail to the Internet.

To create a Send connector that's configured to send messages to the Internet, run the following command.

```
New-SendConnector -Name "To Internet" -AddressSpaces * -
Usage Internet -DNSRoutingEnabled $true
```

For detailed syntax and parameter information, see New-SendConnector.

Step 2: Create a Send connector configured to send messages to the Exchange organization

Use the **New-SendConnector** cmdlet to create a Send connector.

Note:
Before you create the Send connector, you first need to run the **Get-Credential** command to save the user name and password you will use in a temporary variable. You need to do this because the **New-SendConnector** cmdlet will not accept user credentials in plain text.

This Send connector requires the following configuration:

- **Name** To Internal Org (or any descriptive name)
- **Usage type** Internal
- **Address spaces** All accepted domains for the Exchange organization. For example, *.contoso.com.
- DNS routing disabled (smart host routing enabled)
- **Smart hosts** FQDN of one or more Mailbox servers as smart hosts. For example, mbxserver01.contoso.com and mbxserver02.contoso.com.
- **Smart host authentication methods** Basic authentication and Basic authentication over TLS
- **Smart host authentication credentials** Credentials for the user account in the internal domain. You first need to save the user name and password in a temporary variable, because the **New-SendConnector** cmdlet will not accept user credentials in plain text.

To create a Send connector configured to send messages to the Exchange organization, run the following commands.

```
$MailboxCredentials = Get-Credential
New-SendConnector -Name "To Internal Org" -Usage Internal -
AddressSpaces *.contoso.com -DNSRoutingEnabled $false -
SmartHosts mbxserver01.contoso.com,mbxserver02.contoso.com
-SmartHostAuthMechanism BasicAuth,BasicAuthRequireTLS -
AuthenticationCredential $MailboxCredentials
```

For detailed syntax and parameter information, see New-SendConnector.

Step 3: Modify the default Receive connector to only accept messages from the Internet

You should make the following configuration changes to the default Receive connector:

- Modify the name to reflect that the connector will be used solely to receive email from the Internet. The name of the default Receive connector is "Default internal Receive connector <Edge Transport server name>".
- Change the network bindings to accept messages only from the network adapter that is accessible from the Internet. For example, 10.1.1.1 and the standard SMTP TCP port value of 25.

To modify the default Receive connector to only accept messages from the Internet, run the following command.

```
Set-ReceiveConnector "Default internal Receive connector
Edge01" -Name "From Internet" -Bindings 10.1.1.1:25
```

For detailed syntax and parameter information, see Set-ReceiveConnector.

Step 4: Create a Receive connector configured to only accept messages from the Exchange organization

This Receive connector requires the following configuration:

- **Name** From Internal Org (or any descriptive name)
- **Usage type** Internal
- **Local network bindings** Internal network-facing network adapter. For example, 10.1.1.2 and the standard SMTP TCP port value of 25.
- **Remote network settings** IP address of one or more Mailbox servers in the Exchange organization. For example, 192.168.5.10 and 192.168.5.20.
- **Authentication methods** TLS, Basic authentication, Basic authentication over TLS, and Exchange Server authentication.

To create a Receive connector configured to only accept messages from the Exchange organization, run the following command.

```
New-ReceiveConnector -Name "From Internal Org" -Usage
Internal -AuthMechanism
TLS,BasicAuth,BasicAuthRequireTLS,ExchangeServer -Bindings
10.1.1.2:25 -RemoteIPRanges 192.168.5.10,192.168.5.20
```

For detailed syntax and parameter information, see New-ReceiveConnector.

How do you know these steps worked?

To verify that you have successfully configured the required Send connectors and Receive connectors, run the following commands on the Edge Transport server and verify the values displayed are the values you configured.

```
Get-SendConnector | Format-List
Name,Usage,AddressSpaces,SourceTransportServers,DSNRoutingE
nabled,SmartHosts,SmartHostAuthMechanism
Get-ReceiveConnector | Format-List
Name,Usage,AuthMechanism,Bindings,RemoteIPRanges
```

Mailbox server procedures

Mailbox servers in your organization require a Send connector configured to send messages to the Edge Transport server for relay to the Internet.

By default, two Receive connectors are created during the installation of the Mailbox server role.

The connector named Client *ServerName* is configured to accept messages from all POP3 and IMAP messaging clients. The connector named Default *ServerName* is configured to accept messages from an Edge Transport server. No modifications to these connectors are required.

Step 5: Create a Send connector configured to send outgoing messages to the Edge Transport server

This Send connector requires the following configuration:

- **Name** To Edge (or any descriptive name)
- **Usage type** Internal
- **Address spaces** "*" (all domains)
- DNS routing disabled (smart host routing enabled)
- **Smart hosts** IP address or FQDN of the Edge Transport server. For example, edge01.contoso.net.
- **Source Mailbox servers** FQDN of one or more Mailbox servers. For example, mbxserver01.contoso.com and mbxserver02.contoso.com.
- **Smart host authentication methods** Basic authentication and basic authentication over TLS.
- **Smart host authentication credentials** Credentials for the user account on the Edge Transport server. You first need to save the user name and password in a temporary variable, because the **New-SendConnector** cmdlet will not accept user credentials in plain text.

To create a Send connector configured to send outgoing messages to the Edge Transport server, run the following commands.

```
$EdgeCredentials = Get-Credential
New-SendConnector -Name "To Edge" -Usage Internal -
AddressSpaces * -DNSRoutingEnabled $false -SmartHosts
edge01.contoso.com -SourceTransportServers
mbxserver01.contoso.com,mbxserver02.contoso.com -
SmartHostAuthMechanism BasicAuth,BasicAuthRequireTLS -
AuthenticationCredential $EdgeCredentials
```

For detailed syntax and parameter information, see New-SendConnector.

How do you know this step worked?

To verify that you have successfully created a Send connector configured to send outgoing messages to the Edge Transport server, run the following command on a Mailbox server and verify the values displayed are the values you configured.

```
Get-SendConnector | Format-List
Name,Usage,AddressSpaces,DSNRoutingEnabled,SmartHosts,Sourc
eTransportServers,SmartHostAuthMechanism
```

Edge Transport server planning

Topic Last Modified: *2014-04-28*

The Edge Transport server role has been re-introduced in Exchange Service Pack 1. Edge Transport provides improved anti-spam protection for the Exchange organization. The Edge Transport server also applies policies to messages in transport between organizations. This server role is deployed in the perimeter network and outside the Active Directory forest. Edge Transport servers don't have direct access to Active Directory for configuration and recipient information in the way Client Access or Mailbox servers do. The Edge Transport server uses the Active Directory Lightweight Directory Service (AD LDS) to store configuration and recipient information locally.

You can add an Edge Transport server to an existing Exchange 2013 organization. You don't have to perform any additional Active Directory preparation steps when you install the Edge Transport server.

Note:
If you are adding Edge Transport to an existing Exchange 2010 or Exchange 2007 organization, you will need to install specific rollup updates on your legacy servers before installing Exchange 2013 Edge Transport. For details, see Exchange 2013 system requirements.

When you're planning to deploy Edge Transport servers, you should consider the following issues:

- **Server Capacity** Planning for server capacity includes planning to conduct performance monitoring of the Edge Transport server. Performance monitoring will help you understand how hard the server is working. This information will determine the capacity of your current hardware configuration.
- **Transport Features** The Edge Transport server can provide anti-spam protection at the edge of the network. As part of your planning process, you should determine the anti-spam features that you will enable at the Edge Transport server and how they will be configured.
- **Security** The Edge Transport server role is designed to have a minimal attack surface. Therefore, it's important to correctly secure and manage both the physical access and network access to the server. Planning for security will help you make sure that IP connections are only enabled from authorized servers and from authorized users. For more information, see the Deployment security checklist.

The recommended practice is to put the Edge Transport server within a perimeter network. To make sure that the server can send and receive e-mail and receive recipient and configuration data updates from the Microsoft Exchange EdgeSync service, you must allow communication through the ports that are listed in the following table.

Communication port settings for Edge Transport servers

Network interface	Open port	Protocol	Note

Inbound from and outbound to the Internet	25/TCP	SMTP	This port is required for mail flow to and from the Internet.
Inbound from and outbound to the internal network	25/TCP	SMTP	This port is required for mail flow to and from the Exchange organization.
Local only	50389/TCP	LDAP	This port is used to make a local connection to AD LDS.
Inbound from the internal network	50636/TCP	Secure LDAP	This port is required for EdgeSync synchronization.
Inbound from the internal network	3389/TCP	RDP	Opening this port is optional. It provides more flexibility in managing the Edge Transport servers from inside the internal network by letting you use a remote desktop connection to manage the Edge Transport server.

Note:

The Edge Transport server role uses non-standard LDAP ports. The ports specified in this topic are the LDAP communication ports configured when the Edge Transport server role is installed.

- **EdgeSync** The recommended deployment process is to create an Edge Subscription to subscribe the Edge Transport server to the Exchange organization. When you create an Edge Subscription, recipient and configuration data is replicated from Active Directory to AD LDS. You subscribe an Edge Transport server to an Active Directory site. Then the Microsoft Exchange EdgeSync service that is running on the Mailbox servers in that site periodically updates AD LDS by synchronizing data from Active Directory. The Edge Subscription process automatically provisions the Send connectors that are required to enable mail flow from the Exchange organization to the Internet through an Edge Transport server. If you're using the recipient lookup

or safelist aggregation features on the Edge Transport server, you must subscribe the Edge Transport server to the organization.

Configure DNS settings for the Edge Transport server role

The Edge Transport server is deployed outside the Exchange organization as a stand-alone server in the perimeter network or as a member of a perimeter network Active Directory domain. You need to manually configure the correct DNS suffix for the Edge Transport server role before you install Exchange 2013. If a DNS suffix isn't configured, setup will fail.

Because the Edge Transport server is deployed in the perimeter network, it has network interfaces that are connected to multiple network segments. Each of these network segments has a unique IP configuration. The network interface that is connected to the external, or public, network segment should be configured to use a public DNS server for name resolution. This enables the server to resolve SMTP domain names to MX resource records and route mail to the Internet.

The network interface that is connected to the internal, or private, network segment should be configured to use a DNS server that can resolve the names of the Mailbox servers in your organization, or should have a Hosts file available. The Edge Transport servers and the Mailbox servers must be able to use DNS host resolution to locate each other.

To enable name resolution of Mailbox servers by Edge Transport servers, use one of the following methods:

- Manually create resource records for Mailbox servers in a forward lookup zone on the DNS server that's configured on the internal network adapter of the Edge Transport server.
- Edit the Hosts file on the Edge Transport server to include the Host records for the Mailbox servers. The Hosts file is a local text file in the same format as the 4.3 Berkeley Software Distribution (BSD) UNIX /etc/hosts file. This file maps host names to IP addresses, and the file is stored in the \%Systemroot%\System32\Drivers\Etc folder.

To enable name resolution of Edge Transport servers by Mailbox servers, use one of the following methods:

- Manually create resource records for Edge Transport servers in a forward lookup zone on the DNS server that's configured on the Mailbox server.
- To include the Host records for the Edge Transport servers, edit the Hosts file on the Mailbox servers that are located in the subscribed Active Directory site.

Follow these steps to configure DNS settings for the Edge Transport server:

1. Verify that the DNS server settings for each network interface are correct for the network segment.
2. Configure the DNS suffix for the Edge Transport server name using the following steps:
   a. Open Control Panel, and then choose **System Properties**.
   b. Choose the **Computer Name** tab.
   c. Choose **Change**.
   d. On the **Computer Name Changes** page, click **More**.
   e. In the **Primary DNS suffix of this computer** field, type a DNS domain name and suffix for the

Edge Transport server.

This name can't be changed after the Edge Transport server role is installed.

Override DNS settings

You might need to override the default DNS settings on the Exchange server so mail can be routed and delivered correctly. To do this, modify the **Internal DNS Lookups** and **External DNS Lookups** settings of the transport server's properties. These settings override the settings on the network adapter to route e-mail messages.

Edge Transport server cloned configuration

Exchange Server 2013 > Edge Transport servers >

Topic Last Modified: *2014-02-22*

Edge Transport servers store their configuration information in Active Directory Lightweight Directory Services (AD LDS). You can install more than one Edge Transport server in the perimeter network and use a DNS round robin to help balance network traffic among the Edge Transport servers. Round robin is a simple mechanism used by DNS servers to share and distribute loads for network resources.

To ensure all Edge Transport servers use the same configuration information, use the provided cloned configuration scripts to duplicate your source server's configuration on a target server.

Cloned configuration is used to deploy new Edge Transport servers based on a configured source server. Source server configuration information is duplicated and then exported to an XML file. The XML file is then imported to the target server.

This topic provides an overview of the cloned configuration process. For detailed steps about configuring your Edge Transport servers using cloned configuration, see Configure Edge Transport server using cloned configuration.

Contents

Cloned configuration and EdgeSync

Run the EdgeSync process after you import the cloned configuration. To perform recipient lookup and message security tasks, the Edge Transport server requires data that resides in Active Directory. *EdgeSync* is a collection of processes run on an Exchange 2013 Mailbox server to establish one-way replication of recipient and configuration information from Active Directory to the AD LDS instance on an Edge Transport server. EdgeSync copies only information required for the Edge Transport server to perform anti-spam tasks and information about connector configuration required to enable end-to-end mail flow. EdgeSync performs scheduled updates so the information in AD LDS remains current.

Cloned configuration doesn't duplicate a server's Edge Subscription settings. The certificates used by EdgeSync aren't cloned. You must run the EdgeSync process separately for each Edge Transport server. EdgeSync overwrites any settings included in both the cloned configuration information and in EdgeSync replication information. These settings include Send connectors, Receive connectors, accepted domains, and remote domains.

Cloned configuration process

The cloned configuration process consists of three steps:

1. Export the configuration from the source server.

Run the ExportEdgeConfig.ps1 script (located in %ExchangeInstallPath%Scripts) to export the source server's configuration information to an intermediate XML file.

2. Validate the configuration on the target server.

Run the ImportEdgeConfig.ps1 script (located in %ExchangeInstallPath%Scripts). This script checks the existing information in the intermediate XML file to see whether the exported settings are valid for the target server and then creates an answer file. The answer file specifies the server-specific information used when you import the configuration onto the target server. The answer file contains entries for each source server setting that isn't valid for the target server. You can modify these settings so that they're valid for the target server. If all settings are valid, the answer file contains no entries.

3. Import the configuration on the target server.

The ImportEdgeConfig.ps1 script uses the intermediate XML file and the answer file to clone an existing configuration or to restore the server to a specific configuration.

Step 1: Export the configuration from the source server

After you install and configure the Edge Transport server role, run the ExportEdgeConfig.ps1 script (located in %ExchangeInsallPath%Scripts). This script retrieves the source server's configuration information and stores the information in an intermediate XML file.

The following information is exported from the source server and stored in the intermediate XML file:

- Transport service-related information and log file path information:
  - *ReceiveProtocolLogPath*

    - *SendProtocolLogPath*
    - *MessageTrackingLogPath*
    - *PickupDirectoryPath*
    - *RoutingTableLogPath*
- Transport agent-related information, including status and priority settings of each transport agent.
- All Send connector-related information. If any Send connectors are configured to use credentials, the password is written to the intermediate XML file as an encrypted string. You can use the *-key* parameter with the ImportEdgeConfig.ps1 and ExportEdgeConfig.ps1 scripts to specify the 32-byte string to use for password encryption and decryption. If you don't use the *-key* parameter, a default encryption key is used.
- Receive connector-related information. To modify local network binding and port properties, you must modify the configuration information in the answer file that's created in the validate configuration step.
- Accepted domain configuration.
- Remote domain configuration.
- Anti-spam features configuration settings:
    - IP Allow list information. Only IP Allow list entries that were manually configured by the administrator are exported.
    - IP Block list information.
    - Content filter configuration.
    - Recipient filter configuration.
    - Address rewrite entries.
    - Attachment filter entries.

Return to top

Step 2: Validate the configuration on the target server

The target server is an Exchange 2013 server that has a clean installation of the Edge Transport server role. First, run the ImportEdgeConfig.ps1 script (located at %ExchangeInsallPath%Scripts) on the target server to validate the existing information in the intermediate XML file and to create the answer file. The answer file specifies the server-specific information used when you import the configuration onto the target server. The answer file contains entries for each source server setting that isn't valid for the target server. You will need to modify these settings so that they're valid for the target server. If all settings are valid, the answer file contains no entries. The intermediate XML file can be used for different target servers. The answer file is specific to a target server.

The ImportEdgeConfig.ps1 script (located at %ExchangeInsallPath%Scripts) performs the following tasks:

- Verifies that the data paths and log paths can be created on the target server. If the paths can't be created, a blank path is inserted into the answer file.
- For each Send connector in the XML file, the script adds a blank entry for the source IP address in

the answer file.

- For each Receive connector in the XML file, the script adds a blank entry for the local network bindings in the answer file.

You will need to manually modify the answer file to provide the following information about server-specific settings:

- Fill in the data paths and log paths. If these paths are left blank in the answer file, the paths configured in the intermediate XML file will be used in the next step when you import the configuration onto the target server.
- For each Send connector entry, fill in the source IP address. If this field is left blank, an error will occur in the import configuration step.
- For each Receive connector entry, fill in the local network bindings. If the local network bindings are left blank, an error will occur when you try to import the configuration onto the target server.

Return to top

Step 3: Import the configuration onto the target server

You can perform this step on any target server to clone the configuration of an existing Edge Transport server or to restore the server to a specific configuration. Run the ImportEdgeConfig.ps1 script (located at %ExchangeInsallPath%Scripts) to validate and import the new configuration. After you run this script, the target server's configuration will match the settings in the intermediate XML file and the answer file.

Important:
We recommend that you back up the existing server configuration before running the import configuration process, so that if the cloning operation fails, you can restore the server to its previous stable state.

This step uses the server-specific information provided in the answer file. If a setting isn't specified in the answer file, the data in the intermediate XML file will be used. Before the script modifies the configuration, the script validates the data in the intermediate XML file and the answer file.

Import configuration modifies the following target server configuration settings:

- The transport agent configuration is modified.
- The existing connectors on the target server are removed, and the connectors present in the intermediate XML file are added.
- The existing accepted domains are removed, and the accepted domain entries in the intermediate XML file are added.
- The existing remote domains are removed, and the remote domain entries in the intermediate XML file are added.
- The existing IP Allow list entries are removed, and the IP Allow list entries in the intermediate remote domains file are added.
- The existing IP Block list entries are removed, and the IP Block list entries in the intermediate remote domains file are added.

- The following anti-spam configuration is cloned to the target server:
  - Content filter configuration
  - Recipient filter configuration
  - Address rewrite entries
  - Attachment filter entries

Return to top

Transport configuration information

The settings of the transport configuration object define server-wide email transport settings for an Edge Transport server. When you import the intermediate XML file to the target server, all transport configuration object settings except the following are imported:

- General names and creation dates from the exported XML file
- Send connector information
- Receive connector information
- Attachment filter entries
- The **MaxDumpsterSizePerStorageGroup** attribute entry

After the import process is complete, you may also configure the settings by using the **Set-TransportConfig** cmdlet. For more information, see Set-TransportConfig.

The attributes in the following table are associated with the transport configuration object and the default values. You can configure this object on both Mailbox servers and Edge Transport servers. However, many attributes apply only to the Transport service on Exchange 2013 Mailbox servers. Configuring these attributes on an Edge Transport server will have no effect.

Transport configuration attributes and default values

Attribute	Description	Default value
ClearCategories	This attribute specifies whether to clear Microsoft Outlook categories during content conversion.	True
GenerateCopyOfDSNFor	This attribute specifies the delivery status notification (DSN) codes that cause the DSN message to be copied to the postmaster email address. DSN codes are entered as *x.y.z* and are separated by commas.	5.4.8, 5.4.6, 5.4.4, 5.2.4, 5.2.0, 5.1.4

InternalSMTPServers	This attribute specifies a list of internal SMTP server IP addresses or IP address ranges that should be ignored by Sender ID and connection filtering.	Null
JournalingReportNdrTo	This attribute specifies the email address to which journal reports are sent if the journaling mailbox is unavailable. This attribute doesn't apply to the configuration of an Edge Transport server.	Null
MaxDumpsterSizePerStorageGroup	This attribute specifies the maximum size of the transport dumpster on a Mailbox server. This attribute doesn't apply to the configuration of an Edge Transport server.	18 MB
MaxDumpsterTime	This attribute specifies how long an email message should remain in the transport dumpster on a Mailbox server. This attribute doesn't apply to the configuration of an Edge Transport server.	7.00:00:00
MaxReceiveSize	This attribute specifies the maximum message size that can be received by recipients in the organization. This attribute	10 MB

	doesn't apply to the configuration of an Edge Transport server.	
MaxRecipientEnvelopeLimit	This attribute specifies the maximum number of recipients that are allowed in a single email message. This attribute doesn't apply to the configuration of an Edge Transport server.	5,000
MaxSendSize	This attribute specifies the maximum message size that can be sent by senders in the organization. This attribute doesn't apply to the configuration of an Edge Transport server.	10 MB
TLSReceiveDomainSecureList	This attribute specifies the remote domains that will use mutual Transport Layer Security (TLS) authentication through Receive connectors configured to support Domain Security. Multiple domains may be separated by commas. The wildcard character (*) isn't supported in the domains that are listed in this attribute.	Null
TLSSendDomainSecureList	This attribute specifies the remote domains that will use mutual TLS authentication	Null

	when email is sent through a Send connector configured to support Domain Security and the address space of the target domain. Multiple domains may be separated by commas. The wildcard character (*) isn't supported in the domains that are listed in this attribute.	
VerifySecureSubmitEnabled	This attribute verifies that email clients that are submitting messages from mailboxes on Mailbox servers are using encrypted MAPI submission. This attribute doesn't apply to the configuration of an Edge Transport server. The valid values for this attribute are `$true` or `$false`.	False
VoicemailJournalingEnabled	This attribute specifies whether Unified Messaging voice mail is journaled by the Journaling agent. This attribute doesn't apply to the configuration of an Edge Transport server.	True

Note:

If the Edge Transport server is later subscribed to the Exchange organization, the value of the **InternalSMTPServers** attribute will be overwritten during the EdgeSync process. For more information, see Edge Subscriptions.

Return to top

Configure Edge Transport server using cloned configuration

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-22*

You can use the provided Exchange Management Shell scripts (located in %ExchangeInstallPath% Scripts) to duplicate the configuration of an Edge Transport server. This process is referred to as *cloned configuration*. *Cloned configuration* is the practice of deploying new Edge Transport servers based on configuration information from a previously configured source server. The configuration information from the previously configured source server is copied and exported to an XML file, which is then imported to the target server. For an overview of this process, see Edge Transport server cloned configuration.

Edge Transport server configuration information is stored in Active Directory Lightweight Directory Services (AD LDS) and isn't replicated among multiple Edge Transport servers. Using cloned configuration, you can ensure that every Edge Transport server deployed in your perimeter network is using the same configuration.

Two Shell scripts are used to perform cloned configuration tasks:

- **ExportEdgeConfig.ps1** Exports all user-configured settings and data from an Edge Transport server and stores that data in an XML file.
- **ImportEdgeConfig.ps1** During the validate configuration step, the ImportEdgeConfig.ps1 script checks the exported XML file to see whether the server-specific export settings are valid for the target server. If settings need to be modified, the script writes the invalid settings to an answer file you can modify to specify target server information, which is then used during the import configuration step. During the import configuration step, the script imports all user-configured settings and data stored in the intermediate XML file created by the ExportEdgeConfig.ps1 script.

Both these scripts are located in the %ExchangeInstallPath%Scripts folder.

Before you begin

- Estimated time to complete this task: 30 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Edge Transport servers" entry in the Mail flow permissions topic.
- Cloned configuration doesn't duplicate a server's Edge Subscription settings. EdgeSync certificates aren't cloned. You need to run the EdgeSync process separately for each Edge

Transport server. EdgeSync overwrites any settings included in both cloned configuration information and in EdgeSync replication information.

- If any Send connectors are configured to use credentials, the password is written to the intermediate XML file as an encrypted string. You can use the *-key* parameter with the ImportEdgeConfig.ps1 and ExportEdgeConfig.ps1 scripts to specify the 32-byte string to use for password encryption and decryption. If you don't use the *-key* parameter, a default encryption key is used.
- You can only use the Shell to perform this procedure. To learn how to open the Shell in your on-premises Exchange organization, see Open the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Export the source server configuration data to a file on the source server

1. Copy the ExportEdgeConfig.ps1 script to the root folder of your user profile on the source server.
2. To export the source server configuration data to a file on the source server, use the following syntax.

```
./ExportEdgeConfig.ps1 -CloneConfigData:"<configuration file>"
```

For example, to export the source server configuration data to the file C:\CloneConfigData.xml, run the following command.

```
./ExportEdgeConfig.ps1 -CloneConfigData:"C:\CloneConfigData.xml"
```

How do you know this step worked?

You'll know you successfully exported the source configuration data to a file when the confirmation message, "Edge configuration data is exported successfully to: <output file path>" appears.

Step 2: Validate the configuration file and create an answer file on the target server

1. Copy the source server configuration file you exported in the previous step to the target Edge Transport server.
2. Copy the ImportEdgeConfig.ps1 script to the root folder of your user profile on the target server.
3. To validate the configuration file and use the results to create an answer file on the target server, use the following syntax.

```
./ImportEdgeConfig.ps1 -CloneConfigData:"<configuration
file>" -IsImport $false -CloneConfigAnswer:"<answer file>"
```

For example, to validate the configuration file C:\CloneConfigData.xml, and create the answer file C:\CloneConfigAnswer.xml, run the following command.

```
./ImportEdgeConfig.ps1 -CloneConfigData:"C:
\CloneConfigData.xml" -IsImport $false -
CloneConfigAnswer:"C:\CloneConfigAnswer.xml"
```

4. Open the answer file and modify any settings that are invalid for the target server. If no modifications are required, the answer file will have no entries. Save your changes.

How do you know this step worked?

You'll know you successfully validated the configuration file and created an answer file when the confirmation message, "Answer file is successfully created" appears.

Step 3 Import the configuration file on the target server

To import the configuration file on the target server, use the following syntax.

```
./ImportEdgeConfig.ps1 -CloneConfigData:"<Configuration
file>" -IsImport $true -CloneConfigAnswer:"<answer file>"
```

For example, to import the configuration file C:\CloneConfigData.xml by using the answer file C:\CloneConfigAnswer.xml, run the following command.

```
./ImportEdgeConfig.ps1 -CloneConfigData:"C:
\CloneConfigData.xml" -IsImport $true -
CloneConfigAnswer:"C:\CloneConfigAnswer.xml"
```

How do you know this step worked?

You'll know you successfully imported the configuration file on the target server when the confirmation message, "Importing Edge configuration information succeeded" appears.

Use an Exchange 2010 or 2007 Edge Transport server in Exchange 2013

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-22*

The Edge Transport server is available in Microsoft Exchange Server 2013 Service Pack 1 (SP1). However, you can continue to use existing Exchange Server 2007 or Exchange Server 2010 Edge Transport servers that you have deployed in your perimeter network. Or, you can install a new Exchange 2007 or Exchange 2010 Edge Transport server in your perimeter network for a new or upgraded Exchange 2013 organization.

Here are the things you need to know:

- An Exchange 2007 or Exchange 2010 Edge Transport server expects a connection to a Hub Transport server. In Exchange 2013, the Transport service exists on the Mailbox server. Therefore, Internet mail flow occurs between the Transport service on the Mailbox server and the Edge Transport server, which effectively bypasses the Exchange 2013 Client Access server.
- You can subscribe an Exchange 2007 or Exchange 2010 Edge Transport server to an Active Directory site that contains only Exchange 2013 servers. You can import the Edge Subscription file and run EdgeSync on a standalone Exchange 2013 Mailbox server, or on a server where the Mailbox server and the Client Access server are installed on the same computer. You can't import the Edge Subscription file or run EdgeSync on a standalone Exchange 2013 Client Access server.
- The procedures to deploy a new Exchange 2007 or Exchange 2010 Edge Transport server in your Exchange 2013 organization are basically the same as in previous versions of Exchange. However, any procedures that are performed on the Hub Transport server are performed on the Mailbox server in Exchange 2013. The procedures are:
  - Configure Internet Mail Flow Through a Subscribed Edge Transport Server
  - Configure Mail Flow Between an Edge Transport Server and Hub Transport Servers Without Using EdgeSync

Address rewriting on Edge Transport servers

Topic Last Modified: *2014-02-10*

Address rewriting modifies email addresses of senders and recipients in messages that enter or leave your organization through an Edge Transport server. Two transport agents on the Edge Transport server provide the rewriting functionality: the Address Rewriting Inbound Agent and the Address Rewriting Outbound Agent. The primary reason for address rewriting on outbound messages is to present a single, consistent email domain to external recipients. The primary reason for address rewriting on inbound messages is to deliver messages to the correct recipient.

The *address rewrite entry*, which you create, specifies the internal addresses (the email addresses you want to change) and the external addresses (the final email addresses you want). You can specify whether email addresses are rewritten in inbound and outbound messages, or in outbound messages only. You can create address writing entries for a single user (chris@contoso.com to support@contoso.com), all users in a single domain (contoso.com to fabrikam.com), or for users in multiple subdomains with exceptions (*.fabrikam.com to contoso.com, except legal.fabrikam.com).

Important:
Regardless of how you plan to use address rewriting, you need to verify that the resulting email addresses are unique in your organization so you don't end up with duplicates. This is because address rewriting doesn't verify the uniqueness of a rewritten email address.

Contents

Scenarios for address rewriting

The following scenarios are examples of how you can use address rewriting:

- **Group consolidation** Some organizations segment their internal businesses into separate domains that are based on business or technical requirements. This configuration can cause email messages to appear as if they come from separate groups or even separate organizations.

The following example shows how an organization, Contoso, Ltd., can hide its internal subdomains from external recipients:

  - Outbound messages from the northamerica.contoso.com, europe.contoso.com, and asia.contoso.com domains are rewritten so they appear to originate from a single contoso.com

domain. All messages are rewritten as they pass through Edge Transport servers that provide SMTP connectivity between the whole organization and the Internet.
    - Inbound messages to contoso.com recipients are relayed by the Edge Transport server to a Mailbox server. The message is delivered to the correct recipient based on the proxy address that's configured on the recipient's mailbox.
- **Mergers and acquisitions** An acquired company might continue to run as a separate business, but you can use address rewriting to make the two organizations appear as if they're one integrated organization.

The following example shows how Contoso, Ltd. can hide the email domain of the newly acquired company, Fourth Coffee:

- Contoso, Ltd. wants all outbound messages from Fourth Coffee's Exchange organization to appear as if they originate from contoso.com. All messages from both organizations are sent through the Edge Transport servers at Contoso, Ltd., where email messages are rewritten from *user*@fourthcoffee.com to *user*@contoso.com.
- Inbound messages to *user*@contoso.com are rewritten and routed to *user*@fourthcoffee.com mailboxes. Inbound messages that are sent to *user*@fourthcoffee.com are routed directly to Fourth Coffee's email servers.

- **Partners** Many organizations use external partners to provide services for their customers, other organizations, or their own organization. To avoid confusion, the organization might replace the email domain of the partner organization with its own email domain.

The following example shows how Contoso, Ltd. can hide a partner's email domain:

- Contoso, Ltd. provides support for the larger Wingtip Toys organization. Wingtip Toys wants a unified email experience for its customers, and it requires all messages from support personnel at Contoso, Ltd. to appear as if they were sent from Wingtip Toys. All outbound messages that relate to Wingtip Toys are sent through their Edge Transport servers, and all contoso.com email addresses are rewritten to wingtiptoys.com email addresses.
- Inbound messages for support@wingtiptoys.com are accepted by Wingtip Toy's Edge Transport servers, rewritten, and then routed to the support@contoso.com email address.

Return to top

Message properties modified by address rewriting

A standard SMTP email message consists of a *message envelope* and message content. The message envelope contains information required for transmitting and delivering the message between SMTP mail servers. The message content contains message header fields (collectively called the *message header*) and the message body. The message envelope is described in RFC 2821, and the message header is described in RFC 2822.

When a sender composes an email message and submits it for delivery, the message contains the basic information required to comply with SMTP standards, such as a sender, a recipient, the date and time that the message was composed, an optional subject line, and an optional message body. This information is contained in the message itself and, by definition, in the message header.

The sender's mail server generates a message envelope for the message by using the sender's and recipient's information found in the message header. It then transmits the message to the Internet for delivery to the recipient's mail server. Recipients never see the message envelope because it's generated by the message transmission process, and it isn't actually part of the message.

Address rewriting changes an email address by rewriting specific fields in the message header or message envelope. Address rewriting changes several fields in outbound messages but only one field in inbound email messages. The following table shows which SMTP header fields are rewritten in outbound and inbound messages.

Message fields rewritten on outbound and inbound messages

Field name	Location	Outbound messages	Inbound messages
MAIL FROM	Message envelope	Rewritten	Not rewritten
RCPT TO	Message envelope	Not rewritten	Rewritten
To	Message header	Rewritten	Not rewritten
Cc	Message header	Rewritten	Not rewritten
From	Message header	Rewritten	Not rewritten
Sender	Message header	Rewritten	Not rewritten
Reply-To	Message header	Rewritten	Not rewritten
Return-Receipt-To	Message header	Rewritten	Not rewritten
Disposition-Notification-To	Message header	Rewritten	Not rewritten
Resent-From	Message header	Rewritten	Not rewritten
Resent-Sender	Message header	Rewritten	Not rewritten

Return to top

What address rewriting doesn't change

Address rewriting doesn't modify any message header fields that would break SMTP functionality. For example, modifying certain header fields can affect routing loop detection, invalidate the signature, or make a rights-protected message unreadable. Therefore, the following header fields

aren't modified by address rewriting.

- **Return-Path**
- **Received**
- **Message-ID**
- **X-MS-TNEF-Correlator**
- **Content-Type Boundary=string**
- Header fields located inside MIME body parts

Address rewriting ignores domains that aren't controlled by the Exchange organization. In other words, address rewriting doesn't rewrite header fields that contain domains for which the Exchange organization isn't authoritative. Rewriting such domains would cause an uncontrollable form of message relay.

Address rewriting also doesn't modify the header fields of messages that are embedded in another message. Senders and recipients expect embedded messages to remain intact and be delivered without modification, as long as the messages don't trigger transport rules that are implemented between the sender and recipient.

Return to top

Considerations for outbound-only address rewriting

Outbound-only address rewriting on an Edge Transport server modifies the sender's email address as the message leaves the Exchange organization. You can configure outbound-only address rewriting for a single user (chris@contoso.com to support@contoso.com) or for all users in a single domain (contoso.com to fabrikam.com). You are required to configure outbound-only address rewriting for users in multiple subdomains (*.fabrikam.com to .contoso.com).

The rewritten email address must be configured as a proxy address on the affected recipients. For example, if laura@sales.contoso.com is rewritten to laura@contoso.com, the proxy address laura@contoso.com must be configured on Laura's mailbox. This allows replies and inbound messages to be delivered correctly.

Return to top

Considerations for inbound and outbound address rewriting

Inbound and outbound, or *bidirectional* address rewriting on an Edge Transport server modifies the sender's email address in messages that leave the Exchange organization, and it modifies the recipient's email address in messages that enter the Exchange organization.

You can configure outbound-only address rewriting for a single user (chris@contoso.com to support@contoso.com) and all users in a single domain (contoso.com to fabrikam.com). You can’t

configure bidirectional address rewriting for users in multiple subdomains (*.fabrikam.com to contoso.com).

Return to top

Considerations for rewriting email addresses in multiple domains

When you flatten multiple internal domains or subdomains into a single external domain, you need to consider the following factors:

- **Verify unique aliases** All email aliases (the part to the left of the @ sign) must be unique across all subdomains. For example, if there is a joe@sales.contoso.com, there can't be a joe@marketing.contoso.com because the rewritten email address for both users would be joe@contoso.com.
- **Add proxy addresses** The rewritten email address must be configured as a proxy address for all affected senders in the affected domains. For example, if joe@sales.contoso.com is rewritten to joe@contoso.com, you need to add the proxy address joe@contoso.com to Joe's mailbox. This allows replies and inbound messages to be delivered correctly.
- **Mail contacts for non-Exchange organizations** If you're rewriting email addresses from a non-Exchange email system, you need to create email contacts in Exchange to represent the users in the non-Exchange email system. These email contacts must contain the original email addresses and the rewritten email addresses. For example, if joe@unix.contoso.com is rewritten to joe@contoso.com, you need to create a mail contact with joe@unix.contoso.com as the external email address and joe@contoso.com as a proxy address.

Verify unique aliases

When you rewrite email addresses in multiple subdomains, you need to make sure that all email aliases are unique across all your subdomains. For example, consider the following configuration:

The following users are in the subdomains sales.contoso.com, marketing.contoso.com, and research.contoso.com:

- maria@sales.contoso.com
- chris@sales.contoso.com
- david@marketing.contoso.com
- brian@marketing.contoso.com
- chris@research.contoso.com
- adam@research.contoso.com

Suppose you want to rewrite the subdomains sales.contoso.com, marketing.contoso.com, and research.contoso.com into the single domain contoso.com.

When the email addresses in each subdomain are rewritten, a conflict occurs between

chris@sales.contoso.com and chris@research.contoso.com because both email addresses are rewritten to chris@contoso.com. To resolve this situation, you need to change the email address of one of the affected recipients. For example, you can change chris@research.contoso.com to christopher@research.contoso.com so the email address is rewritten to christopher@contoso.com.

Return to top

Priority of address rewrite entries

If a user's email address matches multiple address rewrite entries, the email address is only rewritten once based on the closest match. The following list describes the order of precedence of address rewrite entries from highest priority to lowest priority:

1. **Individual email addresses** An address rewrite entry is configured to rewrite the email address of john@contoso.com to support@contoso.com.
2. **Domain or subdomain mapping** An address rewrite entry is configured to rewrite all contoso.com email addresses to northwindtraders.com or all sales.contoso.com email addresses to contoso.com.
3. **Domain flattening** An address rewrite entry is configured to rewrite *.contoso.com email addresses to contoso.com.

For example, consider an Edge Transport server where the following outbound address rewrite entries are configured:

- *.contoso.com email addresses are rewritten to contoso.com
- japan.sales.contoso.com email addresses are rewritten to contoso.jp

If masato@japan.sales.contoso.com sends an email message, the address is rewritten to masato@contoso.jp, because that entry most closely matches the sender's email address.

Return to top

Digitally signed, encrypted, and rights-protected messages

Address rewriting shouldn't affect most signed, encrypted, or rights-protected messages. If address rewriting were to invalidate or otherwise change the security status of these types of messages in any way, address rewriting isn't applied.

The following values can be rewritten because the information isn't part of message signing, encryption, or rights protection:

- Fields in the message envelope
- Top-level message body headers

The following values aren't rewritten because the information is part of message signing, encryption, or rights protection:

- Header fields located inside MIME body parts that may be signed
- The boundary string parameter of the MIME content type

Return to top

Manage address rewriting on Edge Transport servers

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-10*

You use the Exchange Management Shell on an Edge Transport server for all management tasks related to address rewriting and the address rewriting agents. For more information about address rewriting, see Address rewriting on Edge Transport servers.

You can create address rewrite entries that apply to a single recipient, to all recipients in a specific domain or subdomain, or all recipients in multiple subdomains. Address rewriting can be outbound only, or inbound and outbound (bidirectional). When you create address rewrite entries, remember the following:

- Verify that the resulting email addresses are unique in your organization.
- Only literal strings are supported in the email address values.
- The wildcard character (*) is supported only in the internal address (the addresses you want to change). Valid syntax for using the wildcard character is ***.contoso.com**. The values ***contoso.com** or **sales.*.com** are not allowed.
- When you use the wildcard character, you need to configure the address rewriting as outbound only (you need to set the *OutboundOnly* parameter to the value `$true`).
- When you configure outbound-only address rewriting (by setting the *OutboundOnly* parameter to the value `$true`), you need to configure proxy addresses on the affected recipients. This allows mail that is sent to the rewritten address to be delivered correctly.
- By default, address rewriting is bidirectional for single recipients or all recipients in a specific domain or subdomain (the default value for the *OutboundOnly* parameter is `$false`).

What do you need to know before you begin?

- Estimated time to complete each procedure: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Edge Transport servers" section in the Mail flow permissions topic.
- You can only use the Shell to perform this procedure.
- Be careful when you configure address rewriting. Any changes that you make are immediately

applied when you run the command. Consider running the command with the *WhatIf* parameter. For more information about the *WhatIf* parameter, see WhatIf, Confirm, and ValidateOnly switches.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Use the Shell to enable or disable address rewriting

To completely enable or disable address rewriting, you enable or disable the address rewriting agents. By default, the address rewriting agents on an Edge Transport server are enabled.

To disable address rewriting, run the following commands:

```
Disable-TransportAgent "Address Rewriting Inbound Agent"
Disable-TransportAgent "Address Rewriting Outbound Agent"
```

To enable address rewriting, run the following commands:

```
Enable-TransportAgent "Address Rewriting Inbound Agent"
Enable-TransportAgent "Address Rewriting Outbound Agent"
```

How do you know this worked?

To verify that you have successfully enabled or disabled address rewriting, do the following:

1. Run the following command:

```
Get-TransportAgent
```

2. Verify the values of the **Enabled** property for the Address Rewriting Inbound Agent and the Address Rewriting Outbound Agent are the values you configured.

Use the Shell to view address rewrite entries

To view a summary list of all address rewrite entries, run the following command.

```
Get-AddressRewriteEntry
```

To view details of an address rewrite entry, use the following syntax.

```
Get-AddressRewriteEntry <AddressRewriteEntryIdentity> |
```

```
Format-List
```

The following example displays the details of the address rewrite entry named Rewrite Contoso.com to Northwindtraders.com:

```
Get-AddressRewriteEntry "Rewrite Contoso.com to
Northwindtraders.com" | Format-List
```

Use the Shell to create address rewrite entries

Rewrite the email addresses of single recipients

To rewrite the email address for a single recipient, use the following syntax:

```
New-AddressRewriteEntry -Name "<Descriptive Name>" -
InternalAddress <internal email address> -ExternalAddress
<external email address> [-OutboundOnly <$true | $false>]
```

The following example rewrites the email address of all messages entering and leaving the Exchange organization for the recipient joe@contoso.com. Outbound messages are rewritten so they appear to come from support@nortwindtraders.com. Inbound messages sent to support@northwindtraders.com are rewritten to joe@contoso.com for delivery to the recipient (the *OutboundOnly* parameter is `$false` by default).

```
New-AddressRewriteEntry -Name "joe@contoso.com to
support@northwindtraders.com" -InternalAddress
joe@contoso.com -ExternalAddress
support@northwindtraders.com
```

Rewrite email addresses for recipients in a single domain or subdomain

To rewrite the email addresses for recipients in a single domain or subdomain, use the following syntax:

```
New-AddressRewriteEntry -Name "<Descriptive Name>" -
InternalAddress <domain or subdomain> -ExternalAddress
<domain> [-OutboundOnly <$true | $false>]
```

The following example rewrites the email addresses of all messages entering and leaving the Exchange organization for recipients in the contoso.com domain. Outbound messages are rewritten so they appear to come from the fabrikam.com domain. Inbound messages sent to fabrikam.com email addresses are rewritten to contoso.com for delivery to the recipients (the *OutboundOnly* parameter is `$false` by default).

```
New-AddressRewriteEntry -Name "Contoso to Fabrikam" -
InternalAddress contoso.com -ExternalAddress fabrikam.com
```

The following example rewrites the email addresses of all messages leaving the Exchange organization that are sent by recipients in the sales.contoso.com subdomain. Outbound messages are rewritten so they appear to come from the contoso.com domain. Inbound messages sent to contoso.com email addresses aren't rewritten.

```
New-AddressRewriteEntry -Name "sales.contoso.com to
contoso.com" -InternalAddress sales.contoso.com -
ExternalAddress contoso.com -OutboundOnly $true
```

Rewrite email addresses for recipients in multiple subdomains

To rewrite the email addresses for recipients in a domain and all subdomains, use the following syntax.

```
New-AddressRewriteEntry -Name "<Descriptive Name>" -
InternalAddress *.<domain> -ExternalAddress <domain> -
OutboundOnly $true [-ExceptionList <domain1,domain2...>]
```

The following example rewrites the email addresses of all messages leaving the Exchange organization that are sent by recipients in the contoso.com domain and all subdomains. Outbound messages are rewritten so they appear to come from the contoso.com domain. Inbound messages sent to contoso.com recipients can't be rewritten because a wildcard is used in the *InternalAddress* parameter.

```
New-AddressRewriteEntry -Name "Rewrite all contoso.com
subdomains" -InternalAddress *.contoso.com -ExternalAddress
contoso.com -OutboundOnly $true
```

The following example is just like the previous example, except now messages sent by recipients in the legal.contoso.com and corp.contoso.com subdomains are never rewritten:

```
New-AddressRewriteEntry -Name "Rewrite all contoso.com
subdomains except legal.contoso.com and corp.contoso.com" -
InternalAddress *.contoso.com -ExternalAddress contoso.com
-OutboundOnly $true -ExceptionList
legal.contoso.com,corp.contoso.com
```

How do you know this worked?

To verify that you have successfully created address rewrite entries, do the following:
1. Run the command `Get-AddressRewriteEntry <AddressRewriteEntryIdentity> | Format-List` and

verify the settings displayed are the settings you configured.
2. From a mailbox that's affected by an address rewrite entry, send a test message to an external mailbox. Verify the test message appears to originate from the rewritten email address.
3. Reply to the test message from the external mailbox. Verify the original mailbox receives the reply.

Use the Shell to modify address rewrite entries

The configuration options that are available when you modify an existing address rewrite entry are identical to the configuration options when you create a new address rewrite entry.

Modify address rewrite entries for single recipients

To modify an address rewrite entry that rewrites the email address of a single recipient, use the following syntax:

```
Set-AddressRewriteEntry <AddressRewriteEntryIdentity> -Name
"<Descriptive Name>" -InternalAddress <internal email
address> -ExternalAddress <external email address> -
OutboundOnly <$true | $false>
```

The following example modifies the following properties of the single recipient address rewrite entry named "joe@contoso.com to support@nortwindtraders.com":

- Changes the external address to support@northwindtraders.net.
- Changes the name of the address rewrite entry to "joe@contoso.com to support@northwindtraders.net".
- Changes the value of *OutboundOnly* to `$true`. Note that this change requires you to configure support@northwindtraders.net as a proxy address on Joe's mailbox.

```
Set-AddressRewriteEntry "joe@contoso.com to
support@nortwindtraders.com" -Name "joe@contoso.com to
support@northwindtraders.net" -ExternalAddress
support@northwindtraders.net -OutboundOnly $true
```

Modify address rewrite entries for recipients in single domains or subdomains

To modify an address rewrite entry that rewrites the email addresses of recipients from a single domain or subdomain, use the following syntax.

```
Set-AddressRewriteEntry <AddressRewriteEntryIdentity> -Name
"<Descriptive Name>" -InternalAddress <domain or subdomain>
-ExternalAddress <domain> -OutboundOnly <$true | $false>
```

The following example changes the internal address value of the single domain address rewrite entry named "Northwind Traders to Contoso".

```
Set-AddressRewriteEntry "Northwindtraders to Contoso" -
InternalAddress northwindtraders.net
```

Modify address rewrite entries for recipients in multiple subdomains

To modify an address rewrite entry that rewrites the email address of recipients in a domain and all subdomains, use the following syntax.

```
Set-AddressRewriteEntry <AddressRewriteEntryIdentity> -Name
"<Descriptive Name>" -InternalAddress *.<domain> -
ExternalAddress <domain> -ExceptionList <list of domains>
```

To replace the existing exception list values of a multiple subdomain address rewrite entry, use the following syntax:

```
Set-AddressRewriteEntry <AddressRewriteEntryIdentity> -
ExceptionList <domain1,domain2,...>
```

The following example replaces the existing exception list for the multiple subdomain address rewrite entry named Contoso to Northwind Traders with the values marketing.contoso.com and legal.contoso.com:

```
Set-AddressRewriteEntry "Contoso to Northwind Traders" -
ExceptionList sales.contoso.com,legal.contoso.com
```

To selectively add or remove exception list values from a multiple subdomain address rewrite entry without modifying any existing exception list values, use the following syntax:

```
Set-AddressRewriteEntry <AddressRewriteEntryIdentity> -
ExceptionList @{Add="<domain1>","<domain2>"...;
Remove="<domain1>","<domain2>"...}
```

The following example adds finanace.contoso.com and removes marketing.contoso.com from the exception list of the multiple subdomain address rewrite entry named Contoso to Northwind Traders:

```
Set-AddressRewriteEntry "Contoso to Northwind Traders" -
ExceptionList @{Add="finanace.contoso.com";
Remove="marketing.contoso.com"}
```

How do you know this worked?

To verify that you have successfully modified an address rewrite entry, do the following:
1. Run the command `Get-AddressRewriteEntry <AddressRewriteEntryIdentity> | Format-List` and

verify the settings displayed are the settings you configured.
2. From a mailbox that's affected by an address rewrite entry, send a test message to an external mailbox. Verify the test message appears to originate from the rewritten email address.
3. From the external mailbox, reply to the test message. Verify the original mailbox receives the reply.

Use the Shell to remove address rewrite entries

To remove a single address rewrite entry, use the following syntax:

```
Remove-AddressRewriteEntry <AddressRewriteEntryIdentity>
```

The following example removes the address rewrite entry named "Contoso.com to Northwindtraders.com":

```
Remove-AddressRewriteEntry "Contoso.com to
Northwindtraders.com"
```

To remove multiple address rewrite entries, use the following syntax:

```
Get-AddressRewriteEntry [<search criteria>] | Remove-
AddressRewriteEntry [-WhatIf]
```

The following example removes all address rewrite entries:

```
Get-AddressRewriteEntry | Remove-AddressRewriteEntry
```

The following example simulates the removal of address rewrite entries that contain the text "to contoso.com" in the name. The *WhatIf* switch allows you to preview the result without committing any changes.

```
Get-AddressRewriteEntry "*to contoso.com" | Remove-
AddressRewriteEntry -WhatIf
```

If you are satisfied with the result, run the command again without the *WhatIf* switch to remove the address rewrite entries.

```
Get-AddressRewriteEntry "*to contoso.com" | Remove-
AddressRewriteEntry
```

How do you know this worked?

To verify that you have successfully removed an address rewrite entry, do the following:

1. Run the command `Get-AddressRewriteEntry`, and verify that the address rewrite entries you removed aren't listed.

2. From a mailbox that's affected by an address rewrite entry, send a test message to an external mailbox. Verify the test message is no longer affected by the removed address rewrite entry.
3. From the external mailbox, reply to the test message. Verify the original mailbox receives the reply and that the message is unaffected by the removed address rewrite entry.

Import address rewrite entries on Edge Transport servers

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-10*

You can bulk-create or import address rewriting information into an Edge Transport server by using a comma-separated value (CSV) file. The following list describes common scenarios that require you to do this:

- You are replacing an address rewriting solution with an Edge Transport server.
- You enter into an agreement with a third-party solution provider that requires you to rewrite their email addresses.
- You acquire another organization, and you need to temporarily rewrite the email addresses in the acquired organization.

You can use any text editor, like Notepad, or an application like Microsoft Excel, to create the CSV file. Format the file as described in this topic and save it as a .csv file.

The first row, or *header row*, of the CSV file lists the names of the parameters. Each parameter is separated by a comma. The required and optional parameters are described in the following table.

Parameter	Required or optional	Description
Name	Required	A unique, descriptive name for the address rewrites entry.
InternalAddress	Required	The address you want to change. You can use the following values: • A single email address (chris@contoso.com) • A single domain or subdomain (contoso.com or sales.contoso.com)

		• A domain and all subdomains (*.contoso.com)
ExternalAddress	Required	The final email address you want. You can use the following values: • A single email address if you specified a single email address for *InternalAddress* • A single domain or subdomain for all other values of *InternalAddress*
ExceptionList	Optional	Available only when you are rewriting email addresses in a domain and all subdomains (*.contoso.com). Specifies one or more subdomains you want to exclude from address rewriting. Enclose the value in double quotation marks, and separate multiple values by commas. For example, `"marketing.contoso.com"` or `"marketing.contoso.com,legal.contoso.com"`.
OutboundOnly	Optional	`False` means that addresses are written on inbound and outbound mail. `True` means that addresses are rewritten on outbound mail only, and you need to manually configure the rewritten email address as a proxy address on the affected recipients. The default value is `False`, but you must set it to `True` if *InternalAddress* contains the

		wildcard character (*.contoso.com). The *OutboundOnly* parameter value in the CSV file is `True` or `False`, not `$True` or `$False`.

Each row under the header row represents an individual address rewrite entry. The values in each row must be in the same order as the parameter names in the header row. Each value is separated by a comma.

What do you need to know before you begin?

- Estimated time to complete this task: 30 minutes
- Make sure you understand the ramifications of address rewriting. For example, the rewritten email address must be unique in your Exchange organization, and you might need to configure proxy addresses on the affected recipients. For more information, see Address rewriting on Edge Transport servers and Manage address rewriting on Edge Transport servers.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Address Rewriting agent" entry in the Mail flow permissions topic.
- If you have more than one Edge Transport server, we recommend that you use the procedures in this topic to import the address rewrite entries into a single Edge Transport server and then clone the configuration of that Edge Transport server to the other Edge Transport servers in your organization. For more information about how to clone an Edge Transport server, see Edge Transport server cloned configuration.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

How do you do this?

Step 1: Create the CSV file

When you create the CSV file, consider the following items:

- If you specify values for optional parameters in the CSV file, every row must include a value in that column. If you want to create multiple address rewrite entries where some entries have optional parameters and some entries do not, you need to separate those address rewrite entries into two separate CSV files and then import each CSV file separately.

- If the CSV file contains non-ASCII characters, be sure to save the CSV file with UTF-8 encoding or other Unicode encoding. Depending on the application, saving the CSV file with UTF-8 encoding or other Unicode encoding might be easier when the system locale of the computer matches the language used in the CSV file.

The following example shows how a CSV file can be populated with the optional *ExceptionList* and *OutboundOnly* parameters included:

```
Name,InternalAddress,ExternalAddress,ExceptionList,Outbound
Only
"Wingtip
UK",*.wingtiptoys.co.uk,tailspintoys.com,"legal.wingtiptoys
.co.uk,finance.wingtiptoys.co.uk,support.wingtiptoys.co.uk"
,True
"Wingtip
USA",*.wingtiptoys.com,tailspintoys.com,"legal.wingtiptoys.
com,finance.wingtiptoys.com,support.wingtiptoys.com,corp.wi
ngtiptoys.com",True
"Wingtip
Canada",*.wingtiptoys.ca,tailspintoys.com,"legal.wingtiptoy
s.ca,finance.wingtiptoys.ca,support.wingtiptoys.ca",True
```

Step 2: Import the CSV file

To import the CSV file, use the following syntax:

```
Import-Csv <FileNameAndPath> | ForEach {New-
AddressRewriteEntry -Name $_.Name -InternalAddress
$_.InternalAddress -ExternalAddress $_.ExternalAddress -
OutboundOnly ([Bool]::Parse($_.OutboundOnly)) -
ExceptionList $_.ExceptionList}
```

This example imports the address rewrite entries from C:\My Documents \ImportAddressRewriteEntries.csv.

```
Import-Csv "C:\My Documents
\ImportAddressRewriteEntries.csv" | ForEach {New-
AddressRewriteEntry -Name $_.Name -InternalAddress
$_.InternalAddress -ExternalAddress $_.ExternalAddress -
OutboundOnly ([Bool]::Parse($_.OutboundOnly)) -
ExceptionList $_.ExceptionList}
```

How do you know this step worked?

To verify that you have successfully imported address rewrite entries from a CSV file, do the following:

1. To see all address rewrite entries, run the command `Get-AddressRewriteEntry`.
2. To see details about a specific address rewrite entry, run the command `Get-AddressRewriteEntry <AddressRewriteIdentity> | Format-List`

High availability and site resilience

Exchange Server 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-01*

You can protect your Exchange Server 2013 mailbox databases and the data they contain by configuring your Mailbox servers and databases for high availability and site resilience. Exchange 2013 minimizes the cost and complexity of deploying a highly available and resilient messaging solution while providing high levels of service and data availability and support for very large mailboxes.

Exchange 2013 enables customers of all sizes and in all segments to economically deploy a messaging continuity service in their organization by building on the native replication capabilities and high availability architecture introduced in Exchange 2010. For a list of changes over Exchange 2010 and Exchange 2007, see Changes to high availability and site resilience over previous versions.

Contents

Key terminology

The following key terms are important to understand high availability or site resilience:

Active Manager

An internal Exchange component which runs inside the Microsoft Exchange Replication service

that's responsible for failure monitoring and corrective action through failover within a database availability group (DAG).

AutoDatabaseMountDial

A property setting of a Mailbox server that determines whether a passive database copy will automatically mount as the new active copy, based on the number of log files missing by the copy being mounted.

Continuous replication - block mode

In block mode, as each update is written to the active database copy's active log buffer, it's also shipped to a log buffer on each of the passive mailbox copies in block mode. When the log buffer is full, each database copy builds, inspects, and creates the next log file in the generation sequence.

Continuous replication - file mode

In file mode, closed transaction log files are pushed from the active database copy to one or more passive database copies.

Database availability group

A group of up to 16 Exchange 2013 Mailbox servers that hosts a set of replicated databases.

Database mobility

The ability of an Exchange 2013 mailbox database to be replicated to and mounted on other Exchange 2013 Mailbox servers.

Datacenter

Typically this refers to an Active Directory site; however, it can also refer to a physical site. In the context of this documentation, datacenter equals Active Directory site.

Datacenter Activation Coordination mode

A property of the DAG setting that, when enabled, forces the Microsoft Exchange Replication service to acquire permission to mount databases at startup.

Disaster recovery

Any process used to manually recover from a failure. This can be a failure that affects a single item, or it can be a failure that affects an entire physical location.

Exchange third-party replication API

An Exchange-provided API that enables use of third-party synchronous replication for a DAG instead of continuous replication.

High availability

A solution that provides service availability, data availability, and automatic recovery from failures that affect the service or data (such as a network, storage, or server failure).

Incremental deployment

The ability to deploy high availability and site resilience after Exchange 2013 is installed.

Lagged mailbox database copy

A passive mailbox database copy that has a log replay lag time greater than zero.

Mailbox database copy

A mailbox database (.edb file and logs), which is either active or passive.

Mailbox resiliency

The name of a unified high availability and site resilience solution in Exchange 2013.

Managed availability

A set of internal processes made up of probes, monitors, and responders that incorporate monitoring and high availability across all server roles and all protocols.

**over* (pronounced "star over")

Short for *switchovers* and *failovers*. A switchover is a manual activation of one or more database copies. A failover is an automatic activation of one or more database copies after a failure.

Safety Net

Formerly known as transport dumpster, this is a feature of the transport service that stores a copy of all messages for *X* days. The default setting is 2 days.

Shadow redundancy

A transport server feature that provides redundancy for messages for the entire time they're in transit.

Site resilience

A configuration that extends the messaging infrastructure to multiple Active Directory sites to provide operational continuity for the messaging system in the event of a failure affecting one of the sites.

Return to top

Database availability groups

A DAG is the base component of the high availability and site resilience framework built into Exchange 2013. A DAG is a group of up to 16 Mailbox servers that host a set of databases and provides automatic, database-level recovery from failures that affect individual databases, networks, or servers. Any server in a DAG can host a copy of a mailbox database from any other server in the DAG. When a server is added to a DAG, it works with the other servers in the DAG to provide automatic recovery from failures that affect mailbox databases, such as a disk failure or server failure. For more information about DAGs, see Database availability groups.

Return to top

Mailbox database copies

The high availability and site resilience features used first introduced in Exchange 2010 are used in Exchange 2013 to create and maintain database copies. Exchange 2013 also leverages the concept of database mobility, which is Exchange-managed database-level failovers.

Database mobility disconnects databases from servers and adds support for up to 16 copies of a single database. It also provides a native experience for creating copies of a database.

Setting a database copy as the active mailbox database is known as a *switchover*. When a failure affecting a database or access to a database occurs and a new database becomes the active copy,

this process is known as a *failover*. This process also refers to a server failure in which one or more servers bring online the databases previously online on the failed server. When either a switchover or failover occurs, other Exchange 2013 servers become aware of the switchover almost immediately and redirect client and messaging traffic to the new active database.

For example, if an active database in a DAG fails because of an underlying storage failure, Active Manager will automatically recover by failing over to a database copy on another Mailbox server in the DAG. In Exchange 2013, managed availability adds new behaviors to recover from loss of protocol access to a database, including recycling application worker pools, restarting services and servers, and initiating database failovers.

For more information about mailbox database copies, see Mailbox database copies.

Return to top

Active Manager

Exchange 2013 leverages the Active Manager component introduced in Exchange 2010 to manage the database and database copy health, status, continuous replication, and other aspects of Mailbox server high availability. For more information about Active Manager, see Active Manager.

Return to top

Site resilience

Although Exchange 2013 continues to use DAGs and Windows Failover Clustering for Mailbox server role high availability and site resilience, site resilience isn't the same in Exchange 2013. Site resilience is much better in Exchange 2013 because it has been simplified. The underlying architectural changes that were made in Exchange 2013 have significant impact on the recovery aspects of a site resilience configuration.

In Exchange 2010, mailbox (DAG) and client access (Client Access server array) recovery were tied together. If you lost all of your Client Access servers, the VIP for the array, or a significant portion of your DAG, you were in a situation where you needed to do a datacenter switchover. This is a well-documented and generally well-understood process, although it takes time to perform, and requires human intervention to begin the process.

In Exchange 2013, if you lose your Client Access server array for whatever reason (for example, the load balancer fails), you don't need to perform a datacenter switchover. With the proper configuration, failover happens at the client level and clients are automatically redirected to a second datacenter that has operating Client Access servers, and those operating Client Access servers proxy the communication back to the user's Mailbox server, which remains unaffected by the outage (because you don't do a switchover). Instead of working to recover service, the service recovers itself and you can focus on fixing the core issue (for example, replacing the failed load balancer).

Furthermore, with the namespace simplification, consolidation of server roles, de-coupling of Active Directory site server role requirements, separation of Client Access server array and DAG recovery, and load balancing changes, there are changes in Exchange 2013 that now enable both Client Access server and DAG recovery to be separate and automatic across sites, thereby providing datacenter failover scenarios, if you have three locations.

In Exchange 2010, you could deploy a DAG across two datacenters and host the witness in a third datacenter and enable failover for the Mailbox server role for either datacenter. But you didn't get failover for the solution itself, because the namespace still needed to be manually changed for the non-Mailbox server roles.

In Exchange 2013, the namespace doesn't need to move with the DAG. Exchange leverages fault tolerance built into the namespace through multiple IP addresses, load balancing (and if need be, the ability to take servers in and out of service). Modern HTTP clients work with this redundancy automatically. The HTTP stack can accept multiple IP addresses for a fully qualified domain name (FQDN), and if the first IP address it tries fails hard (that is, it can't connect), it will try the next IP address in the list. In a soft failure (connection is lost after the session is established, perhaps due to an intermittent failure in the service where, for example, a device is dropping packets and needs to be taken out of service), the user might need to refresh their browser.

This means the namespace is no longer a single point of failure as it was in Exchange 2010. In Exchange 2010, perhaps the biggest single point of failure in the messaging system is the FQDN that you give to users because it tells the user where to go. In the Exchange 2010 paradigm, changing where that FQDN goes isn't easy because you have to change DNS, and then handle DNS latency, which in some parts of the world is challenging. And you have name caches in browsers that are typically about 30 minutes or more that also have to be handled.

One of the changes in Exchange 2013 is to enable clients to have more than one place to go. Assuming the client has the ability to use more than one place to go (almost all the client access protocols in Exchange 2013 are HTTP based (examples include Outlook, Outlook Anywhere, EAS, EWS, OWA, and EAC), and all supported HTTP clients have the ability to use multiple IP addresses), thereby providing failover on the client side. You can configure DNS to hand multiple IP addresses to a client during name resolution. The client asks for mail.contoso.com and gets back two IP addresses, or four IP addresses, for example. However many IP addresses the client gets back will be used reliably by the client. This makes the client a lot better off because if one of the IP addresses fails, the client has one or more other IP addresses to try to connect to. If a client tries one and it fails, it waits about 20 seconds and then tries the next one in the list. Thus, if you lose the VIP for the Client Access server array, recovery for the clients happens automatically, and in about 21 seconds.

The benefits include the following:

- In Exchange 2010, if you lose the load balancer in your primary datacenter and you don't have another one in that site, you had to do a datacenter switchover. In Exchange 2013, if you lose the load balancer in your primary site, you simply turn it off (or maybe turn off the VIP) and repair or replace it. Clients that aren't already using the VIP in the secondary datacenter will automatically

fail over to the secondary VIP without any change of namespace, and without any change in DNS. Not only does that mean you no longer have to perform a switchover, but it also means that all of the time normally associated with a datacenter switchover recovery isn't spent. In Exchange 2010, you had to handle DNS latency (hence, the recommendation to set the Time to Live (TTL) to 5 minutes, and the introduction of the failback URL). In Exchange 2013, you don't need to do that because you get fast failover (20 seconds) of the namespace between VIPs (datacenters).

- Because you can fail over the namespace between datacenters, all that's needed to achieve a datacenter failover is a mechanism for failover of the Mailbox server role across datacenters. To get automatic failover for the DAG, you simply architect a solution where the DAG is evenly split between two datacenters, and then place the witness server in a third location so that it can be arbitrated by DAG members in either datacenter, regardless of the state of the network between the datacenters that contain the DAG members.
- In this scenario, the administrator's efforts are geared toward simply fixing the problem, and not spent restoring service. You simply fix the thing that failed; while service has been running and data integrity has been maintained. The urgency and stress level you feel when fixing a broken device is nothing like the urgency and stress you feel when you're working to restore service. It's better for the end user, and less stressful for the administrator.

You can allow failover to occur without having to perform switchbacks (sometimes mistakenly referred to as failbacks). If you lose Client Access servers in your primary datacenter and that results in a 20 second interruption for clients, you might not even care about failing back. At this point, your primary concern would be fixing the core issue (for example, replacing the failed load balancer). After it's back online and functioning, some clients will start using it, and other clients might remain operational through the second datacenter.

Exchange 2013 also provides functionality that enables administrators to deal with intermittent failures. An intermittent failure is where, for example, the initial TCP connection can be made, but nothing happens afterward. An intermittent failure requires some sort of extra administrative action to be taken because it might be the result of a replacement device being put into service. While this repair process is occurring, the device might be powered on and accepting some requests, but not really ready to service clients until the necessary configuration steps are performed. In this scenario, the administrator can perform a namespace switchover by simply removing the VIP for the device being replaced from DNS. Then during that service period, no clients will be trying to connect to it. After the replacement process has completed, the administrator can add the VIP back to DNS, and clients will eventually start using it.

For details about planning and deploying site resilience, see Planning for high availability and site resilience and Deploying high availability and site resilience.

Return to top

Third-party replication API

Exchange 2013 also includes a third-party replication API that enables organizations to use third-party synchronous replication solutions instead of the built-in continuous replication feature.

Microsoft supports third-party solutions that use this API, provided that the solution provides the necessary functionality to replace all native continuous replication functionality that's disabled as a result of using the API. Solutions are supported only when the API is used within a DAG to manage and activate mailbox database copies. Use of the API outside of these boundaries isn't supported. In addition, the solution must meet the applicable Windows hardware support requirements. (Test validation isn't required for support.)

When deploying a solution that uses the built-in third-party replication API, be aware that the solution vendor is responsible for primary support of the solution. Microsoft supports Exchange data for both replicated and non-replicated solutions. Solutions that use data replication must adhere to the Microsoft support policy for data replication, as described in Microsoft Knowledge Base article 895847, Multi-site data replication support for Exchange Server. In addition, solutions that utilize the Windows Failover Cluster resource model must meet Windows cluster supportability requirements as described in Microsoft Knowledge Base article 943984, The Microsoft Support Policy for Windows Server 2008 or Windows Server 2008 R2 Failover Clusters or The Microsoft Support Policy for Windows Server 2012 Failover Clusters.

Microsoft's backup and restore support policy for deployments that use third-party replication API-based solutions is the same as for native continuous replication deployments.

If you're a partner seeking information about the third-party API, contact your Microsoft representative.

Return to top

High availability and site resilience documentation

The following table contains links to topics that will help you learn about and manage DAGs, mailbox database copies, and backup and restore for Exchange 2013.

Topic	Description
Database availability groups	Learn about DAGs, Active Manager, Datacenter Activation Coordination (DAC) mode, and mailbox database copies.
Planning for high availability and site resilience	Learn about the general, hardware, network, software, witness server, and other requirements and best practices for DAGs.
Deploying high availability and site resilience	Explore an example deployment scenario for deploying and configuring DAGs.
Managing high availability and site resilience	Learn about DAG management tasks,

	switchovers and failovers, and maintenance mode.
Monitoring database availability groups	Learn about the built-in cmdlets and scripts for monitoring DAGs and database copies.
Backup, restore, and disaster recovery	Learn about backing up and restoring Exchange databases, recovery databases, and server recovery.

Return to top

Changes to high availability and site resilience over previous versions

Applies to: *Exchange Server 2013 SP1, Exchange Server 2013*

Topic Last Modified: *2014-07-24*

Exchange 2013 uses DAGs and mailbox database copies, along with other features such as single item recovery, retention policies, and lagged database copies, to provide high availability, site resilience, and Exchange native data protection. The high availability platform, Exchange Information Store and Extensible Storage Engine (ESE) have all been enhanced to provide greater availability and easier management, and to reduce costs. These enhancements include:

- **Reduction in IOPS over Exchange 2010** This enables you to leverage larger disks in terms of capacity and IOPS as efficiently as possible.
- **Managed availability** With managed availability, internal monitoring and recovery-oriented features are tightly integrated to help prevent failures, proactively restore services, and initiate server failovers automatically or alert administrators to take action. The focus is on monitoring and managing the end-user experience rather than just server and component uptime to help keep the service continuously available.
- **Managed Store** The Managed Store is the name of the newly rewritten Information Store processes in Exchange 2013. The new Managed Store is written in C# and tightly integrated with the Microsoft Exchange Replication service (MSExchangeRepl.exe) to provide higher availability through improved resiliency.
- **Support for multiple databases per disk** Exchange 2013 includes enhancements that enable you to support multiple databases (mixtures of active and passive copies) on the same disk,

thereby leveraging larger disks in terms of capacity and IOPS as efficiently as possible.

- **AutoReseed** Automatic reseeding capability enables you to quickly restore database redundancy after disk failure. If a disk fails, the database copy stored on that disk is copied from the active database copy to a spare disk on the same server. If multiple database copies were stored on the failed disk, they can all be automatically reseeded on a spare disk. This enables faster reseeds, as the active databases are likely to be on multiple servers and the data is copied in parallel.
- **Automatic recovery from storage failures** This feature continues the innovation introduced in Exchange 2010 to allow the system to recover from failures that affect resiliency or redundancy. In addition to the Exchange 2010 bugcheck behaviors, Exchange 2013 includes additional recovery behaviors for long I/O times, excessive memory consumption by MSExchangeRepl.exe, and severe cases where the system is in such a bad state that threads can't be scheduled.
- **Lagged copy enhancements** Lagged copies can now care for themselves to a certain extent using automatic log play down. Lagged copies will automatically play down log files in a variety of situations, such as page patching and low disk space scenarios. If the system detects that page patching is required for a lagged copy, the logs will be automatically replayed into the lagged copy to perform page patching. Lagged copies will also invoke this auto replay feature when a low disk space threshold has been reached, and when the lagged copy has been detected as the only available copy for a specific period of time. In addition, lagged copies can leverage Safety Net, making recovery or activation much easier.
- **Single copy alert enhancements** The single copy alert introduced in Exchange 2010 is no longer a separate scheduled script. It's now integrated into the managed availability components within the system and is a native function within Exchange.
- **DAG network auto-configuration** DAG networks can be automatically configured by the system based on configuration settings. In addition to manual configuration options, DAGs can also distinguish between MAPI and replication networks and configure DAG networks automatically.

Reduction in IOPS over Exchange 2010

In Exchange 2010, passive database copies have a very low checkpoint depth, which is required for fast failover. In addition, the passive copy performs aggressive pre-reading of data to keep up with a 5-megabyte (MB) checkpoint depth. As a result of using a low checkpoint depth and performing these aggressive pre-read operations, IOPS for a passive database copy was equal to IOPS for an active copy in Exchange 2010.

In Exchange 2013, the system is able to provide fast failover while using a high checkpoint depth on the passive copy (100 MB). Because passive copies have 100-MB checkpoint depth, they've been de-tuned to no longer be so aggressive. As a result of increasing the checkpoint depth and de-tuning the aggressive pre-reads, IOPS for a passive copy is about 50 percent of the active copy IOPS in Exchange 2013.

Having a higher checkpoint depth on the passive copy also results in other changes. On failover in Exchange 2010, the database cache is flushed as the database is converted from a passive copy to

an active copy. In Exchange 2013, ESE logging was rewritten so that the cache is persisted through the transition from passive to active. Because ESE doesn't need to flush the cache, you get fast failover.

One other change was made to the background database maintenance (BDM) process. BDM now processes around 1-2 MB per second per copy.

As a result of these changes, Exchange 2013 provides a significant reduction in IOPS over Exchange 2010.

Managed Availability

Managed Availability is the integration of built-in, active monitoring and the Exchange 2013 high availability platform. With Managed Availability, the system can make a determination on when to fail over a database based on service health. Managed Availability is an internal infrastructure that's deployed on the Client Access and Mailbox server roles in Exchange 2013. Managed Availability includes three main asynchronous components that are constantly doing work. The first component is the probe engine, which is responsible for taking measurements on the server and collecting data. The results of those measurements flow into the second component, the monitor. The monitor contains all of the business logic used by the system based on what is considered healthy on the data collected. Similar to a pattern recognition engine, the monitor looks for the various different patterns on all the collected measurements, and then it decides whether something is considered healthy. Finally, there is the responder engine, which is responsible for recovery actions. When something is unhealthy, the first action is to attempt to recover that component. This could include multi-stage recovery actions; for example, the first attempt may be to restart the application pool, the second may be to restart the service, the third attempt may be to restart the server, and the subsequent attempt may be to take the server offline so that it no longer accepts traffic. If the recovery actions are unsuccessful, the system escalates the issue to a human through event log notifications.

Managed availability is implemented in the form of two services:

- **Exchange Health Manager Service (MSExchangeHMHost.exe)** This is a controller process that's used to manage worker processes. It's used to build, execute, and start and stop the worker process as needed. It's also used to recover the worker process in case that process crashes, to prevent the worker process from being a single point of failure.
- **Exchange Health Manager Worker process (MSExchangeHMWorker.exe)** This is the worker process that's responsible for performing the runtime tasks.

Managed availability uses persistent storage to perform its functions:

- XML configuration files are used to initialize the work item definitions during startup of the worker process.
- The registry is used to store runtime data, such as bookmarks.
- The crimson channel event log infrastructure is used to store the work item results.

For more information about managed availability, see Managed Availability.

Managed Store

All previous versions of Exchange Server, from Exchange Server 4.0 to Exchange Server 2010, have supported running a single instance of the Information Store process (Store.exe) on the Mailbox server role. This single Store instance hosts all databases on the server: active, passive, lagged, and recovery. In the previous Exchange architectures, there is little, if any, isolation between the different databases hosted on a Mailbox server. An issue with a single mailbox database has the potential to negatively affect all other databases, and crashes resulting from a mailbox corruption can affect service for all users whose databases are hosted on that server.

Another challenge with a single Store instance in previous versions of Exchange is that the Extensible Storage Engine (ESE) scales well to 8-12 processor cores, but beyond that, cross-processor communication and cache synchronization issues lead to negative scale. Given today's much larger servers, with 16+ core systems available, this would mean impose the administrative challenge of managing the affinity of 8-12 cores for ESE and using the other cores for non-Store processes (for example, Assistants, Search Foundation, Managed Availability, etc.). Moreover, the previous architecture restricted scale-up for the Store process.

The Store.exe process has evolved considerably throughout the years as Exchange Server itself evolved, but as a single process, ultimately its scalability is limited, and it represents a single point of failure. Because of these limits, Store.exe is gone in Exchange 2013 and replaced by the Managed Store.

For more information, see Managed Store.

Multiple databases per volume

Although the storage improvements in Exchange 2013 are designed primarily for just a bunch of disks (JBOD) configurations, they're available for use by all supported storage configurations. One such feature is the ability to host multiple databases on the same volume. This feature is about Exchange optimizing for large disks. These optimizations result in a much more efficient use of large disks in terms of capacity, IOPS, and reseed times, and they're meant to address the challenges associated with running in a JBOD storage configuration:

- Database sizes must be manageable.
- Reseed operations must be fast and reliable.
- Although storage capacity is increasing, IOPS aren't.
- Disks hosting passive database copies are underutilized in terms of IOPS.
- Lagged copies have asymmetric storage requirements.
- Limited agility exists to recover from low disk space conditions.

The trend of increasing storage capacity is continuing, with 8-terabyte drives expected to be available soon. When using 8-terabyte drives in conjunction with the Exchange maximum database size best practices guidelines (2 terabytes), you would waste more than 5 terabytes of disk space. One solution would be to simply grow the databases larger, but that inhibits manageability

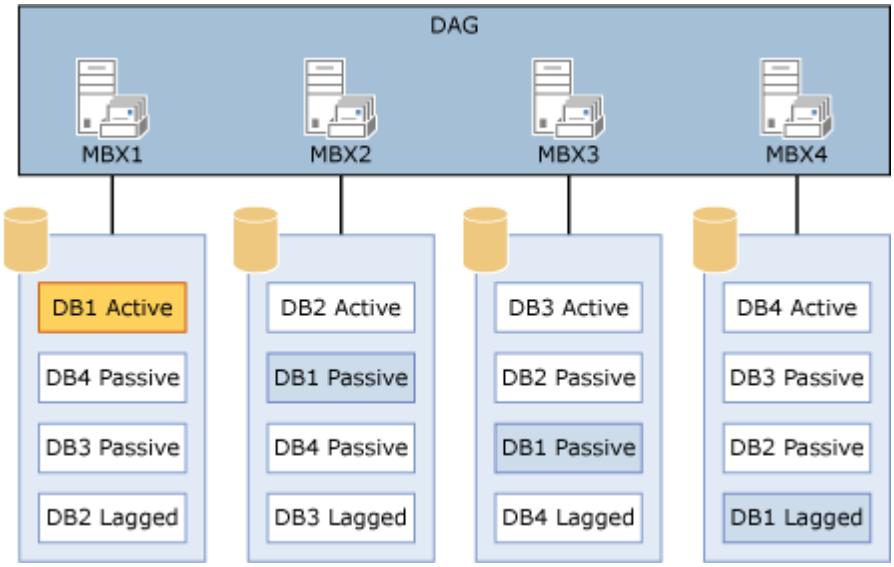
because it introduces long reseed times, including in some cases, operationally unmanageable reseed times, and the reliability of copying that amount of data over the network is compromised.

In addition, in the Exchange 2010 model, the disk storing a passive copy is underutilized in terms of IOPS. In the case of a lagged passive copy, not only is the disk underutilized in terms of IOPS, but it's also asymmetric in terms of its size, relative to the disks used to store the active and non-lagged passive copies.

Continuing a long-standing practice, Exchange 2013 is optimized so that it can use large disks (8 terabytes) in a JBOD configuration more efficiently. In Exchange 2013, with multiple databases per disk, you can have the same size disks storing multiple database copies, including lagged copies. The goal is to drive the distribution of users across the number of volumes that exist, providing you with a symmetric design where during normal operations each DAG member hosts a combination of active, passive, and optional lagged copies on the same volumes.

An example of a configuration that uses multiple databases per volume is illustrated below.

Configuration that uses multiple databases per volume



The above configuration provides a symmetrical design. All four servers have the same four databases all hosted on a single disk per server. The key is that the number of copies of each database that you have should be equal to the number of database copies per disk. In the above example, there are four copies of each database: one active copy, two passive copies, and one lagged copy. Because there are four copies of each database, the proper configuration is one that has four copies per volume. In addition, activation preference is configured so that it's balanced across the DAG and across each server. For example, the active copy will have an activation preference value of 1, the first passive copy will have an activation preference value of 2, the second passive copy will have an activation preference value of 3, and the lagged copy will have an activation preference value of 4.

In addition to having a better distribution of users across the existing volumes, another benefit of using multiple databases per disk is that it reduces the amount of time to restore data protection in the event of a failure that necessitates a reseed (for example, disk failure).

As a database gets bigger, reseeding the database takes longer. For example, a 2-terabyte

database could take 23 hours to reseed, whereas an 8-terabyte database could take as long as 93 hours (almost 4 days). Both seeds would occur at about 20 MB per second. This generally means that a very large database can't be seeded within an operationally reasonable amount of time.

In the case of a single database copy per disk scenario, the seeding operation is effectively source-bound, because it's always seeding the disk from a single source. By dividing the volume into multiple database copies, and by having the active copy of the passive databases on a specified volume stored on separate DAG members, the system is no longer source bound in the context of reseeding the disk. When a failed disk is replaced, it can be reseeded from multiple sources. This allows the system to reseed and restore data protection for these databases in a much shorter amount of time.

When using multiple databases per volume, we recommend adhering to the following best practices and requirements:

- A single logical disk partition per physical disk must be used. Don't create multiple partitions on the disk. Each database copy and its companion files (such as transaction logs and content index) should be hosted in a unique directory on the single partition.
- The number of database copies configured per volume should be equal to the number of copies of each database. For example, if you have four copies of your databases, you should use four database copies per volume.
- Database copies should have the same neighbors. (For example, they should all share the same disk on each server.)
- Activation preference across the DAG should be balanced, such that each database copy on a specified disk has a unique activation preference value.

AutoReseed

Automatic reseed, or AutoReseed, is a feature that's the replacement for what is normally administrator-driven action in response to a disk failure, database corruption event, or other issue that necessitates a reseed of a database copy. AutoReseed is designed to automatically restore database redundancy after a disk failure by using spare disks that have been provisioned on the system.

For more information, see AutoReseed. For detailed steps to configure AutoReseed, see Configure AutoReseed for a database availability group.

Automatic recovery from storage failures

Automatic recovery from storage failures continues the innovation introduced in Exchange 2010 to allow the system to recover from failures that affect resiliency or redundancy. In addition to the Exchange 2010 bugcheck behaviors, Exchange 2013 includes additional recovery behaviors for long I/O times, excessive memory consumption by the Microsoft Exchange Replication service (MSExchangeRepl.exe), and severe cases where threads can't be scheduled.

Even in JBOD environments, storage array controllers can have issues, such as crashing or hanging. Exchange 2010 included hung I/O detection and recovery features that provided enhanced resilience. These features are listed in the following table.

Name	Check	Action	Threshold
ESE Database Hung IO Detection	ESE checks for outstanding I/Os	Generates a failure item in the crimson channel to restart the server	240 seconds
Failure Item Channel Heartbeat	Ensures failure items can be written to and read from crimson channel	Replication service heartbeats crimson channel and restart server on failures	30 seconds
System Disk Heartbeat	Verifies server's system disk state	Periodically sends unbuffered I/O to system disk; restarts server on heartbeat time out	120 seconds

Exchange 2013 enhances server and storage resilience by including new behaviors for other serious conditions. These conditions and behaviors are described in the following table.

Name	Check	Action	Threshold
System bad state	No threads, including non-managed threads, can be scheduled	Restart the server	302 seconds
Long I/O times	I/O operation latency measurements	Restart the server	41 seconds
Replication service memory use	Measure the working set of MSExchangeRepl.exe	1. Log event 4395 in the crimson channel with a service termination request 2. Initiate termination of MSExchangeRepl.exe	4 gigabyte (GB)

		3. If service termination fails, restart the server	
System Event 129 (Bus reset)	Check for Event 129 in System event log	Restart the server	When event occurs
Cluster database hang	Global Update Manager updates are blocked	Restart the server	When event occurs

Lagged copy enhancements

Lagged copy enhancements include integration with Safety Net and automatic play down of log files in certain scenarios. Safety Net is a feature of transport that replaces the Exchange 2010 feature known as transport dumpster. Safety Net is similar to transport dumpster, in that it's a delivery queue that's associated with the Transport service on a Mailbox server. This queue stores copies of messages that were successfully delivered to the active mailbox database on the Mailbox server. Each active mailbox database on the Mailbox server has its own queue that stores copies of the delivered messages. You can specify how long Safety Net stores copies of the successfully delivered messages before they expire and are automatically deleted.

Safety Net takes some responsibility from shadow redundancy in DAG environments. In DAG environments, shadow redundancy doesn't need to keep another copy of the delivered message in a shadow queue while it waits for the delivered message to replicate to the passive copies of mailbox databases on the other Mailbox servers in the DAG. The copy of the delivered message is already stored in Safety Net, so shadow redundancy can redeliver the message from Safety Net if necessary.

With the introduction of Safety Net, activating a lagged database copy becomes significantly easier. For example, consider a lagged copy that has a 2-day replay lag. In that case, you would configure Safety Net for a period of 2 days. If you encounter a situation in which you need to use your lagged copy, you can suspend replication to it, and copy it twice (to preserve the lagged nature of the database and to create an extra copy in case you need it). Then, take a copy and discard all the log files, except for those in the required range. Mount the copy, which triggers an automatic request to Safety Net to redeliver the last two days of mail. With Safety Net, you don't need to hunt for where the point of corruption was introduced. You get the last two days mail, minus the data ordinarily lost on a lossy failover.

Lagged copies can now care for themselves by invoking automatic log replay to play down the log files in certain scenarios:

- When a low disk space threshold is reached
- When the lagged copy has physical corruption and needs to be page patched
- When there are fewer than three available healthy copies (active or passive only; lagged database

copies are not counted) for more than 24 hours

In Exchange 2010, page patching wasn't available for lagged copies. In Exchange 2013, page patching is available for lagged copies through this automatic play down feature. If the system detects that page patching is required for a lagged copy, the logs are automatically replayed into the lagged copy to perform page patching. Lagged copies also invoke this auto replay feature when a low disk space threshold has been reached, and when the lagged copy has been detected as the only available copy for a specific period of time.

Lagged copy play down behavior is disabled by default, and can be enabled by running the following command.

```
Set-DatabaseAvailabilityGroup <DAGName> -
ReplayLagManagerEnabled $true
```

After being enabled, play down occurs when there are fewer than three copies. You can change the default value of 3, by modifying the following DWORD registry value.

HKLM\Software\Microsoft\ExchangeServer\v15\Replay\Parameters\ReplayLagManagerNumAvailableCopies

To enable play down for low disk space thresholds, you must configure the following registry entry.

HKLM\Software\Microsoft\ExchangeServer\v15\Replay\Parameters\ReplayLagPlayDownPercentDiskFreeSpace

After configuring either of these registry settings, restart the Microsoft Exchange DAG Management service for the changes to take effect.

As an example, consider an environment where a given database has 4 copies (3 highly available copies and 1 lagged copy), and the default setting is used for *ReplayLagManagerNumAvailableCopies*. If a non-lagged copy is out-of-service for any reason (for example, it is suspended, etc.) then the lagged copy will automatically play down its log files in 24 hours.

Single copy alert enhancements

Ensuring that your servers are operating reliably and that your mailbox database copies are healthy are primary objectives of daily Exchange 2013 messaging operations. You must actively monitor the hardware, the Windows operating system, and the Exchange services. But when running in an Exchange 2013 mailbox resiliency environment, it's important that you monitor the health and status of the DAG and your mailbox database copies. It's especially vital to perform data redundancy risk management and monitor for periods in which a replicated database is down to just a single copy. This is particularly critical in environments that don't use Redundant Array of Independent Disks (RAID) and instead deploy JBOD configurations. In a RAID environment, a single disk failure doesn't affect an active mailbox database copy. However, in a JBOD environment, a single disk failure will trigger a database failover.

In Exchange 2010, the script CheckDatabaseRedundancy.ps1 was introduced. As its name implies, the purpose of the script was to monitor the redundancy of replicated mailbox databases by validating that there is at least two configured, healthy, and current copies, and to alert an administrator through event log generation when only a single healthy copy of a replicated database exists. In this case, both active and passive copies are counted when determining redundancy.

Single copy conditions include, but aren't limited to:

- Failure of an active copy to replicate to any passive copy.
- Failure of all passive copies, which includes FailedAndSuspended and Failed states in addition to healthy states where the copy is behind in log copying or replay. Note that lagged copies aren't considered behind if they're within ten minutes in replaying their logs to their lag period.
- Failure of the system to accurately know the current log generation of the active copy.

Because it's a top priority for administrators to know when they're down to a single healthy copy of a database, the CheckDatabaseRedundancy.ps1 script has been replaced with integrated, native functionality that's part of managed availability's DataProtection Health Set.

The native functionality still alerts administrators through event log notifications, and to distinguish Exchange 2013 alerts from Exchange 2010, Exchange 2013 uses the following Event IDs:

- Event 4138 (Red Alert)
- Event 4139 (Green Alert)

In Exchange 2013, the native functionality has been enhanced to reduce the level of alert noise that can occur when multiple databases on the same server enter into a single copy condition. In Exchange 2010, single copy alerts were generated on a per-database level. As a result, when there was a server-wide issue that affected multiple databases and multiple database copies, alert storms could occur. Because several failures, such as controller or memory problems, are server-wide, there was a moderately high probability that such an alert storm would occur for each server incident. In Exchange 2013, alerts are now generated on a per-server basis. When an outage affects an entire server and data redundancy becomes at risk for multiple database copies, a single alert per server is now generated.

DAG network auto-configuration

A DAG network is a collection of one or more subnets used for either replication traffic or MAPI traffic. Each DAG contains a maximum of one MAPI network and zero or more replication networks. In Exchange 2010, the initial DAG networks (for example, DAGNetwork01 and DAGNetwork02) were created by the system based on the subnets enumerated by the Cluster service. In environments where multiple networks are used and the interfaces for a specified network (for example, the MAPI network) were on the same subnet, there was little additional configuration that an administrator needed to perform. However, in environments where the interfaces for a specified network were on multiple subnets, the administrator had to perform a task referred to as collapsing DAG networks.

In Exchange 2013, collapsing DAG networks is no longer necessary. Exchange 2013 still uses the

same detection mechanisms to distinguish between the MAPI and replication networks, but it now automatically collapses DAG networks as appropriate.

In addition, by default, DAG networks are now automatically managed by the system. To view DAG network properties using the Exchange Administration Center (EAC), you must configure the DAG for manual network control by modifying the properties of the DAG using EAC, or by using the **Set-DatabaseAvailabilityGroup** cmdlet to set the *ManualDagNetworkConfiguration* parameter to `True`.

Changes to best copy selection

Best copy selection (BCS) is an internal algorithm process for finding the best copy of an individual database to activate, given a list of potential copies for activation and their health and status. Active Manager selects the best available (and unblocked) copy to become the new active database copy when the existing active database copy fails or when an administrator performs a targetless switchover. In Exchange 2010, the BCS process evaluated several aspects of each database copy to determine the best copy to activate. These included:

- Copy queue length
- Replay queue length
- Database status
- Content index status

In Exchange 2013, Active Manager performs the same BCS checks and phases to determine replication health, but it now also includes the use of a constraint of the decreasing order of health states. As a result of these changes, BCS is now called best copy and server selection (BCSS).

BCSS includes several new health checks that are part of the built in managed availability monitoring components in Exchange 2013. There are four new additional checks performed by Active Manager (listed in the order in which they're performed):

1. **All Healthy** Checks for a server hosting a copy of the affected database that has all monitoring components in a healthy state.
2. **Up to Normal Healthy** Checks for a server hosting a copy of the affected database that has all monitoring components with Normal priority in a healthy state.
3. **All Better than Source** Checks for a server hosting a copy of the affected database that has monitoring components in a state that's better than the current server hosting the affected copy.
4. **Same as Source** Checks for a server hosting a copy of the affected database that has monitoring components in a state that's the same as the current server hosting the affected copy.

If BCSS is invoked as a result of a failover that's triggered by a managed availability monitoring component (for example, via a Failover responder), an additional mandatory constraint is enforced where the target server's component health must be better than the server on which the failover occurred. For example, if a failure of Microsoft Office Outlook Web App triggers a managed availability failover via a Failover responder, BCSS must select a server hosting a copy of the affected database on which Outlook Web App is healthy.

DAG Management Service

Cumulative Update 2 (CU2) for the Release to Manufacturing (RTM) version of Exchange 2013 contains a new service on Mailbox servers that are members of a DAG. This service is called the Microsoft Exchange DAG Management Service (MSExchangeDAGMgmt). This new service contains internal DAG monitoring functionality that previously executed inside the Microsoft Exchange Replication service (MSExchangeRepl).

DAGs without a cluster administrative access point

All DAGs running Windows Server 2008 R2 or Windows Server 2012 require at least one IP address on every subnet included in the MAPI network. The IP address(es) assigned to the DAG are used by the DAG's cluster with the cluster's administrative access point (also known as the cluster network name) to enable name resolution and connectivity to the cluster (or more precisely, connectivity to the cluster member that currently owns the cluster core resource group) using the cluster name. Windows Server 2012 R2 enables you to create a failover cluster without an administrative access point. Windows failover clusters without administrative access points have the following characteristics:

- There is no IP address assigned to the cluster, and therefore no IP Address Resource in the cluster core resource group.
- There is no network name assigned to the cluster, and therefore no Network Name Resource in the cluster core resource group
- The name of the cluster is not registered in DNS, and it is not resolvable on the network.
- A cluster name object (CNO) is not created in Active Directory.
- The Windows failover cluster cannot be managed using the Failover Cluster Management tool. It must be managed using Windows PowerShell, and the PowerShell cmdlets must be run against individual cluster members.

When running on Windows Server 2012 R2 or later, Service Pack 1 (SP1) for Exchange 2013 and later enable you to create a DAG without a cluster administrative access point. You can create a DAG without an administrative access point using the Exchange Admin Center or by using the Shell. For more information, see Creating DAGs and Create a database availability group.

Database availability groups

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-18*

A database availability group (DAG) is the base component of the Mailbox server high availability and site resilience framework built into Microsoft Exchange Server 2013. A DAG is a group of up to 16 Mailbox servers that hosts a set of databases and provides automatic database-level recovery from failures that affect individual servers or databases.

A DAG is a boundary for mailbox database replication, database and server switchovers and failovers, and an internal component called *Active Manager*. Active Manager, which runs on every Mailbox server, manages switchovers and failovers within DAGs. For more information about Active Manager, see Active Manager.

Any server in a DAG can host a copy of a mailbox database from any other server in the DAG. When a server is added to a DAG, it works with the other servers in the DAG to provide automatic recovery from failures that affect mailbox databases, such as a disk, server, or network failure.

Contents

Database availability group lifecycle

DAGs leverage the concept of *incremental deployment*, which is the ability to deploy service and data availability for all Mailbox servers and databases after Exchange is installed. After you deploy Exchange 2013 Mailbox servers, you can create a DAG, add Mailbox servers to the DAG, and then replicate mailbox databases between the DAG members.

Note:

It's supported to create a DAG that contains a combination of physical Mailbox servers and virtualized Mailbox servers, provided that the servers and solution comply with the Exchange 2013 system requirements and the requirements set forth in Exchange 2013 virtualization. As with all Exchange high availability configurations, you must ensure that all Mailbox servers in the DAG are sized appropriately to handle the necessary workload during scheduled and unscheduled outages.

A DAG is created by using the New-DatabaseAvailabilityGroup cmdlet. A DAG is initially created as an empty object in Active Directory. This directory object is used to store relevant information about the DAG, such as server membership information and some DAG configuration settings. When you add the first server to a DAG, a failover cluster is automatically created for the DAG. This failover cluster is used exclusively by the DAG, and the cluster must be dedicated to the DAG. Use of the cluster for any other purpose isn't supported.

In addition to a failover cluster being created, the infrastructure that monitors the servers for network or server failures is initiated. The failover cluster heartbeat mechanism and cluster database are then used to track and manage information about the DAG that can change quickly, such as database mount status, replication status, and last mounted location.

During creation, the DAG is given a unique name, and either assigned one or more static IP addresses or configured to use Dynamic Host Configuration Protocol (DHCP), or created without a cluster administrative access point. DAGs without an administrative access point can be created only on servers running Exchange 2013 Service Pack 1 or later on Windows Server 2012 R2 Standard or Datacenter edition. DAGs without cluster administrative access points have the following characteristics:

- There is no IP address assigned to the cluster/DAG, and therefore no IP Address Resource in the cluster core resource group.
- There is no network name assigned to the cluster, and therefore no Network Name Resource in the cluster core resource group
- The name of the cluster/DAG is not registered in DNS, and it is not resolvable on the network.
- A cluster name object (CNO) is not created in Active Directory.
- The cluster cannot be managed using the Failover Cluster Management tool. It must be managed using Windows PowerShell, and the PowerShell cmdlets must be run against individual cluster members.

This example shows how to use the Shell to create a DAG with a cluster administrative access point that will have three servers. Two servers (EX1 and EX2) are on the same subnet (10.0.0.0), and the third server (EX3) is on a different subnet (192.168.0.0).

```
New-DatabaseAvailabilityGroup -Name DAG1 -WitnessServer EX4
-DatabaseAvailabilityGroupIPAddresses 10.0.0.5,192.168.0.5
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer EX1
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer EX2
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer EX3
```

The commands to create a DAG without a cluster administrative access point are very similar:

```
New-DatabaseAvailabilityGroup -Name DAG1 -WitnessServer EX4
-DatabaseAvailabilityGroupIPAddresses
([System.Net.IPAddress])::None
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer EX1
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer EX2
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer EX3
```

The cluster for DAG1 is created when EX1 is added to the DAG. During cluster creation, the **Add-**

DatabaseAvailabilityGroupServer cmdlet retrieves the IP addresses configured for the DAG and ignores the ones that don't match any of the subnets found on EX1. In the first example above, the cluster for DAG1 is created with an IP address of 10.0.0.5, and 192.168.0.5 is ignored. In the second example above, the value of the *DatabaseAvailabilityGroupIPAddresses* parameter instructs the task to create a failover cluster for the DAG that does not have an administrative access point. Thus, the cluster is created with an IP address or network name resource in the core cluster resource group.

Then, EX2 is added, and the **Add-DatabaseAvailabilityGroupServer** cmdlet again retrieves the IP addresses configured for the DAG. There are no changes to the cluster's IP addresses because in EX2 is on the same subnet as EX1.

Then, EX3 is added, and the **Add-DatabaseAvailabilityGroupServer** cmdlet again retrieves the IP addresses configured for the DAG. Because a subnet matching 192.168.0.5 is present on EX3, the 192.168.0.5 address is added as an IP address resource in the cluster group. In addition, an **OR** dependency for the Network Name resource for each IP address resource is automatically configured. The 192.168.0.5 address will be used by the cluster when the cluster core resource group moves to EX3.

For DAGs with cluster administrative access points, Windows failover clustering registers the IP addresses for the cluster in the Domain Name System (DNS) when the Network Name resource is brought online. In addition, when EX1 is added to the cluster, a cluster name object (CNO) is created in Active Directory. The network name, IP address(es), and CNO for the cluster are not used for DAG functions. Administrators and end users don't need to interface with or connect to the cluster/DAG name or IP address for any reason. Some third party applications connect to the cluster administrative access point to perform management tasks, such as backup or monitoring. If you do not use any third party applications that require a cluster administrative access point, and your DAG is running Exchange 2013 SP1 or later on Windows Server 2012 R2, then we recommend creating a DAG without an administrative access point. This simplifies DAG configuration, eliminates the need for one or more IP addresses, and reduces the attack surface of a DAG.

DAGs are also configured to use a witness server and a witness directory. The witness server and witness directory are either automatically configured by the system, or they can be manually configured by the administrator. In the examples above, EX4 (a server that is not and will not be a member of the DAG) is being manually configured as the DAG’s witness server.

By default, a DAG is designed to use the built-in continuous replication feature to replicate mailbox databases among servers in the DAG. If you're using third-party data replication that supports the Third Party Replication API in Exchange 2013, you must create the DAG in third-party replication mode by using the New-DatabaseAvailabilityGroup cmdlet with the *ThirdPartyReplication* parameter. After this mode is enabled, it can't be disabled.

After the DAG is created, Mailbox servers can be added to the DAG. When the first server is added to the DAG, a cluster is formed for use by the DAG. DAGs make use of Windows failover clustering technology, such as the cluster heartbeat, cluster networks, and the cluster database (for storing data that changes, such as database state changes from active to passive or vice versa, or from mounted to dismounted and vice versa). As each subsequent server is added to the DAG, it's joined

to the underlying cluster, the cluster's quorum model is automatically adjusted by Exchange, and the server is added to the DAG object in Active Directory.

After Mailbox servers are added to a DAG, you can configure a variety of DAG properties, such as whether to use network encryption or network compression for database replication within the DAG. You can also configure DAG networks and create additional DAG networks.

After you add members to a DAG and configure the DAG, the active mailbox databases on each server can be replicated to the other DAG members. After you create mailbox database copies, you can monitor the health and status of the copies using a variety of built-in monitoring tools. In addition, you can perform database and server switchovers.

For more information about creating DAGs, managing DAG membership, configuring DAG properties, creating and monitoring mailbox database copies, and performing switchovers, see Managing high availability and site resilience.

Database availability group quorum models

Underneath every DAG is a Windows failover cluster. Failover clusters use the concept of quorum, which uses a consensus of voters to ensure that only one subset of the cluster members (which could mean all members or a majority of members) is functioning at one time. Quorum isn't a new concept for Exchange 2013. Highly available Mailbox servers in previous versions of Exchange also use failover clustering and its concept of quorum. Quorum represents a shared view of members and resources, and the term quorum is also used to describe the physical data that represents the configuration within the cluster that's shared between all cluster members. As a result, all DAGs require their underlying failover cluster to have quorum. If the cluster loses quorum, all DAG operations terminate and all mounted databases hosted in the DAG dismount. In this event, administrator intervention is required to correct the quorum problem and restore DAG operations.

Quorum is important to ensure consistency, to act as a tie-breaker to avoid partitioning, and to ensure cluster responsiveness:

- **Ensuring consistency** A primary requirement for a Windows failover cluster is that each of the members always has a view of the cluster that's consistent with the other members. The cluster hive acts as the definitive repository for all configuration information relating to the cluster. If the cluster hive can't be loaded locally on a DAG member, the Cluster service doesn't start, because it isn't able to guarantee that the member meets the requirement of having a view of the cluster that's consistent with the other members.
- **Acting as a tie-breaker** A quorum witness resource is used in DAGs with an even number of members to avoid split brain syndrome scenarios and to make sure that only one collection of the members in the DAG is considered official. When the witness server is needed for quorum, any member of the DAG that can communicate with the witness server can place a Server Message Block (SMB) lock on the witness server's witness.log file. The DAG member that locks the witness server (referred to as the *locking node*) retains an additional vote for quorum purposes. The DAG members in contact with the locking node are in the majority and maintain quorum. Any

DAG members that can't contact the locking node are in the minority and therefore lose quorum.

- **Ensuring responsiveness** To ensure responsiveness, the quorum model makes sure that, whenever the cluster is running, enough members of the distributed system are operational and communicative, and at least one replica of the cluster's current state can be guaranteed. No additional time is required to bring members into communication or to determine whether a specific replica is guaranteed.

DAGs with an even number of members use the failover cluster's Node and File Share Majority quorum mode, which employs an external witness server that acts as a tie-breaker. In this quorum mode, each DAG member gets a vote. In addition, the witness server is used to provide one DAG member with a weighted vote (for example, it gets two votes instead of one). The cluster quorum data is stored by default on the system disk of each member of the DAG, and is kept consistent across those disks. However, a copy of the quorum data isn't stored on the witness server. A file on the witness server is used to keep track of which member has the most updated copy of the data, but the witness server doesn't have a copy of the cluster quorum data. In this mode, a majority of the voters (the DAG members plus the witness server) must be operational and able to communicate with each other to maintain quorum. If a majority of the voters can't communicate with each other, the DAG's underlying cluster loses quorum, and the DAG will require administrator intervention to become operational again.

DAGs with an odd number of members use the failover cluster's Node Majority quorum mode. In this mode, each member gets a vote, and each member's local system disk is used to store the cluster quorum data. If the configuration of the DAG changes, that change is reflected across the different disks. The change is only considered to have been committed and made persistent if that change is made to the disks on half the members (rounding down) plus one. For example, in a five-member DAG, the change must be made on two plus one members, or three members total.

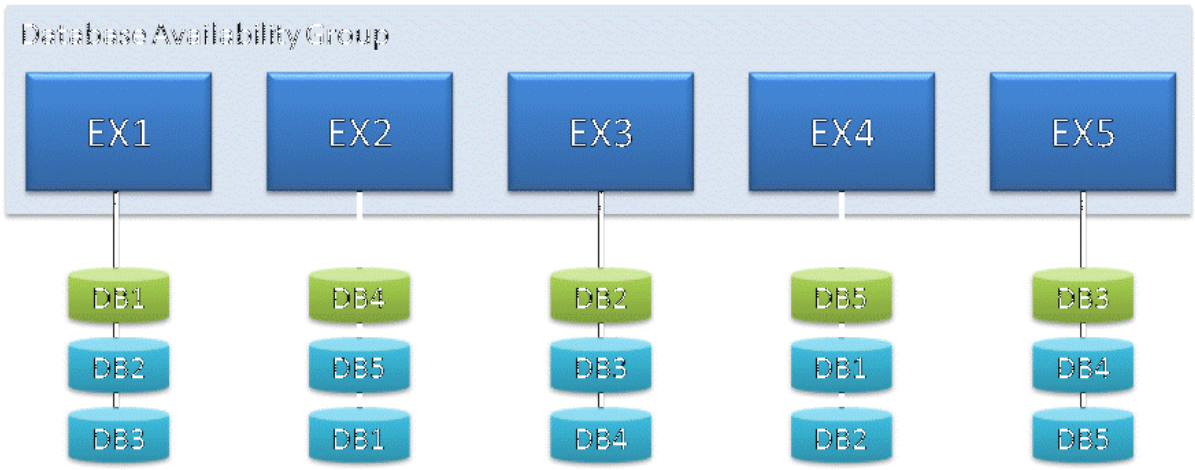
Quorum requires a majority of voters to be able to communicate with each other. Consider a DAG that has four members. Because this DAG has an even number of members, an external witness server is used to provide one of the cluster members with a fifth, tie-breaking vote. To maintain a majority of voters (and therefore quorum), at least three voters must be able to communicate with each other. At any time, a maximum of two voters can be offline without disrupting service and data access. If three or more voters are offline, the DAG loses quorum, and service and data access will be disrupted until you resolve the problem.

Return to top

Using a database availability group for high availability

To illustrate how a DAG can provide high availability for your mailbox databases, consider the following example, which uses a DAG with five members. This DAG is illustrated in the following figure.

DAG with five members

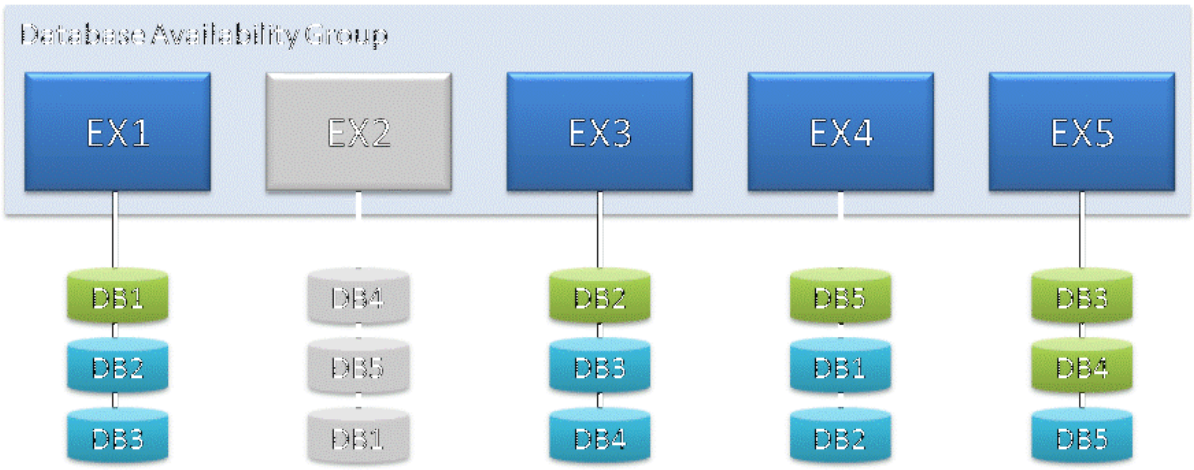


In the preceding figure, the green databases are active mailbox database copies and the blue databases are passive mailbox database copies. In this example, the database copies aren't mirrored across each server, but rather spread across multiple servers. This ensures that no two servers in the DAG have the same set of database copies, providing the DAG with greater resilience to failures, including failures that occur while other components are unavailable as a result of regular maintenance.

Consider the following scenario, using the preceding example DAG, which illustrates resilience to multiple database and server failures.

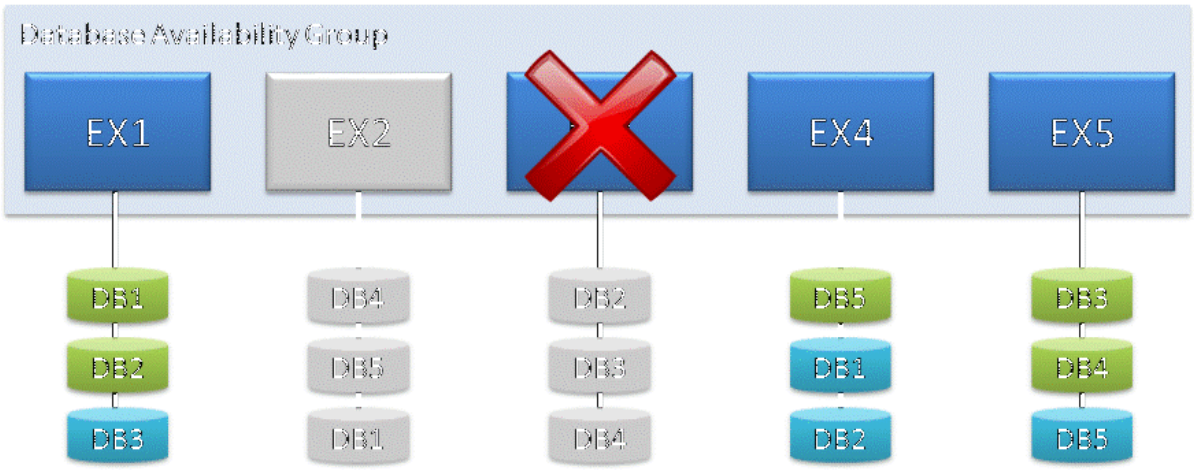
Initially, all databases and servers are healthy. You need to install some operating system updates on EX2, so you put the server into maintenance mode. This causes a server switchover, which activates the copy of DB4 on another Mailbox server. A server switchover moves all active mailbox database copies from their current server to one or more other Mailbox servers in the DAG in preparation for a scheduled outage for the current server. In this example, there's only one active mailbox database on EX2 (DB4), so only one active mailbox database copy is moved.

DAG with a server offline for maintenance



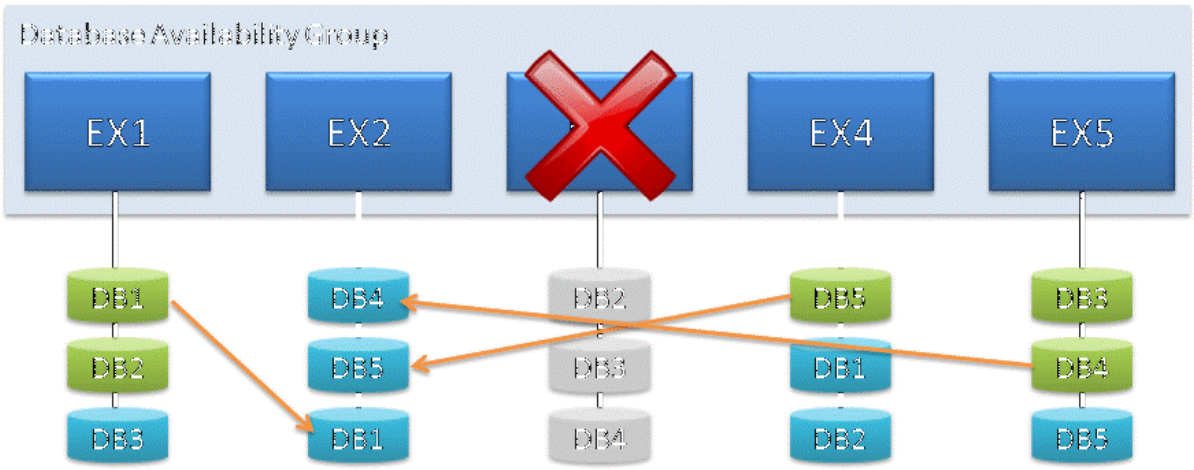
While you perform maintenance on EX2, EX3 experiences a catastrophic hardware failure and goes offline. Prior to going offline, EX3 hosted the active copy of DB2. To recover from the failure, the system automatically activates the copy of DB2 that's hosted on EX1 within 30 seconds. This is illustrated in the following figure.

DAG with a server offline for maintenance and a failed server



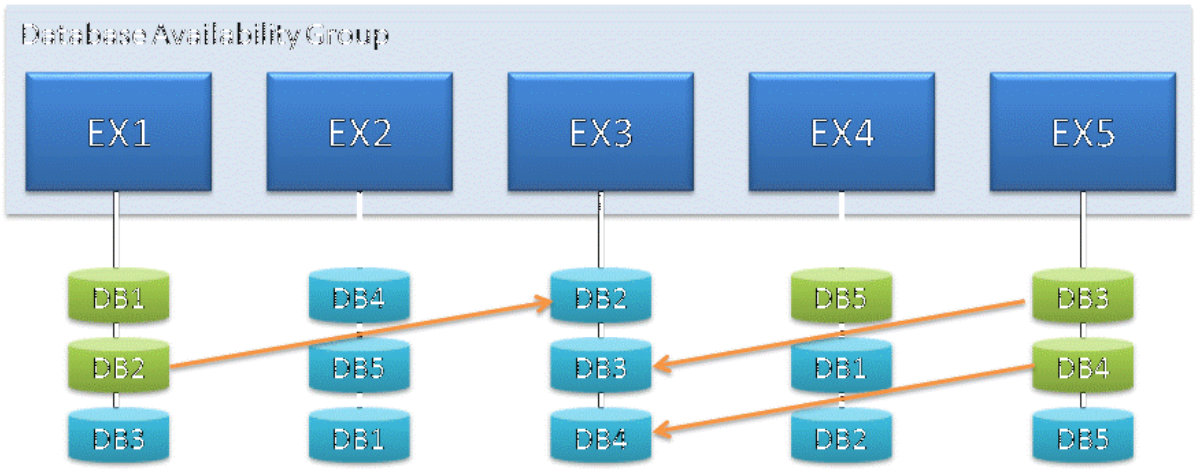
After the scheduled maintenance is completed for EX2, you bring the server online and take it out of maintenance mode. As soon as EX2 is available, the other members of the DAG are notified, and the copies of DB1, DB4, and DB5 hosted on EX2 are automatically synchronized with the active copy of each database. This is illustrated in the following figure.

DAG with a restored server synchronizing its database copies



After the failed hardware component in EX3 is replaced with a new component, EX3 is brought online. After EX3 is available, the other members of the DAG are notified, and the copies of DB2, DB3, and DB4 hosted on EX3 are automatically synchronized with the active copy of each database. This is illustrated in the following figure.

DAG with a repaired server synchronizing its database copies

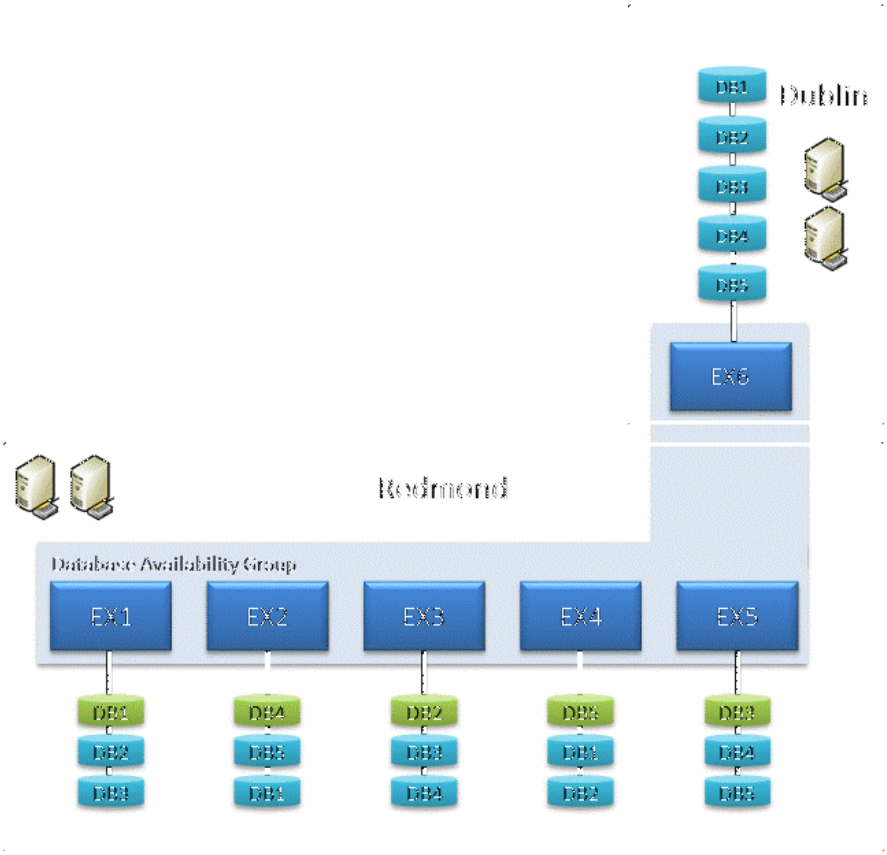


Return to top

Using a database availability group for site resilience

In addition to providing high availability within a datacenter, a DAG can also be extended to one or more datacenters in a configuration that provides site resilience for one or multiple datacenters. In the preceding example figures, the DAG is located in a single datacenter and single Active Directory site. Incremental deployment can be used to extend this DAG to a second datacenter (and a second Active Directory site) by deploying a Mailbox server and the necessary supporting resources (one or more Active Directory servers, and DNS services). The Mailbox server is then added to the DAG, as illustrated in the following figure.

DAG extended across two Active Directory sites



In this example, a passive copy of each active database in the Redmond datacenter is configured on EX6 in the Dublin datacenter. However, there are many other examples of DAG configurations that provide site resilience. For example:

- Instead of hosting only passive database copies, EX6 could host all active copies, or it could host a mixture of active and passive copies.
- In addition to EX6, multiple DAG members could be deployed in the Dublin datacenter, providing protection against additional failures. This configuration also provides additional capacity, so that if the Redmond datacenter fails, the Dublin datacenter can support a much larger user population.

Using multiple database availability groups for site resilience

In the preceding example, a single DAG extends across multiple datacenters, providing site resilience for either or both datacenters. When using a single DAG to provide site resilience in an environment where each datacenter to which you extend the DAG has an active user population, there is a single point of failure in the wide area network (WAN) connection. This is because quorum requires a majority of the voters to be active and able to communicate with each other.

In the preceding example, the majority of voters are located in the Redmond datacenter. If the Dublin datacenter hosts active mailbox databases, and it has a local user population, a WAN

outage would result in a messaging service outage for the Dublin users. When WAN connectivity breaks, only the DAG members in the Redmond datacenter retain quorum and continue providing messaging service.

To eliminate the WAN as a single point of failure when you need to provide site resilience for multiple datacenters that each have an active user population, you should deploy multiple DAGs, where each DAG has a majority of voters in a separate datacenter. When a WAN outage occurs, replication will be blocked until connectivity is restored. Users will have messaging service, because each DAG continues to service its local user population.

Return to top

Active Manager

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-21*

Microsoft Exchange Server 2013 includes a component called *Active Manager* that manages the high availability platform that includes the database availability group (DAG) and mailbox database copies. Active Manager runs inside the Microsoft Exchange Replication service (MSExchangeRepl.exe) on all Mailbox servers. On Mailbox servers that aren't members of a DAG, there is a single Active Manager role: *Standalone Active Manager*. On servers that are members of a DAG, there are two Active Manager roles: *Primary Active Manager* (PAM) and *Standby Active Manager* (SAM). PAM is the Active Manager role in a DAG that decides which copies will be active and passive. PAM is responsible for getting topology change notifications and reacting to server failures. The DAG member that holds the PAM role is always the member that currently owns the cluster quorum resource (default cluster group). If the server that owns the cluster quorum resource fails, the PAM role automatically moves to a surviving server that takes ownership of the cluster quorum resource. In addition, if you need to take the server that hosts the cluster quorum resource offline for maintenance or an upgrade, you must first move the PAM to another server in the DAG. The PAM controls all movement of the active designations between a database's copies. (Only one copy can be active at any specified time, and that copy may be mounted or dismounted.) The PAM also performs the functions of the SAM role on the local system (detecting local database and local Information Store failures).

The SAM provides information on which server hosts the active copy of a mailbox database to other components of Exchange that are running an Active Manager client component (for example, Client Access or Transport services). The SAM detects failures of local databases and the local Information Store. It reacts to failures by asking the PAM to initiate a failover (if the database is replicated). A SAM doesn't determine the target of failover, nor does it update a database's

location state in the PAM. It will access the active database copy location state to answer queries for the active copy of the database that it receives.

Note:
Exchange 2013 isn't a clustered application. Instead, it uses the cluster library functions implemented in clusapi.dll for cluster, group, cluster network (heartbeating), node management, cluster registry, and a few control code functions. In addition, Active Manager stores current mailbox database information (for example, active and passive data, and mounted data) in the cluster database (also known as the cluster registry). Although the information is stored directly in the cluster database, it isn't accessed directly by any other components.

In Exchange 2013, the Microsoft Exchange Replication service periodically monitors the health of all mounted databases. In addition, it also monitors the Extensible Storage Engine (ESE) for any I/O errors or failures. When the service detects a failure, it notifies Active Manager. Active Manager then determines which database copy should be mounted and what it requires to mount that database. In addition, it tracks the active copy of a mailbox database (based on the last mounted copy of the database) and provides the tracking results information to the Client Access server to which the client is connected.

Best Copy Selection

When a failure occurs that prevents access to the active copy of a replicated mailbox database, Active Manager takes several steps to recover from the failure by selecting the best possible passive copy of the affected database to activate. This process was known as best copy selection (BCS) in Exchange 2010, and it's now known as best copy and server selection (BCSS) in Exchange 2013. The general process occurs in the following order:

1. Managed availability or Active Manager detects a failure, or an administrator initiates a targetless switchover.
2. The PAM runs the BCSS internal algorithm.
3. A process called *attempt copy last logs* (ACLL) occurs, which tries to copy any missing log files from the server that hosted the active database copy prior to the failure or switchover.
4. After the ACLL process has completed, the value of the *AutoDatabaseMountDial* for the Mailbox servers hosting copies of the database is compared with the copy queue length of the database being activated. At this point, either:
   - The number of missing log files is equal to or less than the value of *AutoDatabaseMountDial*, in which case Step 5 occurs.
   - The number of missing log files is greater than the value of *AutoDatabaseMountDial*, in which case Active Manager will try to activate next best available copy, if there is one.
5. The PAM issues a mount request to the Microsoft Exchange Information Store via remote procedure call (RPC). At this point, either:
   - The database mounts and is made available to clients.
   - The database doesn't mount, and PAM performs steps 3 and 4 on the next best copy (if one is available).

In Exchange 2010, the BCS process evaluated several aspects of each database copy to determine the best copy to activate. These included:

- Copy queue length
- Replay queue length
- Database status
- Content index status

In Exchange 2013, Active Manager runs through all of the same BCS checks and phases, but it now also includes the use of a constraint of the decreasing order of health states. Specifically, BCSS includes several new health checks that are part of the built in managed availability monitoring components in Exchange 2013. There are four new additional checks performed by Active Manager (listed in the order in which they are performed):

1. **All Healthy** Checks for a server hosting a copy of the affected database that has all monitoring components in a healthy state.
2. **Up to Normal Healthy** Checks for a server hosting a copy of the affected database that has all monitoring components with Normal priority in a healthy state.
3. **All Better than Source** Checks for a server hosting a copy of the affected database that has monitoring components in a state that's better than the current server hosting the affected copy.
4. **Same as Source** Checks for a server hosting a copy of the affected database that has monitoring components in a state that's the same as the current server hosting the affected copy.

If BCSS is invoked as a result of a failover that's triggered by a monitoring component (for example, via a Failover responder), an additional mandatory constraint is enforced where the target server's component health must be better than the server on which the failover occurred. For example, if a failure of Microsoft Office Outlook Web App triggers a failover via a Failover responder, BCSS must select a server hosting a copy of the affected database on which Outlook Web App is healthy.

Best copy selection process

With respect to database failures (not protocol failures), Active Manager in Exchange 2013 performs the same checks as it did in Exchange 2010. Active Manager begins the best copy selection process by creating a list of database copies that are potential candidates for activation. Any database copies that are unreachable or are administratively blocked from activation are ignored and not used during the selection process. The order of the list depends on the value of the *AutoDatabaseMountDial*:

- If the *AutoDatabaseMountDial* is configured with any value other than `Lossless` on all servers that host a copy of the database, Active Manager sorts the resulting list using the copy queue length as the primary key. The calculation is based on LastLogInspected (from the copy's point of view), so the list of potential copies is sorted by the highest value for LastLogInspected (which will be the copy with the lowest copy queue length). If necessary, Active Manager sorts the list a second time, using the value for activation preference as a secondary key to break any tie conditions where two or more passive copies have the same copy queue length. The copy with the lowest activation preference value has the higher priority on the list.

- If the *AutoDatabaseMountDial* is configured with a value of `Lossless` on any server that hosts a copy of the database, Active Manager sorts the resulting list in ascending order by using the value for activation preference as the primary key. In addition, when an administrator performs a lossless server or database switchover without specifying a target, Active Manager also sorts the resulting list in ascending order by using the value for activation preference as the primary key.

Next, Active Manager attempts to locate a mailbox database copy on the list that has a status of Healthy, DisconnectedAndHealthy, DisconnectedAndResynchronizing, or SeedingSource, and then evaluates the activation potential of each of the copies on the list by using an order set of ten criteria. Active Manager determines if any of the candidates for activation meet the first set of criteria:

- It has a content index with a status of Healthy.
- It has a copy queue length less than 10 log files.
- It has a replay queue length less than 50 log files.

If none of the database copies meet the first set of criteria, Active Manager tries to locate a database copy that meets the second set of criteria:

- It has a content index with a status of Crawling.
- It has a copy queue length less than 10 log files.
- It has a replay queue length less than 50 log files.

If none of the database copies meet the second set of criteria, Active Manager tries to locate a database copy that meets the third set of criteria:

- It has a content index with a status of Healthy.
- It has a replay queue length less than 50 log files.

If none of the database copies meet the third set of criteria, Active Manager tries to locate a database copy that meets the fourth set of criteria:

- It has a content index with a status of Crawling.
- It has a replay queue length less than 50 log files.

If none of the database copies meet the fourth set of criteria, Active Manager tries to locate a database copy that meets the fifth set of criteria:

- It has a replay queue length less than 50 log files.

If none of the database copies meet the fifth set of criteria, Active Manager tries to locate a database copy that meets the sixth set of criteria:

- It has a content index with a status of Healthy.
- It has a copy queue length less than 10 log files.

If none of the database copies meet the sixth criteria, Active Manager tries to locate a database copy that meets the seventh set of criteria:

- It has a content index with a status of Crawling.
- It has a copy queue length less than 10 log files.

If none of the database copies meet the seventh set of criteria, Active Manager tries to locate a database copy that meets the eighth set of criteria:

- It has a content index with a status of Healthy.

If none of the database copies meet all of the eighth set of criteria, Active Manager tries to locate a database copy that meets the ninth set of criteria:

- It has a content index with a status of Crawling.

If none of the database copies meet the ninth set of criteria, Active Manager tries to activate any database copy with a status of Healthy, DisconnectedAndHealthy, DisconnectedAndResynchronizing, or SeedingSource (the tenth set of criteria). If it can't find any database copies that meet the tenth set of criteria, it isn't able to automatically activate a database copy.

After one or more copies are located that meet one or more sets of criteria, the ACLL process runs to copy any log files from the original source to the potential new active copy. After the ACLL process has completed, the PAM issues a mount request and either the database mounts and is made available to clients or the database doesn't mount and the PAM searches for the next best copy (if one is available).

Datacenter Activation Coordination mode

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-14*

Datacenter Activation Coordination (DAC) mode is a property of a database availability group (DAG). DAC mode is disabled by default but should be enabled for all DAGs with two or more members that use continuous replication. DAC mode shouldn't be enabled for DAGs that use third-party replication mode unless specified by the third-party vendor.

DAC mode is used to control the database mount on startup behavior of a DAG. This control is designed to prevent split brain from occurring at the database level during a datacenter switchback. Split brain, also known as split brain syndrome, is a condition that results in a database copying being mounted as an active copy on two members of the same DAG that are unable to communicate. Split brain is prevented using DAC mode because DAC mode requires DAG members to obtain permission to mount databases before they can be mounted.

For example, consider a scenario where a primary datacenter contains two DAG members and the witness server, and a second datacenter contains two other DAG members. In this scenario, the DAG is not in DAC mode. The primary datacenter loses power, so you activate the DAG in the second datacenter. Eventually power to the primary datacenter is restored, and the DAG members in the primary datacenter, which had quorum before the power failure, will start up and mount their databases. Because the primary datacenter was restored without network connectivity to the

second datacenter, and because the DAG was not in DAC mode, the active databases within the DAG entered a split brain condition.

How DAC mode works

DAC mode is designed to prevent split brain from occurring by including a protocol called Datacenter Activation Coordination Protocol (DACP). When DAC mode is enabled, DAG members won't automatically mount databases even if they have quorum. Instead DACP is used to determine the current state of the DAG and whether Active Manager should attempt to mount the databases.

You might think of DAC mode as an application level of quorum for mounting databases. To understand the purpose of DACP and how it works, it's important to understand the primary scenario it's intended to handle. Consider the two-datacenter scenario. Suppose there is a complete power failure in the primary datacenter. In this event, all of the servers and the WAN are down, so the organization makes the decision to activate the standby datacenter. In almost all such recovery scenarios, when power is restored to the primary datacenter, WAN connectivity is typically not immediately restored. This means that the DAG members in the primary datacenter will power up, but they won't be able to communicate with the DAG members in the activated standby datacenter. The primary datacenter should always contain the majority of the DAG quorum voters, which means that when power is restored, even in the absence of WAN connectivity to the DAG members in the standby datacenter, the DAG members in the primary datacenter have a majority and therefore have quorum. This is a problem because with quorum, these servers may be able to mount their databases, which in turn would cause divergence from the actual active databases that are now mounted in the activated standby datacenter.

DACP was created to address this issue. Active Manager stores a bit in memory (either a 0 or a 1) that tells the DAG whether it's allowed to mount local databases that are assigned as active on the server. When a DAG is running in DAC mode, each time Active Manager starts up the bit is set to 0, meaning it isn't allowed to mount databases. Because it's in DAC mode, the server must try to communicate with all other members of the DAG that it knows to get another DAG member to give it an answer as to whether it can mount local databases that are assigned as active to it. The answer comes in the form of the bit setting for other Active Managers in the DAG. If another server responds that its bit is set to 1, it means servers are allowed to mount databases, so the server starting up sets its bit to 1 and mounts its databases.

But when you recover from a primary datacenter power outage where the servers are recovered but WAN connectivity has not been restored, all of the DAG members in the primary datacenter will have a DACP bit value of 0; and therefore none of the servers starting back up in the recovered primary datacenter will mount databases, because none of them can communicate with a DAG member that has a DACP bit value of 1.

DAC mode for DAGs with two members

DAGs with two members have inherent limitations that prevent the DACP bit alone from fully

protecting against application-level split brain syndrome. For DAGs with only two members, DAC mode also uses the boot time of the DAG's witness server to determine whether it can mount databases on startup. The boot time of the witness server is compared to the time when the DACP bit was set to 1.

- If the time the DACP bit was set is earlier than the boot time of the witness server, the system assumes that the DAG member and witness server were rebooted at the same time (perhaps because of power loss in the primary datacenter), and the DAG member isn't permitted to mount databases.
- If the time that the DACP bit was set is more recent than the boot time of the witness server, the system assumes that the DAG member was rebooted for some other reason (perhaps a scheduled outage in which maintenance was performed or perhaps a system crash or power loss isolated to the DAG member), and the DAG member is permitted to mount databases.

Important:
Because the witness server's boot time is used to determine whether a DAG member can mount its active databases on startup, you should never restart the witness server and the sole DAG member at the same time. Doing so may leave the DAG member in a state where it can't mount databases on startup. If this happens, you must run the Restore-DatabaseAvailabilityGroup cmdlet on the DAG. This resets the DACP bit and permits the DAG member to mount databases.

Other benefits of DAC mode

In addition to preventing split brain syndrome at the application level, DAC mode also enables the use of the built-in site resilience cmdlets used to perform datacenter switchovers. These include the following:

- Stop-DatabaseAvailabilityGroup
- Restore-DatabaseAvailabilityGroup
- Start-DatabaseAvailabilityGroup

Performing a datacenter switchover for DAGs that aren't in DAC mode involves using a combination of Exchange tools and cluster management tools. For more information, see Datacenter Switchovers.

Enabling DAC mode

DAC mode can be enabled only by using the Exchange Management Shell. Specifically, you can use the Set-DatabaseAvailabilityGroup cmdlet to enable DAC mode, as illustrated in the following example.

```
Set-DatabaseAvailabilityGroup -Identity DAG2 -
DatacenterActivationMode DagOnly
```

In the preceding example, DAG2 is enabled for DAC mode.

For more information about enabling DAC mode, see Configure database availability group properties and Set-DatabaseAvailabilityGroup.

Mailbox database copies

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-05*

Microsoft Exchange Server 2013 leverages the concept of database mobility, which is Exchange-managed database-level failovers. Database mobility disconnects databases from servers, adds support for up to 16 copies of a single database, and provides a native experience for adding database copies to a database.

Key characteristics

The key characteristics of mailbox database copies are:

- Up to 16 copies of an Exchange 2013 mailbox database can be created on multiple Mailbox servers, provided the servers are grouped into a database availability group (DAG), which is a boundary for continuous replication. Exchange 2013 mailbox databases can be replicated only to other Exchange 2013 Mailbox servers within a DAG. You can't replicate a database outside of a DAG, nor can you replicate an Exchange 2013 mailbox database to a server running Exchange 2010 or earlier. For detailed information about DAGs, see Database availability groups.
- All Mailbox servers in a DAG must be in the same Active Directory domain.
- Mailbox database copies support the concepts of replay lag time and truncation lag time. Appropriate planning must be performed before enabling these features.
- All database copies can be backed up using an Exchange-aware, Volume Shadow Copy Service (VSS)-based backup application.
- Database copies can be created only on Mailbox servers that don't host the active copy of a database. You can't create two copies of the same database on the same server.
- All copies of a database use the same path on each server containing a copy. The database and log file paths for a database copy on each Mailbox server must not conflict with any other database paths.
- Database copies can be created in the same or different Active Directory sites, and on the same or different network subnets.
- Database copies aren't supported between Mailbox servers with round trip network latency greater than 500 milliseconds (ms).

Mailbox database copies

You can create a mailbox database copy at any time. Mailbox database copies can be distributed across Mailbox servers in a flexible and granular way.

You can create a mailbox database copy using the **Add mailbox database copy** wizard in the Exchange Administration Center or by using the **Add-MailboxDatabaseCopy** cmdlet in the Exchange Management Shell.

When creating a mailbox database copy, specify the following parameters:

- *Identity* This parameter specifies the name of the database being copied. Database names must be unique within the Exchange organization.
- *MailboxServer* This parameter specifies the name of the Mailbox server that will host the database copy. This server must be a member of the same DAG and must not already host a copy of the database.

Optionally, you can also specify:

- *ActivationPreference* This parameter specifies the activation preference number, which is used as part of Active Manager's best copy selection process. It's also used to redistribute active mailbox databases throughout the DAG when using the RedistributeActiveDatabases.ps1 script. The value for the activation preference is a number equal to or greater than one, where one is at the top of the preference order. The position number cannot be larger than the number of mailbox database copies.
- *ReplayLagTime* This parameter specifies the amount of time that the Microsoft Exchange Replication service should wait before replaying log files that are copied to the database copy. The format for this parameter is (Days.Hours:Minutes:Seconds). The default setting for this value is 0 seconds. The maximum allowable setting for this value is 14 days. The minimum allowable setting is 0 seconds. Setting the value for replay lag time to 0 turns off log replay delay.
- *TruncationLagTime* This parameter specifies the amount of time that the Microsoft Exchange Replication service should wait before truncating log files that have replayed into a copy of the database. The time period begins after the log has been successfully replayed into the copy of the database. The format for this parameter is (Days.Hours:Minutes:Seconds). The default setting for this value is 0 seconds. The maximum allowable setting for this value is 14 days. The minimum allowable setting is 0 seconds. Setting the value for truncation lag time to 0 turns off log truncation delay.
- *SeedingPostponed* This parameter specifies that the task shouldn't automatically seed the database copy on the specified Mailbox server. This option is typically used when you intend to seed a new mailbox database copy by using an existing passive copy of the database (for example, adding a second copy of a specific database to a remote location). When you use this parameter, you must manually seed the database copy using the Update-MailboxDatabaseCopy cmdlet.

For more information about creating, using, and managing mailbox database copies, see Managing mailbox database copies.

AutoReseed

Exchange Server 2013 > High availability and site resilience > Database availability groups >

Applies to: *Exchange Server 2013 SP1, Exchange Server 2013*

Topic Last Modified: *2014-08-07*

Automatic Reseed, or AutoReseed, is a feature that's the replacement for what is normally administrator-driven action in response to a disk failure, database corruption event, or other issue that necessitates a reseed of a database copy. AutoReseed is designed to automatically restore database redundancy after a disk failure by using spare disks that have been provisioned on the system.

Overview of Autoreseed

In an AutoReseed configuration, a standardized storage presentation structure is used, and the administrator picks the starting point. AutoReseed is about restoring redundancy as soon as possible after a drive fails. This involves pre-mapping a set of volumes (including spare volumes) and databases using mount points. In the event of a disk failure where the disk is no longer available to the operating system, or is no longer writable, a spare volume is allocated by the system, and the affected database copies are reseeded automatically.

1. The Microsoft Exchange Replication service periodically scans for copies that have a status of FailedAndSuspended. If all database copies on a volume configured for AutoReseed are in a FailedandSuspended state for 15 consecutive minutes, the AutoReseed workflow is initiated.
2. AutoReseed will try to resume the failed and suspended copies up to three times, with a 5-minute sleep in between each attempt. Sometimes, after a FailedandSuspended database copy is resumed, the copy remains in a Failed state. This can happen for a variety of reasons, so this step is designed to handle those cases; AutoReseed will automatically suspend a database copy that has been Failed for 10 consecutive minutes to keep the workflow running. If the suspend and resume actions don’t result in a healthy database copy, the workflow continues.
3. When it finds a copy with that status, it performs some prerequisite checks. For example, it will verify that a spare disk is available, that the database and its log files are configured on the same volume, and in the appropriate locations that match the required naming conventions.
4. If the prerequisite checks pass successfully, the Disk Reclaimer function within the Microsoft Exchange Replication service allocates, remaps and formats a spare disk according to the timelines in the table below. AutoReseed will attempt to assign a spare volume up to 5 times, with 1-hour sleeps in between each try.
5. Once a spare has been assigned, AutoReseed will perform an InPlaceSeed operation using the SafeDeleteExistingFiles seeding switch. All databases that were on the affected disk are re-seeded using the active copy of the database as the seeding source.
6. After the seeding operation has been completed, the Microsoft Exchange Replication service

verifies that the newly seeded copy is healthy.

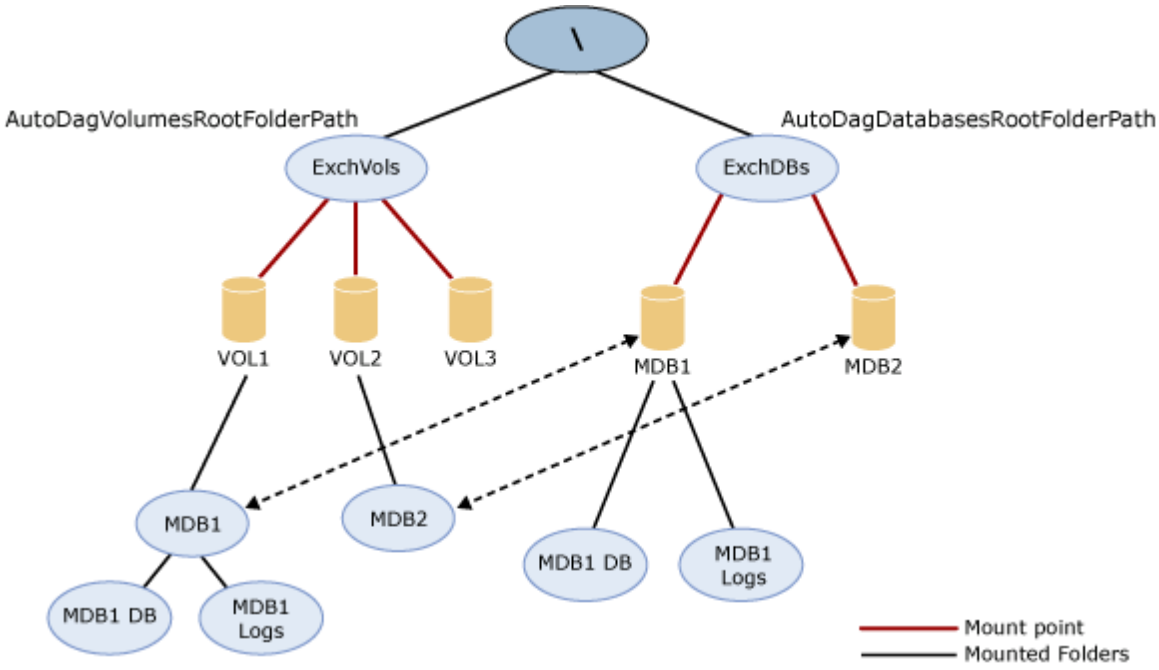
Once all retries are exhausted, the workflow stops. If, after 3 days, the database copy is still FailedandSuspended, the workflow state is reset and it starts again from Step 1. This reset/resume behavior is useful (and intentional) since it can take a few days to replace a failed disk, controller, etc.

At this point, if the failure was a disk failure, it would require manual intervention by an operator or administrator to remove and replace the failed disk and reconfigure the replacement disk as a spare.

AutoReseed is configured using three properties of the DAG. Two of the properties refer to the two mount points that are in use. Exchange 2013 leverages the fact that Windows Server allows multiple mount points per volume. The *AutoDagVolumesRootFolderPath* property refers to the mount point that contains all of the available volumes. This includes volumes that host databases and spare volumes. The *AutoDagDatabasesRootFolderPath* property refers to the mount point that contains the databases. A third DAG property, *AutoDagDatabaseCopiesPerVolume*, is used to configure the number of database copies per volume.

An example AutoReseed configuration is illustrated below.

Example AutoReseed configuration



In this example, there are three volumes, two of which will contain databases (VOL1 and VOL2), and one of which is a blank, formatted spare (VOL3).

To configure AutoReseed:

1. All three volumes are mounted under a single mount point. In this example, a mount point of C:\ExchVols is used. This represents the directory used to get storage for Exchange databases.
2. The root directory of the mailbox databases is mounted as another mount point. In this example, a mount point of C:\ExchDBs is used. Next, a directory structure is created so that a parent directory is created for the database, and under the parent directory, two subdirectories are created: one database file and one for the log files.

3. Databases are created. The above example illustrates a simple design using a single database per volume. Thus, on VOL1, there are three directories: the parent directory and two subdirectories (one for MDB1's database file, and one for its logs). Although not depicted in the example image, on VOL2, there would also be three directories: the parent directory, and under that, a directory for MDB2's database file, and one for its log files.

In this configuration, if MDB1 or MDB2 were to experience a failure, a copy of the failed database will be automatically reseeded to VOL3.

Disk Reclaimer

The AutoReseed component that allocates and formats spare disks is called the *Disk Reclaimer*. The Disk Reclaimer component automatically formats spare disks in preparation for automatic reseeding at different intervals, depending on the state of the disk. In order for the Disk Reclaimer to format a disk, certain conditions must met:

- The Disk Reclaimer must be enabled. It is enabled by default, but it can be disabled using Set-DatabaseAvailabilityGroup.
- The volume must have a mount point in the root volumes path (by default, C:\ExchangeVolumes).
- The volume must not have any mount points in the database volumes path (by default, C:\ExchangeDatabases).
- If the volume contains any files, none of the files must have been touched for 24 hours.

In addition to the above conditions, the Disk Reclaimer will only attempt to format a given volume once a day. The following table describes the formatting behavior of the Disk Reclaimer.

State of Disk and Database Copies	Formatting Interval
Disk is unformatted, or formatted but empty, or formatted but contains files that have not been touched for 24 hours, and there are healthy active database copies in the local Active Directory site that can be used as a seeding source.	1 day
Disk is unformatted, or formatted but empty, or formatted but contains files that have not been touched for 24 hours, but there are no healthy active database copies in the local Active Directory site that can be used as a seeding source.	2 days
Disk is unformatted, or formatted but empty,	2 weeks

or formatted but contains files that have not been touched for 24 hours, and there are healthy active database copies in the local Active Directory site that can be used as a seeding source, but there are unknown files outside of the database file (EDB file) and log files.	
Disk is unformatted, or formatted but empty, or formatted but contains files that have not been touched for 24 hours, and there are healthy active database copies in the local Active Directory site that can be used as a seeding source, but there are one or more database files (EDB files) for databases that are not present in Active Directory.	2 weeks

Planning for high availability and site resilience

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-05*

During the planning phase, the system architects, administrators, and other key stakeholders should identify the business requirements and the architectural requirements for the deployment; in particular, the requirements about high availability and site resilience.

There are general requirements that must be met for deploying these features, as well as hardware, software, and networking requirements that must also be met.

Contents

General requirements

Before deploying a database availability group (DAG) and creating mailbox database copies, make sure that the following system-wide recommendations are met:

- Domain Name System (DNS) must be running. Ideally, the DNS server should accept dynamic updates. If the DNS server doesn't accept dynamic updates, you must create a DNS host (A) record for each Exchange server. Otherwise, Exchange won't function properly.
- Each Mailbox server in a DAG must be a member server in the same domain.
- Adding an Exchange 2013 Mailbox server that's also a directory server to a DAG isn't supported.
- The name you assign to the DAG must be a valid, available, and unique computer name of 15 characters or less.

Return to top

Hardware requirements

Generally, there are no special hardware requirements specific to DAGs or mailbox database copies. The servers used must meet all of the requirements set forth in the topics for Exchange 2013 prerequisites and Exchange 2013 system requirements.

Return to top

Storage requirements

Generally, there are no special storage requirements specific to DAGs or mailbox database copies. DAGs don't require or use cluster-managed shared storage. Cluster-managed shared storage is supported for use in a DAG only when the DAG is configured to use a solution that leverages the Third Party Replication API built into Exchange 2013.

Return to top

Software requirements

DAGs are available in both Exchange 2013 Standard and Exchange 2013 Enterprise. In addition, a DAG can contain a mix of servers running Exchange 2013 Standard and Exchange 2013 Enterprise.

Each member of the DAG must also be running the same operating system. Exchange 2013 is

supported on the Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2 operating systems. All members of a specific DAG must run the same operating system. Windows Server 2012 R2 is supported only for DAG members that are running Exchange 2013 Service Pack 1 or later.

In addition to meeting the prerequisites for installing Exchange 2013, there are operating system requirements that must be met. DAGs use Windows Failover Clustering technology, and as a result, they require the Enterprise or Datacenter version of Windows Server 2008 R2, or the Standard or Datacenter version of the Windows Server 2012 or Windows Server 2012 R2 operating systems.

Return to top

Network requirements

There are specific networking requirements that must be met for each DAG and for each DAG member. Each DAG must have a single *MAPI network*, which is used by a DAG member to communicate with other servers (for example, other Exchange 2013 servers or directory servers), and zero or more *Replication networks*, which are networks dedicated to log shipping and seeding.

In previous versions of Exchange, we recommended at least two networks (one MAPI network and one Replication network) for DAGs. In Exchange 2013, multiple networks are supported, but our recommendation depends on your physical network topology. If you have multiple physical networks between DAG members that are physically separate from one another, then using a separate MAPI and Replication network provides additional redundancy. If you have multiple networks that are partially physically separate but converge into a single physical network (for example, a single WAN link), then using a single network (preferably 10 gigabit Ethernet) for both MAPI and Replication traffic is recommended. This provides simplicity for the network and the network path.

Consider the following when designing the network infrastructure for your DAG:

- Each member of the DAG must have at least one network adapter that's able to communicate with all other DAG members. If you're using a single network path, we recommend that you use a minimum of 1 gigabit Ethernet, but preferably 10 gigabit Ethernet. In addition, when using a single network adapter in each DAG member, we recommend that you design the overall solution with the single network adapter and path in mind.
- Using two network adapters in each DAG member provides you with one MAPI network and one Replication network, with redundancy for the Replication network and the following recovery behaviors:
  - In the event of a failure affecting the MAPI network, a server failover will occur (assuming there are healthy mailbox database copies that can be activated).
  - In the event of a failure affecting the Replication network, if the MAPI network is unaffected by the failure, log shipping and seeding operations will revert to use the MAPI network, even if the MAPI network has it's *ReplicationEnabled* property set to False. When the failed Replication network is restored to health and ready to resume log shipping and seeding operations, you

must manually switch over to the Replication network. To change replication from the MAPI network to a restored Replication network, you can either suspend and resume continuous replication by using the **Suspend-MailboxDatabaseCopy** and **Resume-MailboxDatabaseCopy** cmdlets, or restart the Microsoft Exchange Replication service. We recommend using suspend and resume operations to avoid the brief outage caused by restarting the Microsoft Exchange Replication service.

- Each DAG member must have the same number of networks. For example, if you plan on using a single network adapter in one DAG member, all members of the DAG must also use a single network adapter.
- Each DAG must have no more than one MAPI network. The MAPI network must provide connectivity to other Exchange servers and other services, such as Active Directory and DNS.
- Additional Replication networks can be added, as needed. You can also prevent an individual network adapter from being a single point of failure by using network adapter teaming or similar technology. However, even when using teaming, this doesn't prevent the network itself from being a single point of failure. Moreover, teaming adds unnecessary complexity to the DAG.
- Each network in each DAG member server must be on its own network subnet. Each server in the DAG can be on a different subnet, but the MAPI and Replication networks must be routable and provide connectivity, such that:
  - Each network in each DAG member server is on its own network subnet that's separate from the subnet used by each other network in the server.
  - Each DAG member server's MAPI network can communicate with each other DAG member's MAPI network.
  - Each DAG member server's Replication network can communicate with each other DAG member's Replication network.
  - There is no direct routing that allows heartbeat traffic from the Replication network on one DAG member server to the MAPI network on another DAG member server, or vice versa, or between multiple Replication networks in the DAG.
- Regardless of their geographic location relative to other DAG members, each member of the DAG must have round trip network latency no greater than 500 milliseconds between each other member. As the round trip latency between two Mailbox servers hosting copies of a database increases, the potential for replication not being up to date also increases. Regardless of the latency of the solution, customers should validate that the networks between all DAG members is capable of satisfying the data protection and availability goals of the deployment. Configurations with higher latency values may require special tuning of DAG, replication, and network parameters, such as increasing the number of databases or decreasing the number of mailboxes per database, to achieve the desired goals.
- Round trip latency requirements may not be the most stringent network bandwidth and latency requirement for a multi-datacenter configuration. You must evaluate the total network load, which includes client access, Active Directory, transport, continuous replication, and other application traffic, to determine the necessary network requirements for your environment.
- DAG networks support Internet Protocol version 4 (IPv4) and IPv6. IPv6 is supported only when IPv4 is also used; a pure IPv6 environment isn't supported. Using IPv6 addresses and IP address

ranges is supported only when both IPv6 and IPv4 are enabled on that computer, and the network supports both IP address versions. If Exchange 2013 is deployed in this configuration, all server roles can send data to and receive data from devices, servers, and clients that use IPv6 addresses.

- Automatic Private IP Addressing (APIPA) is a feature of Windows that automatically assigns IP addresses when no Dynamic Host Configuration Protocol (DHCP) server is available on the network. APIPA addresses (including manually assigned addresses from the APIPA address range) aren't supported for use by DAGs or by Exchange 2013.

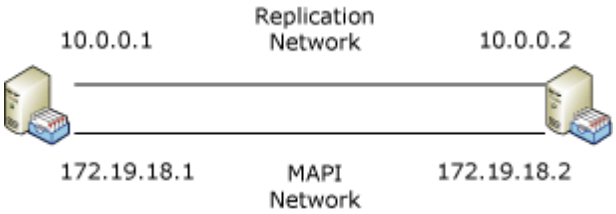
DAG name and IP address requirements

During creation, each DAG is given a unique name, and either assigned one or more static IP addresses, or configured to use DHCP. Regardless of whether you use static or dynamically assigned addresses, any IP address assigned to the DAG must be on the MAPI network.

Each DAG running on Windows Server 2008 R2 or Windows Server 2012 requires a minimum of one IP address on the MAPI network. A DAG requires additional IP addresses when the MAPI network is extended across multiple subnets. DAGs running on Windows Server 2012 R2 that are created without a cluster administrative access point do not require an IP address.

The following figure illustrates a DAG where all nodes in the DAG have the MAPI network on the same subnet.

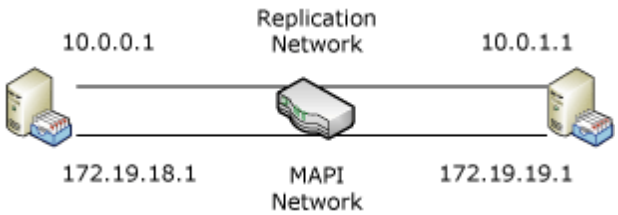
DAG with MAPI network on same subnet



In this example, the MAPI network in each DAG member is on the 172.19.18.*x* subnet. As a result, the DAG requires a single IP address on that subnet.

The next figure illustrates a DAG that has a MAPI network that extends across two subnets: 172.19.18.*x* and 172.19.19.*x*.

DAG with MAPI network on multiple subnets



In this example, the MAPI network in each DAG member is on a separate subnet. As a result, the DAG requires two IP addresses, one for each subnet on the MAPI network.

Each time the DAG's MAPI network is extended across an additional subnet, an additional IP address for that subnet must be configured for the DAG. Each IP address that's configured for the DAG is assigned to and used by the DAG's underlying failover cluster. The name of the DAG is also

used as the name for the underlying failover cluster.

At any specific time, the cluster for the DAG will use only one of the assigned IP addresses. Windows Failover Clustering registers this IP address in DNS when the cluster IP address and Network Name resources are brought online. In addition to using an IP address and network name, a cluster name object (CNO) is created in Active Directory. The name, IP address, and CNO for the cluster are used internally by the system to secure the DAG and for internal communication purposes. Administrators and end users don't need to interface with or connect to the DAG name or IP address.

Note:
Although the cluster's IP address and network name are used internally by the system, there is no hard dependency in Exchange 2013 that these resources be available. Even if the underlying cluster's administrative access point (e.g., it's IP address and Network Name resources) is offline, internal communication still occurs within the DAG by using the DAG member server names. However, we recommend that you periodically monitor the availability of these resources to ensure that they aren't offline for more than 30 days. If the underlying cluster is offline for more than 30 days, the cluster CNO account may be invalidated by the garbage collection mechanism in Active Directory.

Network adapter configuration for DAGs

Each network adapter must be configured properly based on its intended use. A network adapter that's used for a MAPI network is configured differently from a network adapter that's used for a Replication network. In addition to configuring each network adapter correctly, you must also configure the network connection order in Windows so that the MAPI network is at the top of the connection order. For detailed steps about how to modify the network connection order, see Modify the protocol bindings and network provider order.

MAPI network adapter configuration

A network adapter intended for use by a MAPI network should be configured as described in the following table.

Networking features	Settings
Client for Microsoft Networks	Enabled
QoS Packet Scheduler	Optionally enabled
File and Printer Sharing for Microsoft Networks	Enabled
Internet Protocol version 6 (TCP/IP v6)	Enabled
Internet Protocol version 4 (TCP/IP v4)	Enabled

Link-Layer Topology Discovery Mapper I/O Driver	Enabled
Link-Layer Topology Discovery Responder	Enabled

The TCP/IP v4 properties for a MAPI network adapter are configured as follows:

- The IP address for a DAG member's MAPI network can be manually assigned or configured to use DHCP. If DHCP is used, we recommend using persistent reservations for the server's IP address.
- The MAPI network typically uses a default gateway, although one isn't required.
- At least one DNS server address must be configured. Using multiple DNS servers is recommended for redundancy.
- The **Register this connection's addresses in DNS** check box should be selected.

Replication network adapter configuration

A network adapter intended for use by a Replication network should be configured as described in the following table.

Networking features	Settings
Client for Microsoft Networks	Disabled
QoS Packet Scheduler	Optionally enabled
File and Printer Sharing for Microsoft Networks	Disabled
Internet Protocol version 6 (TCP/IP v6)	Enabled
Internet Protocol version 4 (TCP/IP v4)	Enabled
Link-Layer Topology Discovery Mapper I/O Driver	Enabled
Link-Layer Topology Discovery Responder	Enabled

The TCP/IP v4 properties for a Replication network adapter are configured as follows:

- The IP address for a DAG member's Replication network can be manually assigned or configured to use DHCP. If DHCP is used, we recommend using persistent reservations for the server's IP address.
- Replication networks typically don't have default gateways, and if the MAPI network has a default gateway, no other networks should have default gateways. Routing of network traffic on a Replication network can be configured by using persistent, static routes to the corresponding network on other DAG members using gateway addresses that have the ability to route between

the Replication networks. All other traffic not matching this route will be handled by the default gateway that's configured on the adapter for the MAPI network.
- DNS server addresses shouldn't be configured.
- The **Register this connection's addresses in DNS** check box shouldn't be selected.

Return to top

Witness server requirements

A *witness server* is a server outside a DAG that's used to achieve and maintain quorum when the DAG has an even number of members. DAGs with an odd number of members don't use a witness server. All DAGs with an even number of members must use a witness server. The witness server can be any computer running Windows Server. There is no requirement that the version of the Windows Server operating system of the witness server matches the operating system used by the DAG members.

Quorum is maintained at the cluster level, underneath the DAG. A DAG has quorum when the majority of its members are online and can communicate with the other online members of the DAG. This notion of quorum is one aspect of the concept of quorum in Windows failover clustering. A related and necessary aspect to quorum in failover clusters is the *quorum resource*. The quorum resource is a resource inside a failover cluster that provides a means for arbitration leading to cluster state and membership decisions. The quorum resource also provides persistent storage for storing configuration information. A companion to the quorum resource is the *quorum log*, which is a configuration database for the cluster. The quorum log contains information such as which servers are members of the cluster, what resources are installed in the cluster, and the state of those resources (for example, online or offline).

It's critical that each DAG member have a consistent view of how the DAG's underlying cluster is configured. The quorum acts as the definitive repository for all configuration information relating to the cluster. The quorum is also used as a tie-breaker to avoid *split-brain* syndrome. Split brain syndrome is a condition that occurs when DAG members can't communicate with each other but are running. Split brain syndrome is prevented by always requiring a majority of the DAG members (and in the case of DAGs with an even number of member, the DAG witness server) to be available and interacting for the DAG to be operational.

Return to top

Planning for site resilience

Every day, more businesses recognize that access to a reliable and available messaging system is fundamental to their success. For many organizations, the messaging system is part of the business continuity plans, and their messaging service deployment is designed with site resilience in mind. Fundamentally, many site resilient solutions involve the deployment of hardware in a second datacenter.

Ultimately, the overall design of a DAG, including the number of DAG members and the number of mailbox database copies, will depend on each organization's recovery service level agreements (SLAs) that cover various failure scenarios. During the planning stage, the solution's architects and administrators identify the requirements for the deployment, including in particular the requirements for site resilience. They identify the locations to be used and the required recovery SLA targets. The SLA will identify two specific elements that should be the basis for the design of a solution that provides high availability and site resilience: the recovery time objective and the recovery point objective. Both of these values are measured in minutes. The recovery time objective is how long it takes to restore service. The recovery point objective refers to how current the data is after the recovery operation has completed. An SLA may also be defined for restoring the primary datacenter to full service after its problems are corrected.

The solution's architects and administrators will also identify which set of users require site resilience protection, and determine if the multiple site solution will be an active/passive or active/active configuration. In an active/passive configuration, no users are normally hosted in the standby datacenter. In an active/active configuration, users are hosted in both locations, and some percentage of the total number of databases within the solution has a preferred active location in a second datacenter. When service for the users of one datacenter fails, those users are activated in the other datacenter.

Constructing the appropriate SLAs often requires answering the following basic questions:

- What level of service is required after the primary datacenter fails?
- Do users need their data or just messaging services?
- How rapidly is data required?
- How many users must be supported?
- How will users access their data?
- What is the standby datacenter activation SLA?
- How is service moved back to the primary datacenter?
- Are the resources dedicated to the site resilience solution?

By answering these questions, you begin to shape a site resilient design for your messaging solution. A core requirement of recovery from site failure is to create a solution that gets the necessary data to the backup datacenter that hosts the backup messaging service.

Certificate planning

There are no unique or special design considerations for certificates when deploying a DAG in a single datacenter. However, when extending a DAG across multiple datacenters in a site resilient configuration, there are some specific considerations with respect to certificates. Generally, your certificate design will depend on the clients in use, as well as the certificate requirements by other applications that use certificates. But there are some specific recommendations and best practices you should follow with respect to the type and number of certificates.

As a best practice, you should minimize the number of certificates you use for your Exchange servers and reverse proxy servers. We recommend using a single certificate for all of these service

endpoints in each datacenter. This approach minimizes the number of certificates that are needed, which reduces both cost and complexity for the solution.

For Outlook Anywhere clients, we recommend that you use a single subject alternative name (SAN) certificate for each datacenter, and include multiple host names in the certificate. To ensure Outlook Anywhere connectivity after a database, server, or datacenter switchover, you must use the same Certificate Principal Name on each certificate, and configure the Outlook Provider Configuration object in Active Directory with the same Principal Name in Microsoft-Standard Form (msstd). For example, if you use a Certificate Principal Name of mail.contoso.com, you would configure the attribute as follows.

```
Set-OutlookProvider EXPR -CertPrincipalName
"msstd:mail.contoso.com"
```

Some applications that integrate with Exchange have specific certificate requirements that may require using additional certificates. Exchange 2013 can co-exist with Office Communications Server (OCS). OCS requires certificates with 1024-bit or greater certificates that use the OCS server name for the Certificate Principal Name. Because using an OCS server name for the Certificate Principal Name would prevent Outlook Anywhere from working properly, you would need to use an additional and separate certificate for the OCS environment.

Network planning

In addition to the specific networking requirements that must be met for each DAG, as well as for each server that's a member of a DAG, there are some requirements and recommendations that are specific to site resilience configurations. As with all DAGs, whether the DAG members are deployed in a single site or in multiple sites, the round-trip return network latency between DAG members must be no greater than 500 milliseconds. In addition, there are specific configuration settings that are recommended for DAGs that are extended across multiple sites:

- **MAPI networks should be isolated from Replication networks** Windows network policies, Windows firewall policies, or router access control lists (ACLs) should be used to block traffic between the MAPI network and the Replication networks. This configuration is necessary to prevent network heartbeat cross talk.
- **Client-facing DNS records should have a Time to Live (TTL) value of 5 minutes** The amount of downtime that clients experience is dependent not just on how quickly a switchover can occur, but also on how quickly DNS replication occurs and the clients query for updated DNS information. DNS records for all Exchange client services, including Microsoft Office Outlook Web App, Microsoft Exchange ActiveSync, Exchange Web services, Outlook Anywhere, SMTP, POP3, and IMAP4 in both the internal and external DNS servers should be set with a TTL of 5 minutes.
- **Use static routes to configure connectivity across Replication networks** To provide network connectivity between each of the Replication network adapters, use persistent static routes. This is a quick and one-time configuration that's performed on each DAG member when using static IP addresses. If you're using DHCP to obtain IP addresses for your Replication networks, you can also

use it to assign static routes for the replication, thereby simplifying the configuration process.

General site resilience planning

In addition to the requirements listed above for high availability, there are other recommendations for deploying Exchange 2013 in a site resilient configuration (for example, extending a DAG across multiple datacenters). What you do during the planning phase will directly affect the success of your site resilience solution. For example, poor namespace design can cause difficulties with certificates, and an incorrect certificate configuration can prevent users from accessing services.

To minimize the time it takes to activate a second datacenter, and allow the second datacenter to host the service endpoints of a failed datacenter, the appropriate planning must be completed. The following are examples:

- The SLA goals for the site resilience solution must be well understood and documented.
- The servers in the second datacenter must have sufficient capacity to host the combined user population of both datacenters.
- The second datacenter must have all services enabled that are provided in the primary datacenter (unless the service isn't included as part of the site resilience SLA). This includes Active Directory, networking infrastructure (for example, DNS or TCP/IP), telephony services (if Unified Messaging is in use), and site infrastructure (such as power or cooling).
- For some services to be able to service users from the failed datacenter, they must have the proper server certificates configured. Some services don't allow instancing (for example, POP3 and IMAP4) and only allow the use of a single certificate. In these cases, either the certificate must be a SAN certificate that includes multiple names, or the multiple names must be similar enough so that a wildcard certificate can be used (assuming the security policies of the organization allows the use of wildcard certificates).
- The necessary services must be defined in the second datacenter. For example, if the first datacenter has three different SMTP URLs on different transport servers, the appropriate configuration must be defined in the second datacenter to enable at least one (if not all three) transport server to host the workload.
- The necessary network configuration must be in place to support the datacenter switchover. This might mean making sure that the load balancing configurations are in place, that global DNS is configured, and that the Internet connection is enabled with the appropriate routing configured.
- The strategy for enabling the DNS changes necessary for a datacenter switchover must be understood. The specific DNS changes, including their TTL settings, must be defined and documented to support the SLA in effect.
- A strategy for testing the solution must also be established and factored into the SLA. Periodic validation of the deployment is the only way to guarantee that the quality and viability of the deployment doesn't degrade over time. After the deployment is validated, we recommend that the part of the configuration that directly affects the success of the solution be explicitly documented. In addition, we recommend that you enhance your change management processes around those segments of the deployment.

Return to top

Deploying high availability and site resilience

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-21*

Microsoft Exchange Server 2013 uses the concept known as *incremental deployment* for both high availability and site resilience. You simply install two or more Exchange 2013 Mailbox servers as stand-alone servers, and then incrementally configure them and mailbox databases for high availability and site resilience, as needed.

Overview of the deployment process

While the actual steps used by each organization may vary slightly, the overall process for deploying Exchange 2013 in a highly available or site resilient configuration is generally the same. After performing the necessary planning and design tasks for building and deploying a database availability group (DAG) and creating mailbox database copies, you would:

1. Create a DAG. For detailed steps, see Create a database availability group.
2. If necessary, pre-stage the cluster name object (CNO). Pre-staging the CNO is required when deploying a DAG with Mailbox servers running Windows Server 2012. If you are deploying a DAG without an administrative access point using Mailbox servers running Windows Server 2012 R2, then you do not need to pre-stage a CNO. Pre-staging is also required in environments where computer account creation is restricted or where computer accounts are created in a container other than the default computers container. For detailed steps, see Pre-stage the cluster name object for a database availability group.
3. Add two or more Mailbox servers to the DAG. For detailed steps, see Manage database availability group membership.
4. Configure the DAG properties as needed:
   a. Optionally configure DAG encryption and compression, replication port, DAG IP addresses, and other DAG properties. For detailed steps, see Configure database availability group properties.
   b. Enable Datacenter Activation Coordination (DAC) mode for the DAG. This protects the DAG from database-level split brain conditions during switchback to the primary datacenter after a datacenter switchover has been performed, and it enables the use of the built-in DAG recovery cmdlets. For more information, see Datacenter Activation Coordination mode.

5. Add mailbox database copies across Mailbox servers in the DAG. For detailed steps, see Add a mailbox database copy.

Example deployment: four-member DAG in two datacenters

This example details how an organization, Contoso, Ltd., is configuring and deploying a four-member DAG that will be extended across two physical locations: Redmond, Washington and Portland, Oregon.

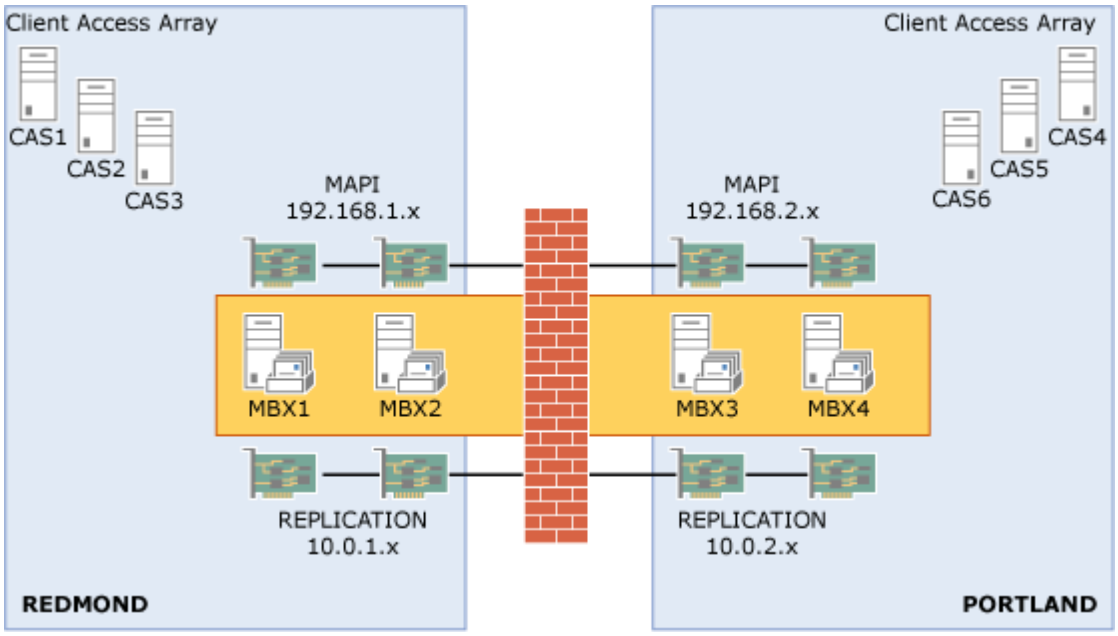
Base infrastructure

Each location contains the infrastructure elements that are necessary to operate a messaging infrastructure based on Exchange 2013, namely:

- Directory services (either Active Directory or Active Directory Domain Services (AD DS))
- Domain Name System (DNS) name resolution
- Multiple Exchange 2013 Client Access servers
- Multiple Exchange 2013 Mailbox servers

The following figure illustrates the Contoso configuration.

Database availability group extended across two sites



Network configuration

As illustrated in the previous figure, the solution involves the use of multiple subnets and multiple networks. Each Mailbox server in the DAG has two network adapters on separate subnets. In each Mailbox server, one network adapter will be used for the MAPI network (192.168.*x.x*) and one network adapter will be used for the Replication network (10.0.*x.x*). Only the MAPI network provides

connectivity to Active Directory, DNS services, other Exchange servers and clients. The adapter used for the Replication network in each member provides connectivity only to the Replication network adapters in the other members of the DAG.

The settings for each network adapter in each node are detailed in the following table.

Name	IPv4 address	Subnet mask	Default gateway
MBX1 (MAPI)	192.168.1.4	255.255.255.0	192.168.1.1
MBX2 (MAPI)	192.168.1.5	255.255.255.0	192.168.1.1
MBX3 (MAPI)	192.168.2.4	255.255.255.0	192.168.2.1
MBX4 (MAPI)	192.168.2.5	255.255.255.0	192.168.2.1
MBX1 (Replication)	10.0.1.4	255.255.0.0	None
MBX2 (Replication)	10.0.1.5	255.255.0.0	None
MBX3 (Replication)	10.0.2.4	255.255.0.0	None
MBX4 (Replication)	10.0.2.5	255.255.0.0	None

As shown in the preceding table, adapters used for Replication networks don't use default gateways. To provide network connectivity between each of the Replication network adapters, Contoso uses persistent static routes, which they configure by using the Netsh.exe tool.

To configure routing for the Replication network adapters on MBX1 and MBX2, the following command was run on each server.

```
netsh interface ipv4 add route 10.0.2.0/24 <NetworkName>
10.0.1.254
```

To configure routing for the Replication network adapters on MBX3 and MBX4, the following command was run on each server.

```
netsh interface ipv4 add route 10.0.1.0/24 <NetworkName>
10.0.2.254
```

The following additional network settings have also been configured:

- The **Register this connection's addresses in DNS** check box is selected for each DAG member's MAPI network adapter, and cleared for each Replication network adapter.
- At least one DNS server address is configured for each DAG member's MAPI network adapter, and none are configured for the Replication network adapters. For redundancy, Contoso is using multiple DNS server addresses for their MAPI network adapters.

- Contoso doesn't use the Windows Firewall and have turned it off on their servers.

After the network adapters have been configured, Contoso is ready to create a DAG and add the Mailbox servers to the DAG.

Database availability group creation and configuration

The administrator has decided to create a Windows PowerShell command-line interface script that performs several tasks:

- It uses the New-DatabaseAvailabilityGroup cmdlet to create the DAG. Because REDMOND is considered to be the primary datacenter, Contoso has chosen to use a witness server in the same datacenter, namely, CAS1.
- It uses the Set-DatabaseAvailabilityGroup cmdlet to preconfigure an alternate witness server and alternate witness directory in case a datacenter switchover is ever necessary.
- It uses the Add-DatabaseAvailabilityGroupServer cmdlet to add each of the four Mailbox servers to the DAG.
- It uses the Set-DatabaseAvailabilityGroup cmdlet to configure the DAG for DAC mode. For more information about DAC mode, see Datacenter Activation Coordination mode.

The following are the commands used in the script:

```
New-DatabaseAvailabilityGroup -Name DAG1 -WitnessServer
CAS1 -WitnessDirectory C:\DAGWitness\DAG1.contoso.com -
DatabaseAvailabilityGroupIPAddresses
192.168.1.8,192.168.2.8
```

The preceding command creates the DAG DAG1, configures CAS1 to act as the witness server, configures a specific witness directory (C:\DAGWitness\DAG1.contoso.com), and configures two IP addresses for the DAG (one for each subnet on the MAPI network).

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
AlternateWitnessDirectory C:\DAGWitness\DAG1.contoso.com -
AlternateWitnessServer CAS4
```

The preceding command configures DAG1 to use an alternate witness server of CAS4 and an alternate witness directory on CAS4 that uses the same path that was configured on CAS1.

Note:

Using the same path isn't required; Contoso has chosen to do this to standardize their configuration.

```
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer MBX1
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
```

```
MailboxServer MBX3
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer MBX2
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer MBX4
```

The preceding commands add each of the Mailbox servers, one at a time, to the DAG. The commands also install the Windows Failover Clustering component on each Mailbox server (if it isn't already installed), create a failover cluster, and join each Mailbox server to the newly created cluster.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
DatacenterActivationMode DagOnly
```

The preceding command enables DAC mode for the DAG.

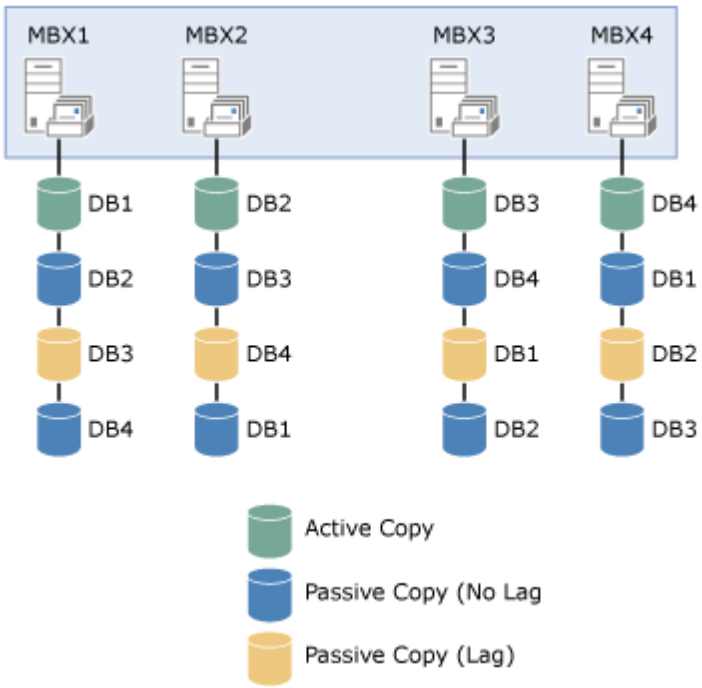
Mailbox databases and mailbox database copies

After creating the DAG and adding the Mailbox servers to the DAG, Contoso prepares to create mailbox databases and mailbox database copies. To meet their criteria for failure resistance, Contoso is planning to configure each mailbox database with three non-lagged database copies, and one lagged database copy. The lagged copy will have a configured log replay delay of three days.

This configuration provides a total of four copies for each database (one active, two non-lagged passives, and a lagged passive). Contoso plans on having four active databases per server. With four active databases per server, and three passive copies of each database, the Contoso solution contains 16 total database copies.

As shown in the following figure, Contoso is taking a balanced approach to their database layout.

Database copy layout for Contoso, Ltd



Each Mailbox server hosts an active mailbox database copy, two non-lagged passive database copies, and one lagged passive database copy. The lagged copy of each active mailbox database is hosted on a Mailbox server in the other site.

To create this configuration, the administrator runs several commands.

On MBX1, run the following commands.

```
Add-MailboxDatabaseCopy -Identity DB1 -MailboxServer MBX2
Add-MailboxDatabaseCopy -Identity DB1 -MailboxServer MBX4
Add-MailboxDatabaseCopy -Identity DB1 -MailboxServer MBX3 -
ReplayLagTime 3.00:00:00 -SeedingPostponed
Suspend-MailboxDatabaseCopy -Identity DB1\MBX3 -
SuspendComment "Seed from MBX4" -Confirm:$False
Update-MailboxDatabaseCopy -Identity DB1\MBX3 -SourceServer
MBX4
Suspend-MailboxDatabaseCopy -Identity DB1\MBX3 -
ActivationOnly
```

On MBX2, run the following commands.

```
Add-MailboxDatabaseCopy -Identity DB2 -MailboxServer MBX1
Add-MailboxDatabaseCopy -Identity DB2 -MailboxServer MBX3
Add-MailboxDatabaseCopy -Identity DB2 -MailboxServer MBX4 -
ReplayLagTime 3.00:00:00 -SeedingPostponed
Suspend-MailboxDatabaseCopy -Identity DB2\MBX4 -
SuspendComment "Seed from MBX3" -Confirm:$False
Update-MailboxDatabaseCopy -Identity DB2\MBX4 -SourceServer
```

```
MBX3
Suspend-MailboxDatabaseCopy -Identity DB2\MBX4 -
ActivationOnly
```

On MBX3, run the following commands.

```
Add-MailboxDatabaseCopy -Identity DB3 -MailboxServer MBX4
Add-MailboxDatabaseCopy -Identity DB3 -MailboxServer MBX2
Add-MailboxDatabaseCopy -Identity DB3 -MailboxServer MBX1 -
ReplayLagTime 3.00:00:00 -SeedingPostponed
Suspend-MailboxDatabaseCopy -Identity DB3\MBX1 -
SuspendComment "Seed from MBX2" -Confirm:$False
Update-MailboxDatabaseCopy -Identity DB3\MBX1 -SourceServer
MBX2
Suspend-MailboxDatabaseCopy -Identity DB3\MBX1 -
ActivationOnly
```

On MBX4, run the following commands.

```
Add-MailboxDatabaseCopy -Identity DB4 -MailboxServer MBX3
Add-MailboxDatabaseCopy -Identity DB4 -MailboxServer MBX1
Add-MailboxDatabaseCopy -Identity DB4 -MailboxServer MBX2 -
ReplayLagTime 3.00:00:00 -SeedingPostponed
Suspend-MailboxDatabaseCopy -Identity DB4\MBX2 -
SuspendComment "Seed from MBX1" -Confirm:$False
Update-MailboxDatabaseCopy -Identity DB4\MBX2 -SourceServer
MBX1
Suspend-MailboxDatabaseCopy -Identity DB4\MBX2 -
ActivationOnly
```

In the preceding examples for the **Add-MailboxDatabaseCopy** cmdlet, the *ActivationPreference* parameter wasn't specified. The task automatically increments the activation preference number with each copy that's added. The original database always has a preference number of 1. The first copy added with the **Add-MailboxDatabaseCopy** cmdlet is automatically assigned a preference number of 2. Assuming no copies are removed, the next copy added is automatically assigned a preference number of 3, and so forth. Thus, in the preceding examples, the passive copy in the same datacenter as the active copy has an activation preference number of 2; the non-lagged passive copy in the remote datacenter has an activation preference number of 3, and the lagged passive copy in the remote datacenter has an activation preference number of 4.

Although there are two copies of each active database across the WAN in the other location, seeding over the WAN was only performed once. This is because Contoso is leveraging the

Exchange 2013 ability to use a passive copy of a database as the source for seeding. Using the Add-MailboxDatabaseCopy cmdlet with the *SeedingPostponed* parameter prevents the task from automatically seeding the new database copy being created. Then, the administrator can suspend the un-seeded copy, and by using the Update-MailboxDatabaseCopy cmdlet with the *SourceServer* parameter, the administrator can specify the local copy of the database as the source of the seeding operation. As a result, seeding of the second database copy added to each location happens locally and not over the WAN.

Note:
In the preceding example, the non-lagged database copy is seeded over the WAN, and that copy is then used to seed the lagged copy of the database that's in the same datacenter as the non-lagged copy.

Contoso has configured one of the passive copies of each mailbox database as a lagged database copy to provide protection against the extremely rare but catastrophic case of database logical corruption. As a result, the administrator is configuring the lagged copies as blocked for activation by using the Suspend-MailboxDatabaseCopy cmdlet with the *ActivationOnly* parameter. This ensures that the lagged database copies won't be activated if a database or server failover occurs.

Validating the solution

After the solution has been deployed and configured, the administrator performs several tasks that validate the solution's readiness prior to moving production mailboxes to the databases in the DAG. The solution should be tested and inspected using several methods, including failure simulations. To validate the solution, the administrator performs several tasks.

To verify the overall health of the DAG, the administrator runs the Test-ReplicationHealth cmdlet. This cmdlet checks several aspects of the replication and replay status to provide information about each Mailbox server and database copy in the DAG.

To verify replication and replay activity, the administrator runs the Get-MailboxDatabaseCopyStatus cmdlet. This cmdlet can provide real-time status information about a specific mailbox database copy or for all mailbox database copies on a specific server. For more information about monitoring the health and status of replicated databases in a DAG, see Monitoring database availability groups.

To verify that switchovers work as expected, the administrator uses the Move-ActiveMailboxDatabase cmdlet to perform a series of database switchovers and server switchovers. When these tasks have completed successfully, the administrator uses the same cmdlet to move the active database copies back to their original locations.

To verify the expected behaviors in various failure scenarios, the administrator performs several tasks that either simulate failures or actually cause failures to occur. For example, the administrator might:

- Unplug the power cord on MBX1, thereby triggering a server failover. The administrator then verifies that DB1 becomes active on another server (preferably MBX2, based on the activation

preference values).

- Unplug the network cable for the MAPI network adapter on MBX2, thereby triggering a server failover. The administrator then verifies that DB2 becomes active on another server (preferably MBX1, based on the activation preference values).
- Take the disk used by the active copy of DB3 offline, thereby triggering a database failover. The administrator then verifies that DB3 becomes active on another server (preferably MBX4, based on activation preference values).

There may be other failure scenarios that are tested by an organization, based on the business needs. After simulating a single failure (such as pulling the power plug), and verifying the solution's recovery behavior, the administrator may revert the solution back to its original configuration. In some cases, the solution may be tested for multiple concurrent failures. Ultimately, your solution test plan will dictate whether the solution is reverted back to its original configuration after each failure simulation has been completed.

In addition, an administrator may decide to disconnect the network connection between the two datacenters, thereby simulating a site failure. Performing a datacenter switchover is a much more involved and coordinated process; however, we recommend the process if the solution being deployed is intended to provide site resilience for the messaging services and data.

Transitioning to operations

After the solution has been deployed, it can be extended further using incremental deployment. At this point, management of the solution would also transition to operation processes, in which the following tasks would be performed:

- Monitor the health and status of DAGs and mailbox database copies. For more information, see Monitoring database availability groups.
- Perform database switchovers as needed. For detailed steps about how to perform a database switchover, see Activate a mailbox database copy.

For more information about managing the solution, see Managing high availability and site resilience.

Managing high availability and site resilience

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-05*

After you build, validate, and deploy a Microsoft Exchange Server 2013 high availability or site resilience solution, the solution transitions from the deployment phase to the operational phase of the overall solution lifecycle. The operational phase consists of several tasks, and all tasks are related to one of the following areas: database availability groups (DAGs), mailbox database copies, performing proactive monitoring, and managing switchovers and failovers.

Contents

Database availability group management

The operational management tasks associated with DAGs include:

- **Creating one or more DAGs** Creating a DAG is typically a one-time procedure performed during the deployment phase of the solution lifecycle. However, there may be reasons for creating DAGs that occur during the operational phase, for example:
  - The DAG is configured for third-party replication mode, and you want to revert to using continuous replication. You can't convert a DAG back to continuous replication; you need to create a DAG.
  - You have servers in multiple domains. All members of the same DAG must also be members of the same domain.
- **Managing DAG membership** Managing DAG members is an infrequent task typically performed during the deployment phase of the solution lifecycle. However, because of the flexibility provided by incremental deployment, managing DAG membership may also be performed throughout the solution lifecycle.
- **Configuring DAG properties** Each DAG has various properties that can be configured as needed. These properties include:
  - **Witness server and witness directory** The witness server is a server outside the DAG that acts as a quorum voter when the DAG contains an even number of members. The witness directory is a directory created and shared on the witness server for use by the system in maintaining a quorum.
  - **IP addresses** Each DAG will have one or more IPv4 addresses, and optionally, one or more IPv6 addresses. The IP addresses assigned to the DAG are used by the DAG's underlying cluster. The number of IPv4 addresses assigned to the DAG equals the number of subnets that comprise the MAPI network used by the DAG. You can configure the DAG to use static IP addresses or to obtain addresses automatically by using Dynamic Host Configuration Protocol (DHCP).
  - **Datacenter Activation Coordination mode** Datacenter Activation Coordination mode is a property setting on a DAG that's designed to prevent split-brain conditions at the database level, in a scenario in which you're restoring service to a primary datacenter after a datacenter

switchover has been performed. For more information about Datacenter Activation Coordination mode, see Datacenter Activation Coordination mode.

  - **Alternate witness server and alternate witness directory** The alternate witness server and alternate witness directory are values that you can preconfigure as part of the planning process for a datacenter switchover. These refer to the witness server and witness directory that will be used when a datacenter switchover has been performed.
  - **Replication port** By default, all DAGs use TCP port 64327 for continuous replication. You can modify the DAG to use a different TCP port for replication by using the *ReplicationPort* parameter of the Set-DatabaseAvailabilityGroup cmdlet.
  - **Network discovery** You can force the DAG to rediscover networks and network interfaces. This operation is used when you add or remove networks or introduce new subnets. Rediscovery of all DAG networks can be forced by using the *DiscoverNetworks* parameter of the Set-DatabaseAvailabilityGroup cmdlet.
  - **Network compression** By default, DAGs use compression only between DAG networks on different subnets. You can enable compression for all DAG networks or for seeding operations only, or you can disable compression for all DAG networks.
  - **Network encryption** By default, DAGs use encryption only between DAG networks on different subnets. You can enable encryption for all DAG networks or for seeding operations only, or you can disable encryption for all DAG networks.
- **Shutting down DAG members** The Exchange 2013 high availability solution is integrated with the Windows shutdown process. If an administrator or application initiates a shutdown of a Windows server in a DAG that has a mounted database that's replicated to one or more DAG members, the system will try to activate another copy of the mounted databases prior to allowing the shutdown process to complete. However, this new behavior doesn't guarantee that all of the databases on the server being shut down will experience a lossless activation. As a result, it's a best practice to perform a server switchover prior to shutting down a server that's a member of a DAG.

For detailed steps about how to create a DAG, see Create a database availability group. For detailed steps about how to configure DAGs and DAG properties, see Configure database availability group properties. For more information about each of the preceding management tasks, and about managing DAGs in general, see Managing database availability groups.

Return to top

Mailbox database copy management

The operational management tasks associated with mailbox database copies include:

- **Adding mailbox database copies** When you add a copy of a mailbox database, continuous replication is automatically enabled between the existing database and the database copy.
- **Configuring mailbox database copy properties** You can configure a variety of properties, such as the database activation policy, the amount of time, if any, for replay lag and truncation lag, and the activation preference for the database copy.

- **Suspending or resuming a mailbox database copy** You can suspend a mailbox database copy in preparation for seeding, or for other forms of maintenance. You can also suspend a mailbox database copy for activation only. This configuration prevents the system from automatically activating the copy as a result of a failure, but it still allows the system to keep the database copy up to date with log shipping and replay.
- **Updating a mailbox database copy** Updating, also known as *seeding*, is the process in which a copy of a mailbox database is added to another Mailbox server. This becomes the baseline database for the copy. After the initial first seed of the baseline database copy, only in rare circumstances will the database need to be seeded again.
- **Activating a mailbox database copy** Activating is the process of designating a specific passive copy as the new active copy of a mailbox database. This process is referred to as a *switchover*. For more information, see "Switchovers and Failovers" later in this topic.
- **Removing a mailbox database copy** You can remove a mailbox database copy at any time. Occasionally, it may be necessary to remove a mailbox database copy. For example, you can't remove a Mailbox server from a DAG until all mailbox database copies are removed from the server. In addition, you must remove all copies of a mailbox database before you can change the path for a mailbox database.

For detailed steps about how to add a mailbox database copy, see Add a mailbox database copy. For detailed steps about how to configure mailbox database copies, see Configure mailbox database copy properties. For more information about each of the preceding management tasks, and about managing mailbox database copies in general, see Managing mailbox database copies. For detailed steps about how to remove a mailbox database copy, see Remove a mailbox database copy.

Return to top

Proactive monitoring

Making sure that your servers are operating reliably and that your database copies are healthy are key objectives for daily messaging operations. Exchange 2013 includes a number of features that can be used to perform a variety of health monitoring tasks for DAGs and mailbox database copies, including:

- Get-MailboxDatabaseCopyStatus
- Test-ReplicationHealth
- Crimson channel event logging

In addition to monitoring the health and status, it's also critical to monitor for situations that can compromise availability. For example, we recommend that you monitor the redundancy of your replicated databases. It's critical to avoid situations where you're down to a single copy of a database. This scenario should be treated with the highest priority and resolved as soon as possible.

For more detailed information about monitoring the health and status of DAGs and mailbox database copies, see Monitoring database availability groups.

Return to top

Switchovers and failovers

A *switchover* is a manual process in which an administrator manually activates one or more mailbox database copies. Switchovers, which can occur at the database or server level, are typically performed as part of preparation for maintenance activities. Switchover management involves performing database or server switchovers as needed. For example, if you need to perform maintenance on a Mailbox server in a DAG, you would first perform a server switchover so that the server didn't host any active mailbox database copies. For detailed steps about how to perform a database switchover, see Activate a mailbox database copy. Switchovers can also be performed at the datacenter level.

A *failover* is the automatic activation by the system of one or more database copies in reaction to a failure. For example, the loss of a disk drive in a RAID-less environment will trigger a database failover. The loss of the MAPI network or a power failure will trigger a server failover.

Return to top

Managing database availability groups

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-21*

A database availability group (DAG) is a set of up to 16 Microsoft Exchange Server 2013 Mailbox servers that provides automatic, database-level recovery from a database, server, or network failure. DAGs use continuous replication and a subset of Windows failover clustering technologies to provide high availability and site resilience. Mailbox servers in a DAG monitor each other for failures. When a Mailbox server is added to a DAG, it works with the other servers in the DAG to provide automatic, database-level recovery from database failures.

When you create a DAG, it's initially empty. When you add the first server to a DAG, a failover cluster is automatically created for the DAG. In addition, the infrastructure that monitors the servers for network or server failures is initiated. The failover cluster heartbeat mechanism and cluster database are then used to track and manage information about the DAG that can change quickly, such as database mount status, replication status, and last mounted location.

Contents

DAG membership

Configuring DAG properties

DAG networks

Configuring DAG members

Performing maintenance on DAG members

Shutting down DAG members

Installing updates on DAG members

Creating DAGs

A DAG can be created using the New Database Availability Group wizard in the Exchange Admin Center (EAC), or by running the **New-DatabaseAvailabilityGroup** cmdlet in the Exchange Management Shell. When creating a DAG, you provide a name for the DAG, and optional witness server and witness directory settings. In addition, you can assign one or more IP addresses to the DAG, either by using static IP addresses or by allowing the DAG to be automatically assigned the necessary IP addresses using Dynamic Host Configuration Protocol (DHCP). You can manually assign IP addresses to the DAG by using the *DatabaseAvailabilityGroupIpAddresses* parameter. If you omit this parameter, the DAG attempts to obtain an IP address by using a DHCP server on your network.

If you are creating a DAG that will contain Mailbox servers that are running Windows Server 2012 R2, you also have the option of creating a DAG without a cluster administrative access point. In that case, the cluster will not have a cluster name object (CNO) in Active Directory, and the cluster core resource group will not contain a network name resource or an IP address resource.

For detailed steps about how to create a DAG, see Create a database availability group.

When you create a DAG, an empty object representing the DAG with the name you specified and an object class of **msExchMDBAvailabilityGroup** is created in Active Directory.

DAGs use a subset of Windows failover clustering technologies, such as the cluster heartbeat, cluster networks, and cluster database (for storing data that changes or can change quickly, such as database state changes from active to passive or the reverse, or from mounted to dismounted or the reverse). Because DAGs rely on Windows failover clustering, they can only be created on Exchange 2013 Mailbox servers running the Windows Server 2008 R2 Enterprise or Datacenter operating system, Windows Server 2012 Standard or Datacenter operating system, or Windows Server 2012 R2 Standard or Datacenter operating system.

Note:

The failover cluster created and used by the DAG must be dedicated to the DAG. The cluster can't be used for any other high availability solution or for any other purpose. For example, the failover cluster can't be used to cluster other applications or services. Using a DAG's underlying failover cluster for purposes other than the DAG isn't supported.

DAG witness server and witness directory

When creating a DAG, you need to specify a name for the DAG no longer than 15 characters that's unique within the Active Directory forest. In addition, each DAG is configured with a witness server and witness directory. The witness server and its directory are used only when there's an even number of members in the DAG and then only for quorum purposes. You don't need to create the witness directory in advance. Exchange automatically creates and secures the directory for you on the witness server. The directory shouldn't be used for any purpose other than for the DAG witness server.

The requirements for the witness server are as follows:

- The witness server can't be a member of the DAG.
- The witness server must be in the same Active Directory forest as the DAG.
- The witness server must be running Windows Server 2012, Windows Server 2008 R2, Windows Server 2008, Windows Server 2003 R2, or Windows Server 2003.
- A single server can serve as a witness for multiple DAGs. However, each DAG requires its own witness directory.

Regardless of what server is used as the witness server, if the Windows Firewall is enabled on the intended witness server, you must enable the Windows Firewall exception for File and Printer Sharing.

Important:

If the witness server you specify isn't an Exchange 2013 or Exchange 2010 server, you must add the Exchange Trusted Subsystem universal security group (USG) to the local Administrators group on the witness server prior to creating the DAG. These security permissions are necessary to ensure that Exchange can create a directory and share on the witness server as needed.

Neither the witness server nor the witness directory needs to be fault tolerant or use any form of redundancy or high availability. There's no need to use a clustered file server for the witness server or employ any other form of resiliency for the witness server. There are several reasons for this. With larger DAGs (for example, six members or more), several failures are required before the witness server is needed. Because a six-member DAG can tolerate as many as two voter failures without losing quorum, it would take as many as three voters failing before the witness server would be needed to maintain a quorum. Also, if there's a failure that affects your current witness server (for example, you lose the witness server because of a hardware failure), you can use the Set-DatabaseAvailabilityGroup cmdlet to configure a new witness server and witness directory (provided you have a quorum).

Note:

You can also use the **Set-DatabaseAvailabilityGroup** cmdlet to configure the witness server and witness directory in the original location if the witness server lost its storage or if someone changed the witness directory or share permissions.

Witness server placement considerations

The placement of a DAG's witness server will depend on your business requirements and the options available to your organization. Exchange 2013 includes support for new DAG configuration options that are not recommended or not possible in previous versions of Exchange. These options include using a third location, such as a third datacenter or a branch office.

The following table lists general witness server placement recommendations for different deployment scenarios.

Deployment Scenario	Recommendations
Single DAG deployed in a single datacenter	Locate witness server in the same datacenter as DAG members
Single DAG deployed across two datacenters; no additional locations available	Locate witness server in primary datacenter
Multiple DAGs deployed in a single datacenter	Locate witness server in the same datacenter as DAG members. Additional options include: • Using the same witness server for multiple DAGs • Using a DAG member to act as a witness server for a different DAG
Multiple DAGs deployed across two datacenters	Locate witness server in the datacenter that is considered primary for each DAG. Additional options include: • Using the same witness server for multiple DAGs • Using a DAG member to act as a witness server for a different DAG
Single or Multiple DAGs deployed across more than two datacenters	In this configuration, the witness server should be located in the datacenter where you want the majority of quorum votes to exist.

When a DAG has been deployed across two datacenters, a new configuration option in Exchange 2013 is to use a third location for hosting the witness server. If your organization has a third location with a network infrastructure that is isolated from network failures that affect the two datacenters in which your DAG is deployed, then you can deploy the DAG's witness server in that

third location, thereby configuring your DAG with the ability automatically failover databases to the other datacenter in response to a datacenter-level failure event.

Specifying a witness server and witness directory during DAG creation

When creating a DAG, you must provide a name for the DAG. You can optionally also specify a witness server and witness directory.

When creating a DAG, the following combinations of options and behaviors are available:

- You can specify only a name for the DAG, and leave the **Witness server** and **Witness directory** fields blank. In this scenario, the wizard searches the local Active Directory site for a Client Access server that doesn't have the Mailbox server installed, and it automatically creates the default directory (%SystemDrive%:\DAGFileShareWitnesses\<*DAGFQDN*>) and default share (<*DAGFQDN*>) on that server and uses that Client Access server as the witness server. For example, consider the witness server CAS3 on which the operating system has been installed onto drive C. A DAG named DAG1 in the contoso.com domain would use a default witness directory of C:\DAGFileShareWitnesses\DAG1.contoso.com, which would be shared as \\CAS3\DAG1.contoso.com.
- You can specify a name for the DAG, the witness server that you want to use, and the directory you want created and shared on the witness server.
- You can specify a name for the DAG and the witness server that you want to use, and leave the **Witness directory** field blank. In this scenario, the wizard creates the default directory on the specified witness server.
- You can specify a name for the DAG, leave the **Witness server** field blank, and specify the directory you want created and shared on the witness server. In this scenario, the wizard searches for a Client Access server that doesn't have the Mailbox server installed, and it automatically creates the specified DAG on that server, shares the directory, and uses that Client Access server as the witness server.

When a DAG is formed, it initially uses the Node Majority quorum model. When the second Mailbox server is added to the DAG, the quorum is automatically changed to a Node and File Share Majority quorum model. When this change occurs, the DAG's cluster begins using the witness server for maintaining quorum. If the witness directory doesn't exist, Exchange automatically creates it, shares it, and provisions the share with full control permissions for the CNO computer account for the DAG.

Note:
Using a file share that's part of a Distributed File System (DFS) namespace isn't supported.

If Windows Firewall is enabled on the witness server before the DAG is created, it may block the creation of the DAG. Exchange uses Windows Management Instrumentation (WMI) to create the directory and file share on the witness server. If Windows Firewall is enabled on the witness server and there are no firewall exceptions configured for WMI, the **New-DatabaseAvailabilityGroup** cmdlet fails with an error. If you specify a witness server, but not a witness directory, you receive the following error message.

The task was unable to create the default witness directory on server <*Server Name*>. Please manually specify a witness directory.

If you specify a witness server and witness directory, you receive the following warning message.

Unable to access file shares on witness server '*ServerName*'. Until this problem is corrected, the database availability group may be more vulnerable to failures. You can use the Set-DatabaseAvailabilityGroup cmdlet to try the operation again. Error: The network path was not found.

If Windows Firewall is enabled on the witness server after the DAG is created but before servers are added, it may block the addition or removal of DAG members. If Windows Firewall is enabled on the witness server and there are no firewall exceptions configured for WMI, the **Add-DatabaseAvailabilityGroupServer** cmdlet displays the following warning message.

Failed to create file share witness directory 'C:\DAGFileShareWitnesses\DAG_FQDN' on witness server '*ServerName*'. Until this problem is corrected, the database availability group may be more vulnerable to failures. You can use the Set-DatabaseAvailabilityGroup cmdlet to try the operation again. Error: WMI exception occurred on server '*ServerName*': The RPC server is unavailable. (Exception from HRESULT: 0x800706BA)

To resolve the preceding error and warnings, do one of the following:

- Manually create the witness directory and share on the witness server, and assign the CNO for the DAG full control for the directory and share.
- Enable the WMI exception in Windows Firewall.
- Disable Windows Firewall.

Return to top

DAG membership

After a DAG has been created, you can add servers to or remove servers from the DAG using the Manage Database Availability Group wizard in the EAC, or using the **Add-DatabaseAvailabilityGroupServer** or **Remove-DatabaseAvailabilityGroupServer** cmdlets in the Shell. For detailed steps about how to manage DAG membership, see Manage database availability group membership.

Note:
Each Mailbox server that's a member of a DAG is also a node in the underlying cluster used by the DAG. As a result, at any one time, a Mailbox server can be a member of only one DAG.

If the Mailbox server being added to a DAG doesn't have the failover clustering component

installed, the method used to add the server (for example, the **Add-DatabaseAvailabilityGroupServer** cmdlet or the Manage Database Availability Group wizard) installs the failover clustering feature.

When the first Mailbox server is added to a DAG, the following occurs:

- The Windows failover clustering component is installed, if it isn't already installed.
- A failover cluster is created using the name of the DAG. This failover cluster is used exclusively by the DAG, and the cluster must be dedicated to the DAG. Use of the cluster for any other purpose isn't supported.
- A CNO is created in the default computers container.
- The name and IP address of the DAG is registered as a Host (A) record in Domain Name System (DNS).
- The server is added to the DAG object in Active Directory.
- The cluster database is updated with information on the databases mounted on the added server.

In a large or multiple site environment, especially those in which the DAG is extended to multiple Active Directory sites, you must wait for Active Directory replication of the DAG object containing the first DAG member to complete. If this Active Directory object isn't replicated throughout your environment, adding the second server may cause a new cluster (and new CNO) to be created for the DAG. This is because the DAG object appears empty from the perspective of the second member being added, thereby causing the **Add-DatabaseAvailabilityGroupServer** cmdlet to create a cluster and CNO for the DAG, even though these objects already exist. To verify that the DAG object containing the first DAG server has been replicated, use the **Get-DatabaseAvailabilityGroup** cmdlet on the second server being added to verify that the first server you added is listed as a member of the DAG.

When the second and subsequent servers are added to the DAG, the following occurs:

- The server is joined to the Windows failover cluster for the DAG.
- The quorum model is automatically adjusted:
  - A Node Majority quorum model is used for DAGs with an odd number of members.
  - A Node and File Share Majority quorum model is used for DAGs with an even number of members.
- The witness directory and share are automatically created by Exchange when needed.
- The server is added to the DAG object in Active Directory.
- The cluster database is updated with information about mounted databases.

Note:

The quorum model change should happen automatically. However, if the quorum model doesn't automatically change to the proper model, you can run the **Set-DatabaseAvailabilityGroup** cmdlet with only the *Identity* parameter to correct the quorum settings for the DAG.

Pre-staging the cluster name object for a DAG

The CNO is a computer account created in Active Directory and associated with the cluster's Name

resource. The cluster's Name resource is tied to the CNO, which is a Kerberos-enabled object that acts as the cluster's identity and provides the cluster's security context. The formation of the DAG's underlying cluster and the CNO for that cluster is performed when the first member is added to the DAG. When the first server is added to the DAG, remote PowerShell contacts the Microsoft Exchange Replication service on the Mailbox server being added. The Microsoft Exchange Replication service installs the failover clustering feature (if it isn't already installed) and begins the cluster creation process. The Microsoft Exchange Replication service runs under the LOCAL SYSTEM security context, and it's under this context in which cluster creation is performed.

⚠Warning:

If your DAG members are running Windows Server 2012, you must pre-stage the CNO prior to adding the first server to the DAG. If your DAG members are running Windows Server 2012 R2, and you create a DAG without a cluster administrative access point, then a CNO will not be created, and you do not need to create a CNO for the DAG.

In environments where computer account creation is restricted, or where computer accounts are created in a container other than the default computers container, you can pre-stage and provision the CNO. You create and disable a computer account for the CNO, and then either:

- Assign full control of the computer account to the computer account of the first Mailbox server you're adding to the DAG.
- Assign full control of the computer account to the Exchange Trusted Subsystem USG.

Assigning full control of the computer account to the computer account of the first Mailbox server you're adding to the DAG ensures that the LOCAL SYSTEM security context will be able to manage the pre-staged computer account. Assigning full control of the computer account to the Exchange Trusted Subsystem USG can be used instead because the Exchange Trusted Subsystem USG contains the machine accounts of all Exchange servers in the domain.

For detailed steps about how to pre-stage and provision the CNO for a DAG, see Pre-stage the cluster name object for a database availability group.

Removing servers from a DAG

Mailbox servers can be removed from a DAG by using the Manage Database Availability Group wizard in the EAC or the **Remove-DatabaseAvailabilityGroupServer** cmdlet in the Shell. Before a Mailbox server can be removed from a DAG, all replicated mailbox databases must first be removed from the server. If you attempt to remove a Mailbox server with replicated mailbox databases from a DAG, the task fails.

There are scenarios in which you must remove a Mailbox server from a DAG before performing certain operations. These scenarios include:

- **Performing a server recovery operation** If a Mailbox server that's a member of a DAG is lost, or otherwise fails and is unrecoverable and needs replacement, you can perform a server recovery operation using the **Setup /m:RecoverServer** switch. However, before you can perform the recovery operation, you must first remove the server from the DAG using the Remove-

DatabaseAvailabilityGroupServer cmdlet with the *ConfigurationOnly* parameter.

- **Removing the database availability group** There may be situations in which you need to remove a DAG (for example, when disabling third-party replication mode). If you need to remove a DAG, you must first remove all servers from the DAG. If you attempt to remove a DAG that contains any members, the task fails.

Return to top

Configuring DAG properties

After servers have been added to the DAG, you can use the EAC or the Shell to configure the properties of a DAG, including the witness server and witness directory used by the DAG, and the IP addresses assigned to the DAG.

Configurable properties include:

- **Witness server** The name of the server that you want to host the file share for the file share witness. We recommend that you specify a Client Access server as the witness server. This enables the system to automatically configure, secure, and use the share, as needed, and enables the messaging administrator to be aware of the availability of the witness server.
- **Witness directory** The name of a directory that will be used to store file share witness data. This directory will automatically be created by the system on the specified witness server.
- **Database availability group IP addresses** One or more IP addresses must be assigned to the DAG, unless the DAG members are running Windows Server 2012 R2 and you are creating a DAG without an IP address. Otherwise, the DAG's IP addresses can be configured using manually assigned static IP addresses, or they can be automatically assigned to the DAG using a DHCP server in your organization.

The Shell enables you to configure DAG properties that aren't available in the EAC, such as DAG IP addresses, network encryption and compression settings, network discovery, the TCP port used for replication, and alternate witness server and witness directory settings, and to enable Datacenter Activation Coordination mode.

For detailed steps about how to configure DAG properties, see Configure database availability group properties.

DAG network encryption

DAGs support the use of encryption by leveraging the encryption capabilities of the Windows Server operating system. DAGs use Kerberos authentication between Exchange servers. Microsoft Kerberos security support provider (SSP) EncryptMessage and DecryptMessage APIs handle encryption of DAG network traffic. Microsoft Kerberos SSP supports multiple encryption algorithms. (For the complete list, see section 3.1.5.2, "Encryption Types" of Kerberos Protocol Extensions). The Kerberos authentication handshake selects the strongest encryption protocol supported in the list: typically Advanced Encryption Standard (AES) 256-bit, potentially with a SHA Hash-based Message Authentication Code (HMAC) to maintain integrity of the data. For details, see HMAC.

Network encryption is a property of the DAG and not a DAG network. You can configure DAG network encryption using the **Set-DatabaseAvailabilityGroup** cmdlet in the Shell. The possible encryption settings for DAG network communications are shown in the following table.

DAG network communication encryption settings

Setting	Description
Disabled	Network encryption isn't used.
Enabled	Network encryption is used on all DAG networks for replication and seeding.
InterSubnetOnly	Network encryption is used on DAG networks when replicating across different subnets. This is the default setting.
SeedOnly	Network encryption is used on all DAG networks for seeding only.

DAG network compression

DAGs support built-in compression. When compression is enabled, DAG network communication uses XPRESS, which is the Microsoft implementation of the LZ77 algorithm. For details, see An Explanation of the Deflate Algorithm and section 3.1.4.11.1.2.1 "LZ77 Compression Algorithm" of Wire Format Protocol Specification. This is the same type of compression used in many Microsoft protocols, in particular, MAPI RPC compression between Microsoft Outlook and Exchange.

As with network encryption, network compression is also a property of the DAG and not a DAG network. You configure DAG network compression by using the Set-DatabaseAvailabilityGroup cmdlet in the Shell. The possible compression settings for DAG network communications are shown in the following table.

DAG network communication compression settings

Setting	Description
Disabled	Network compression isn't used.
Enabled	Network compression is used on all DAG networks for replication and seeding.
InterSubnetOnly	Network compression is used on DAG networks when replicating across different

	subnets. This is the default setting.
SeedOnly	Network compression is used on all DAG networks for seeding only.

Return to top

DAG networks

A DAG network is a collection of one or more subnets used for either replication traffic or MAPI traffic. Each DAG contains a maximum of one MAPI network and zero or more replication networks. In a single network adapter configuration, the network is used for both MAPI and replication traffic. Although a single network adapter and path is supported, we recommend that each DAG have a minimum of two DAG networks. In a two-network configuration, one network is typically dedicated for replication traffic, and the other network is used primarily for MAPI traffic. You can also add network adapters to each DAG member and configure additional DAG networks as replication networks.

Note:

When using multiple replication networks, there's no way to specify an order of precedence for network use. Exchange randomly selects a replication network from the group of replication networks to use for log shipping.

In Exchange 2010, manual configuration of DAG networks was necessary in many scenarios. By default in Exchange 2013, DAG networks are automatically configured by the system. Before you can create or modify DAG networks, you must first enable manual DAG network control by running the following command:

```
Set-DatabaseAvailabilityGroup <DAGName> -
ManualDagNetworkConfiguration $true
```

After you've enabled manual DAG network configuration, you can use the **New-DatabaseAvailabilityGroupNetwork** cmdlet in the Shell to create a DAG network. For detailed steps about how to create a DAG network, see Create a database availability group network.

You can use the **Set-DatabaseAvailabilityGroupNetwork** cmdlet in the Shell to configure DAG network properties. For detailed steps about how to configure DAG network properties, see Configure database availability group network properties. Each DAG network has required and optional parameters to configure:

- **Network name** A unique name for the DAG network of up to 128 characters.
- **Network description** An optional description for the DAG network of up to 256 characters.
- **Network subnets** One or more subnets entered using a format of *IPAddress/Bitmask* (for example, 192.168.1.0/24 for Internet Protocol version 4 (IPv4) subnets; 2001:DB8:0:C000::/64 for Internet Protocol version 6 (IPv6) subnets).

- **Enable replication** In the EAC, select the check box to dedicate the DAG network to replication traffic and block MAPI traffic. Clear the check box to prevent replication from using the DAG network and to enable MAPI traffic. In the Shell, use the *ReplicationEnabled* parameter in the Set-DatabaseAvailabilityGroupNetwork cmdlet to enable and disable replication.

Note:
Disabling replication for the MAPI network doesn't guarantee that the system won't use the MAPI network for replication. When all configured replication networks are offline, failed, or otherwise unavailable, and only the MAPI network remains (which is configured as disabled for replication), the system uses the MAPI network for replication.

The initial DAG networks (for example, MapiDagNetwork and ReplicationDagNetwork01) created by the system are based on the subnets enumerated by the Cluster service. Each DAG member must have the same number of network adapters, and each network adapter must have an IPv4 address (and optionally, an IPv6 address as well) on a unique subnet. Multiple DAG members can have IPv4 addresses on the same subnet, but each network adapter and IP address pair in a specific DAG member must be on a unique subnet. In addition, only the adapter used for the MAPI network should be configured with a default gateway. Replication networks shouldn't be configured with a default gateway.

For example, consider DAG1, a two-member DAG where each member has two network adapters (one dedicated for the MAPI network and the other for a replication network). Example IP address configuration settings are shown in the following table.

Example network adapter settings

Server-network adapter	IP address/subnet mask	Default gateway
EX1-MAPI	192.168.1.15/24	192.168.1.1
EX1-Replication	10.0.0.15/24	Not applicable
EX2-MAPI	192.168.1.16	192.168.1.1
EX2-Replication	10.0.0.16	Not applicable

In the following configuration, there are two subnets configured in the DAG: 192.168.1.0 and 10.0.0.0. When EX1 and EX2 are added to the DAG, two subnets will be enumerated and two DAG networks will be created: MapiDagNetwork (192.168.1.0) and ReplicationDagNetwork01 (10.0.0.0). These networks will be configured as shown in the following table.

Enumerated DAG network settings for a single-subnet DAG

Name	Subnets	Interfaces	MAPI access enabled	Replication enabled
MapiDagNetwork	192.168.1.0/24	EX1 (192.168.1.15)	True	True

		EX2 (192.168.1.16)		
ReplicationDagNetwork01	10.0.0.0/24	EX1 (10.0.0.15) EX2 (10.0.0.16)	False	True

To complete the configuration of ReplicationDagNetwork01 as the dedicated replication network, disable replication for MapiDagNetwork by running the following command.

```
Set-DatabaseAvailabilityGroupNetwork -Identity DAG1
\MapiDagNetwork -ReplicationEnabled:$false
```

After replication is disabled for MapiDagNetwork, the Microsoft Exchange Replication service uses ReplicationDagNetwork01 for continuous replication. If ReplicationDagNetwork01 experiences a failure, the Microsoft Exchange Replication service reverts to using MapiDagNetwork for continuous replication. This is done intentionally by the system to maintain high availability.

DAG networks and multiple subnet deployments

In the preceding example, even though there are two different subnets in use by the DAG (192.168.1.0 and 10.0.0.0), the DAG is considered a single-subnet DAG because each member uses the same subnet to form the MAPI network. When DAG members use different subnets for the MAPI network, the DAG is referred to as a *multi-subnet DAG*. In a multi-subnet DAG, the proper subnets are automatically associated with each DAG network.

For example, consider DAG2, a two-member DAG where each member has two network adapters (one dedicated for the MAPI network and the other for a replication network), and each DAG member is located in a separate Active Directory site, with its MAPI network on a different subnet. Example IP address configuration settings are shown in the following table.

Example network adapter settings for a multi-subnet DAG

Server-network adapter	IP address/subnet mask	Default gateway
EX1-MAPI	192.168.0.15/24	192.168.0.1
EX1-Replication	10.0.0.15/24	Not applicable
EX2-MAPI	192.168.1.15	192.168.1.1
EX2-Replication	10.0.1.15	Not applicable

In the following configuration, there are four subnets configured in the DAG: 192.168.0.0, 192.168.1.0, 10.0.0.0, and 10.0.1.0. When EX1 and EX2 are added to the DAG, four subnets will be enumerated, but only two DAG networks will be created: MapiDagNetwork (192.168.0.0, 192.168.1.0) and ReplicationDagNetwork01 (10.0.0.0, 10.0.1.0). These networks will be configured as

shown in the following table.

Enumerated DAG network settings for a multi-subnet DAG

Name	Subnets	Interfaces	MAPI access enabled	Replication enabled
MapiDagNetwork	192.168.0.0/24 192.168.1.0/24	EX1 (192.168.0.15) EX2 (192.168.1.15)	True	True
ReplicationDagNetwork01	10.0.0.0/24 10.0.1.0/24	EX1 (10.0.0.15) EX2 (10.0.1.15)	False	True

DAG networks and iSCSI networks

By default, DAGs perform discovery of all networks detected and configured for use by the underlying cluster. This includes any Internet SCSI (iSCSI) networks in use as a result of using iSCSI storage for one or more DAG members. As a best practice, iSCSI storage should use dedicated networks and network adapters. These networks shouldn't be managed by the DAG or its cluster, or used as DAG networks (MAPI or replication). Instead, these networks should be manually disabled from use by the DAG, so they can be dedicated to iSCSI storage traffic. To disable iSCSI networks from being detected and used as DAG networks, configure the DAG to ignore any currently detected iSCSI networks using the Set-DatabaseAvailabilityGroupNetwork cmdlet, as shown in this example:

```
Set-DatabaseAvailabilityGroupNetwork -Identity DAG2
\DAGNetwork02 -ReplicationEnabled:$false -
IgnoreNetwork:$true
```

This command will also disable the network for use by the cluster. Although the iSCSI networks will continue to appear as DAG networks, they won't be used for MAPI or replication traffic after running the above command.

Return to top

Configuring DAG members

Mailbox servers that are members of a DAG have some properties specific to high availability that should be configured as described in the following sections:

- Automatic database mount dial
- Database copy automatic activation policy
- Maximum active databases

Automatic database mount dial

The *AutoDatabaseMountDial* parameter specifies the automatic database mount behavior after a database failover. You can use the Set-MailboxServer cmdlet to configure the *AutoDatabaseMountDial* parameter with any of the following values:

- `BestAvailability` If you specify this value, the database automatically mounts immediately after a failover if the copy queue length is less than or equal to 12. The copy queue length is the number of logs recognized by the passive copy that needs to be replicated. If the copy queue length is more than 12, the database doesn't automatically mount. When the copy queue length is less than or equal to 12, Exchange attempts to replicate the remaining logs to the passive copy and mounts the database.
- `GoodAvailability` If you specify this value, the database automatically mounts immediately after a failover if the copy queue length is less than or equal to six. The copy queue length is the number of logs recognized by the passive copy that needs to be replicated. If the copy queue length is more than six, the database doesn't automatically mount. When the copy queue length is less than or equal to six, Exchange attempts to replicate the remaining logs to the passive copy and mounts the database.
- `Lossless` If you specify this value, the database doesn't automatically mount until all logs generated on the active copy have been copied to the passive copy. This setting also causes the Active Manager best copy selection algorithm to sort potential candidates for activation based on the database copy's activation preference value and not its copy queue length.

The default value is `GoodAvailability`. If you specify either `BestAvailability` or `GoodAvailability`, and all the logs from the active copy can't be copied to the passive copy being activated, you may lose some mailbox data. However, the Safety Net feature (which is enabled by default) helps protect against most data loss by resubmitting messages that are in the Safety Net queue.

Example: configuring automatic database mount dial

The following example configures a Mailbox server with an *AutoDatabaseMountDial* setting of `GoodAvailability`.

```
Set-MailboxServer -Identity EX1 -AutoDatabaseMountDial
GoodAvailability
```

Database copy automatic activation policy

The *DatabaseCopyAutoActivationPolicy* parameter specifies the type of automatic activation available for mailbox database copies on the selected Mailbox servers. You can use the Set-MailboxServer cmdlet to configure the *DatabaseCopyAutoActivationPolicy* parameter with any of the following values:

- `Blocked` If you specify this value, databases can't be automatically activated on the selected Mailbox servers.
- `IntrasiteOnly` If you specify this value, the database copy is allowed to be activated on servers in the same Active Directory site. This prevents cross-site failover or activation. This property is for incoming mailbox database copies (for example, a passive copy being made an active copy). Databases can't be activated on this Mailbox server for database copies that are active in another Active Directory site.
- `Unrestricted` If you specify this value, there are no special restrictions on activating mailbox database copies on the selected Mailbox servers.

Example: configuring database copy automatic activation policy

The following example configures a Mailbox server with a *DatabaseCopyAutoActivationPolicy* setting of `Blocked`.

```
Set-MailboxServer -Identity EX1 -
DatabaseCopyAutoActivationPolicy Blocked
```

Maximum active databases

The *MaximumActiveDatabases* parameter (also used with the Set-MailboxServer cmdlet) specifies the number of databases that can be mounted on a Mailbox server. You can configure Mailbox servers to meet your deployment requirements by ensuring that an individual Mailbox server doesn't become overloaded.

The *MaximumActiveDatabases* parameter is configured with a whole number numeric value. When the maximum number is reached, the database copies on the server won't be activated if a failover or switchover occurs. If the copies are already active on a server, the server won't allow databases to be mounted.

Example: configuring maximum active databases

The following example configures a Mailbox server to support a maximum of 20 active databases.

```
Set-MailboxServer -Identity EX1 -MaximumActiveDatabases 20
```

Return to top

Performing maintenance on DAG members

Before performing any type of software or hardware maintenance on a DAG member, you should first place the DAG member into maintenance mode. This involves moving all active databases off the server and blocking active databases from moving to the server. It also ensures that all critical DAG support functionality that may be on the server (for example, the Primary Active Manager (PAM) role) is moved to another server and blocked from moving back to the server. Specifically, you should perform the following tasks:

1. To begin the process of draining the transport queues, run `Set-ServerComponentState <ServerName> -Component HubTransport -State Draining -Requester Maintenance`
2. To initiate the draining of the transport queues, run `Restart-Service MSExchangeTransport`
3. To begin the process of draining all Unified Messaging calls, run `Set-ServerComponentState <ServerName> -Component UMCallRouter -State Draining -Requester Maintenance`
4. To redirect messages pending delivery in the local queues to the Mailbox server specified by the Target parameter, run `Redirect-Message -Server <ServerName> -Target <MailboxServerFQDN>`
5. To pause the cluster node, which prevents the node from being and becoming the PAM, run `Suspend-ClusterNode <ServerName>`
6. To move all active databases currently hosted on the DAG member to other DAG members, run `Set-MailboxServer <ServerName> -DatabaseCopyActivationDisabledAndMoveNow $True`
7. To prevent the server from hosting active database copies, run `Set-MailboxServer <ServerName> -DatabaseCopyAutoActivationPolicy Blocked`
8. To place the server into maintenance mode, run `Set-ServerComponentState <ServerName> -Component ServerWideOffline -State Inactive -Requester Maintenance`

To verify that a server is ready for maintenance, perform the following tasks:

1. To verify the server has been placed into maintenance mode, run `Get-ServerComponentState <ServerName> | ft Component,State -Autosize`
2. To verify the server is not hosting any active database copies, run `Get-MailboxServer <ServerName> | ft DatabaseCopy* -Autosize`
3. To verify that the node is paused, run `Get-ClusterNode <ServerName> | fl`
4. To verify that all transport queues have been drained, run `Get-Queue`

After the maintenance is complete and the DAG member is ready to return to service, you can take the DAG member out of maintenance mode and put it back into production by performing the following tasks:

- To designate that the server is out of maintenance mode, run `Set-ServerComponentState <ServerName> -Component ServerWideOffline -State Active -Requester Maintenance`
- To allow the server to accept Unified Messaging calls, run `Set-ServerComponentState <ServerName> -Component UMCallRouter -State Active -Requester Maintenance`
- To resume the node in the cluster and enable full cluster functionality for the server, run `Resume-ClusterNode <ServerName>`
- To allow databases to become active on the server, run `Set-MailboxServer <ServerName> -DatabaseCopyActivationDisabledAndMoveNow $False`
- To remove the automatic activation blocks, run `Set-MailboxServer <ServerName> -DatabaseCopyAutoActivationPolicy Unrestricted`
- To enable the transport queues and allow the server to accept and process messages, run `Set-ServerComponentState <ServerName> -Component HubTransport -State Active -Requester Maintenance`
- To resume transport activity, run `Restart-Service MSExchangeTransport`

To verify that a server is ready for production use, perform the following tasks:

1. To verify the server is not maintenance mode, run `Get-ServerComponentState <ServerName> | ft Component,State -Autosize`

If you are installing an Exchange update, and the update process fails, it can leave some server components in an inactive state, which will be displayed in the output of the above Get-ServerComponentState cmdlet. To resolve this, run the following commands:

- `Set-ServerComponentState <ServerName> -Component ServerWideOffline -State Active -Requester Functional`
- `Set-ServerComponentState <ServerName> -Component Monitoring -State Active -Requester Functional`
- `Set-ServerComponentState <ServerName> -Component RecoveryActionsEnabled -State Active -Requester Functional`

Return to top

Shutting down DAG members

The Exchange 2013 high availability solution is integrated with the Windows shutdown process. If an administrator or application initiates a shutdown of a Windows server in a DAG that has a mounted database that's replicated to one or more DAG members, the system attempts to activate another copy of the mounted database prior to allowing the shutdown process to complete.

However, this new behavior doesn't guarantee that all of the databases on the server being shut down will experience a `lossless` activation. As a result, it's a best practice to perform a server switchover prior to shutting down a server that's a member of a DAG.

Return to top

Installing updates on DAG members

Installing Microsoft Exchange Server 2013 updates on a server that's a member of a DAG is a relatively straightforward process. When you install an update on a server that's a member of a DAG, several services are stopped during the installation, including all Exchange services and the Cluster service. The general process for applying updates to a DAG member is as follows:

1. Use the steps described above to put the DAG member in maintenance mode.
2. Install the update.
3. Use the steps described above to take the DAG member out of maintenance mode and put it back into production.
4. Optionally, use the RedistributeActiveDatabases.ps1 script to rebalance the active database copies across the DAG.

You can download the latest update for Exchange 2013 from the Microsoft Download Center.

Return to top

Create a database availability group

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-21*

A database availability group (DAG) is a set of up to 16 Microsoft Exchange Server 2013 Mailbox servers that provide automatic database-level recovery from a database, server, or network failure. When a Mailbox server is added to a DAG, it works with the other servers in the DAG to provide automatic, database-level recovery from database, server, and network failures.

Looking for other management tasks related to DAGs? Check out Managing database availability groups.

What do you need to know before you begin?

- Estimated time to complete: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.
- When creating a DAG with Mailbox servers running Windows Server 2012, you must pre-stage the cluster name object (CNO) before adding members to the DAG. If you are creating a DAG without an administrative access point with Mailbox servers running Windows Server 2012 R2, then you do not need to pre-stage a CNO for the DAG. For detailed steps, see Pre-stage the cluster name object for a database availability group.
- When creating a DAG, you provide a unique name for the DAG of up to 15 characters. In addition to providing a name for the DAG, you must also assign one or more IP addresses (either IPv4 or both IPv4 and IPv6) to the DAG, unless you are creating a Windows Server 2012 R2 DAG without an administrative access point and you are not assigning any IP addresses to the DAG. Otherwise, the IP addresses you assign must be on each subnet intended for the MAPI network and must be available for use. If you specify one or more IPv4 addresses and your system is configured to use IPv6, the task will also attempt to automatically assign the DAG one or more IPv6 addresses.
- When creating a DAG, you can optionally specify a witness server and witness directory. If you specify a witness server, we recommend that you use a Client Access server that doesn't have the Mailbox server role installed. This allows an Exchange administrator to be aware of the availability of the witness, and it ensures that all of the necessary security permissions needed for using the witness server are in place.

The following combinations of options and behaviors are available:

  - You can specify only a name for the DAG and leave the **Witness server** and **Witness directory** fields empty. In this scenario, the task will search for a Client Access server that doesn't have the

Mailbox server role installed. It will automatically create the default witness directory and share on that Client Access server and configure the DAG to use that server as its witness server.

- You can specify a name for the DAG, the witness server that you want to use, and the directory you want created and shared on the witness server.
- You can specify a name for the DAG and the witness server that you want to use, and leave the **Witness directory** field empty. In this scenario, the task will create the default witness directory on the specified witness server.
- You can specify a name for the DAG, leave the **Witness server** field empty, and specify the directory you want created and shared on the witness server. In this scenario, the wizard will search for a Client Access server that doesn't have the Mailbox server role installed, and it will automatically create the specified witness directory on that server, share the directory, and configure the DAG to use that Client Access server as its witness server.

Important:

If the witness server you specify isn't an Exchange 2013 or Exchange 2010 server, you must add the Exchange Trusted Subsystem universal security group to the local Administrators group on the witness server. These security permissions are necessary to ensure that Exchange can create a directory and share on the witness server as needed. If the proper permissions aren't configured, the following error is returned:

```
Error: An error occurred during discovery of the database availability group
topology. Error: An error occurred while attempting a cluster operation. Error:
Cluster API "AddClusterNode() (MaxPercentage=12) failed with 0x80070005. Error:
Access is denied."
```

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to create a database availability group

1. In the EAC, go to **Servers** > **Database Availability Groups**.
2. Click **+** to create a DAG.
3. On the **new database availability group** page, provide the following information for the DAG:
   - **Database availability group name** Use this field to type a valid and unique name for the DAG of up to 15 characters. The name is equivalent to a computer name, and a corresponding CNO will be created in Active Directory with that name. This name will be both the name of the DAG and the name of the underlying cluster.
   - **Witness server** Use this field to specify a witness server for the DAG. If you leave this field blank, the system will attempt to automatically select a Client Access server in the local Active Directory site that isn't installed on a computer with the Mailbox server to be used as the witness server.

Note:

If you specify a witness server, you must use either a host name or a fully qualified domain name (FQDN). Using an IP address or a wildcard name isn't supported. In addition, the witness server can't be a member of the DAG.

- **Witness directory** Use this field to type the path to a directory on the witness server that will be used to store witness data. If the directory doesn't exist, the system will create it for you on the witness server. If you leave this field blank, the default directory (%SystemDrive%\DAGFileShareWitnesses\<DAG FQDN>) will be created on the witness server.
- **Database availability group IP addresses** Use this field to assign one or more static IPv4 addresses to the DAG. Enter an IPv4 address and click **+** to add it. Leave this field blank if you want the DAG to use Dynamic Host Configuration Protocol (DHCP) to obtain the necessary IPv4 addresses. Optionally, enter 255.255.255.255 to create a DAG without an IP address or cluster administrative access point, which applies only to DAGs that will contain Mailbox servers running Windows Server 2012 R2.

4. Click **Save** to create the DAG.

Use the Shell to create a database availability group

This example creates the DAG DAG1 that's configured to use the witness server FILESRV1 and the local directory C:\DAG1. DAG1 is also configured to use DHCP for the DAG's IP addresses.

```
New-DatabaseAvailabilityGroup -Name DAG1 -WitnessServer
FILESRV1 -WitnessDirectory C:\DAG1
```

This example creates the DAG DAG2. The system automatically selects a Client Access server in the local Active Directory site that does not contain the Mailbox server role as the DAG's witness server. DAG2 is assigned a single static IP address because in this example all DAG members have the MAPI network on the same subnet.

```
New-DatabaseAvailabilityGroup -Name DAG2 -
DatabaseAvailabilityGroupIPAddresses 10.0.0.8
```

This example creates the DAG DAG3. DAG3 is configured to use the witness server MBX2 and the local directory C:\DAG3. DAG3 is assigned multiple static IP addresses because its DAG members are on different subnets on the MAPI network.

```
New-DatabaseAvailabilityGroup -Name DAG3 -WitnessServer
MBX2 -WitnessDirectory C:\DAG3 -
DatabaseAvailabilityGroupIPAddresses 10.0.0.8,192.168.0.8
```

This example creates the DAG DAG4 that's configured to use DHCP. In addition, the witness server will be automatically selected by the system, and the default witness directory will be created.

```
New-DatabaseAvailabilityGroup -Name DAG4
```

This example creates the DAG DAG5 that will not have an administrative access point (valid for Windows Server 2012 R2 DAGs only). In addition, MBX4 will be used as the witness server for the DAG, and the default witness directory will be created.

```
New-DatabaseAvailabilityGroup -Name DAG5 -
DatabaseAvailabilityGroupIPAddresses
([System.Net.IPAddress]::None) -WitnessServer MBX4
```

How do you know this worked?

To verify that you've successfully created a DAG, do one of the following:

- In the EAC, navigate to **Servers** > **Database Availability Groups**. The newly created DAG is displayed.
- In the Shell, run the following command to verify the DAG was created and to display DAG property information.

```
Get-DatabaseAvailabilityGroup <DAGName> | Format-List
```

For more information

Database availability groups

Configure database availability group properties

Set-DatabaseAvailabilityGroup

New-DatabaseAvailabilityGroup

New-DatabaseAvailabilityGroupNetwork

Add-DatabaseAvailabilityGroupServer

Pre-stage the cluster name object for a database availability group

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-14*

In environments where computer account creation is restricted or where computer accounts are created in a container other than the default computers container, you can pre-stage the cluster name object (CNO) and then provision the CNO by assigning permissions to it. Pre-staging the CNO is also required for Windows Server 2012 and Windows Server 2012 R2 DAG members due to permissions changes in Windows for computer objects. When deploying a database availability group (DAG) using Mailbox servers that are running Windows Server 2012 or Windows Server 2012 R2, you must pre-stage and provision the CNO, unless you are deploying a DAG without a cluster administrative access point. DAGs without cluster administrative access points do not use CNOs; therefore pre-staging is not required for those DAGs.

You create and disable a computer account for the CNO, and then either:

- Assign full control of the computer account to the computer account of the first Mailbox server you're adding to the DAG.
- Assign full control of the computer account to the Exchange Trusted Subsystem universal security group (USG).

What do you need to know before you begin?

- Estimated time to complete: 1 minute
- You must use an account that has permissions to create computer objects in Active Directory.
- After completing the following steps, allow time for Active Directory replication to occur. After the object is replicated, you can add the first member to the DAG.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Pre-stage the CNO

1. Open Active Directory Users and Computers.
2. Expand the forest node.
3. Right-click the organizational unit (OU) in which you want to create the new account, select **New**, and then select **Computer**.
4. In **New Object - Computer**, type the computer account name for the CNO in the **Computer name** box. This is the name that you'll use for the DAG. Click **OK** to create the account.
5. Right-click the new computer account, and then click **Disable Account**. Click **Yes** to confirm the disable action, and then click **OK**.

Assign permissions to the CNO

1. Open Active Directory Users and Computers.
2. If Advanced Features aren't enabled, turn them on by clicking **View**, and then clicking **Advanced**

Features.

3. Right-click the new computer account, and then click **Properties**.
4. In **<Computer Name> Properties**, on the **Security** tab, click **Add** to add either the computer account for the first node to be added to the DAG or to add the Exchange Trusted Subsystem USG:
   - To add the Exchange Trusted Subsystem, type **Exchange Trusted Subsystem** in the **Enter the object names to select** field. Click **OK** to add the USG. Select the Exchange Trusted Subsystem USG and in the **Permissions for Exchange Trusted Subsystem** field, select **Full Control** in the **Allow** column. Click **OK** to save the permission settings.
   - To add the computer account for the first node to be added to the DAG, click **Object Types**. In the **Object Types** dialog box, clear the **Built-in security principals**, **Groups**, and **Users** check boxes. Select the **Computers** check box and click **OK**. In the **Enter the object names to select** field, type the name of the first Mailbox server to be added to the DAG, and then click **OK**. Select the first node's computer account, and in the **Permissions for <NodeName>** field, select **Full Control** in the **Allow** column. Click **OK** to save the permission settings.

How do you know this worked?

To verify that you've successfully created the CNO, do the following:

1. Open Active Directory Users and Computers.
2. Expand the forest node.
3. Open the OU in which you created the account, and then verify that the account is listed.

Configure database availability group properties

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-24*

You can use the EAC or the Shell to configure the properties of a database availability group (DAG), including DAG IP address configuration and the witness server and witness directory used by the DAG. The Shell enables you to configure DAG properties that aren't available in the EAC, such as alternate witness server and alternate witness directory information, the TCP port used for replication, and datacenter activation coordination (DAC) mode.

What do you need to know before you begin?

- Estimated time to complete: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.
- DAG property values are stored in both Active Directory and the cluster database. However, some properties are stored only in the cluster database. As a result, the underlying cluster for the DAG must be running and have quorum to set the properties for:
  - ReplicationPort
  - NetworkCompression
  - NetworkEncryption
  - DiscoverNetworks
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure database availability group properties

1. In the EAC, go to **Servers** > **Database Availability Groups**.
2. Select the DAG you want to configure and click ✎.
3. Use the **General** page to view DAG membership and operational status, and to configure the DAG's witness server, witness directory, and automatic network configuration:
   - **Witness server** The host name or fully qualified domain name (FQDN) of the witness server for the DAG. Although this is a required property for all DAGs, the witness server is used when there is an even number of DAG members and the quorum model in use by the cluster is Node and File Share Majority.
   - **Witness directory** The full path of the directory used to store the witness.log file on the witness server. Although this is a required property for all DAGs, the witness directory is used only when the DAG's witness server is in use.
   - **Operational servers** A read-only field that displays a list of DAG members and their current operational status.
   - **Configure the database group network manually** A check box that you select when you want to configure all DAG networks manually. When you leave the check box clear, the system configures DAG networks automatically based on network interface configuration. If the check box is clear, the **Set-DatabaseAvailabilityGroupNetwork** and **New-DatabaseAvailabilityGroupNetwork** cmdlets are disabled for administrative use against the DAG.

4. Use the **IP Addresses** page to view and modify the IP addresses assigned to the DAG:
   - Select an existing IP address and click ✎ to modify it.
   - Select an existing IP address and click the minus icon (delete) to remove it.
   - Enter an IP address and click **+** to add it to the DAG.
5. Click **Save** to save any changes that were made.

Use the Shell to configure database availability group properties

This example sets the witness directory to C:\DAG1DIR for the DAG DAG1.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
WitnessDirectory C:\DAG1DIR
```

This example preconfigures an alternate witness server of CAS3 and an alternate witness directory of C:\DAGFileShareWitnesses\DAG1.contoso.com for the DAG DAG1.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
AlternateWitnessDirectory C:\DAGFileShareWitnesses
\DAG1.contoso.com -AlternateWitnessServer CAS3
```

This example configures the DAG DAG1 to use Dynamic Host Configuration Protocol (DHCP) to obtain an IP address.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
DatabaseAvailabilityGroupIPAddresses 0.0.0.0
```

This example configures the DAG DAG1 to use a static IP address of 10.0.0.8.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
DatabaseAvailabilityGroupIPAddresses 10.0.0.8
```

This example configures the multi-subnet DAG DAG1 with multiple static IP addresses.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
DatabaseAvailabilityGroupIPAddresses 10.0.0.8,10.0.1.8
```

This example configures the DAG DAG1 for DAC mode.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
DatacenterActivationMode DagOnly
```

This example configures the replication port for the DAG DAG1 to be 63132.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
ReplicationPort 63132
```

Note:
After changing the default replication port for a DAG, you must manually modify the Windows Firewall exceptions on each member of the DAG to allow communication to occur over the specified port.

How do you know this worked?

To verify that you've successfully configured the DAG, do the following:

- In the Shell, run the following command to display DAG configuration settings and verify the DAG was configured successfully.

```
Get-DatabaseAvailabilityGroup <DAGName> | Format-List
```

For more information

Create a database availability group

Remove a database availability group

Create a database availability group network

Manage database availability group membership

Get-DatabaseAvailabilityGroup

Set-DatabaseAvailabilityGroup

Manage database availability group membership

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-13*

When you add a server to a database availability group (DAG), it works with the other servers in the DAG to provide automatic database-level recovery from database, server, or network failures. When you remove a server from a DAG, it's no longer automatically protected from failures.

Looking for other management tasks related to DAGs? Check out Managing database availability

groups.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes per server
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.
- DAGs use Windows Failover Clustering (WFC) technologies. Each Mailbox server that's a member of a DAG is also a node in the underlying cluster used by the DAG. As a result, at any specific time, a Mailbox server can be a member of only one DAG. Because DAGs use WFC technology, all servers added to a DAG must be running the same operating system: either Windows Server 2008 R2 Enterprise or Datacenter Edition, or the Standard or Datacenter Edition of Windows Server 2012 or Windows Server 2012 R2.
- If you're adding Mailbox servers running Windows Server 2012, you must pre-stage the cluster name object (CNO) for the DAG. If you're adding Mailbox servers running Windows Server 2012 R2, and your DAG does not have an administrative access point, then you do not need to pre-stage a CNO, as DAGs without administrative access points do not have a CNO. For detailed steps, see Pre-stage the cluster name object for a database availability group.
- Before you can add members to a DAG, you must first create a DAG. For detailed steps, see Create a database availability group.
- You must remove all replicated database copies from the server before you can remove it from a DAG.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to manage database availability group membership

1. In the EAC, go to **Servers** > **Database Availability Groups**.
2. Select the DAG you want to configure, and then click .
   - To add one or more Mailbox servers to the DAG, click **+**, select the servers from the list, click **Add**, and then click **OK**.
   - To remove one or more Mailbox servers from the DAG, select the servers, and then click the minus (-) icon.
3. Click **Save** to save the changes.

4. When the task has completed successfully, click **Close**.

Use the Shell to manage database availability group membership

This example adds the Mailbox server MBX1 to the DAG DAG1.

```
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer MBX1
```

This example removes the Mailbox server MBX1 from the DAG DAG1. Before running this command, make sure that no replicated databases exist on the Mailbox server.

```
Remove-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer MBX1
```

This example removes the configuration settings for the Mailbox server MBX4 from the DAG DAG2. MBX4 is expected to be offline for an extended period, so its configuration is being removed from the DAG while it's offline to establish quorum with the remaining online DAG members.

```
Remove-DatabaseAvailabilityGroupServer -Identity DAG2 -
MailboxServer MBX4 -ConfigurationOnly
```

How do you know this worked?

To verify that you've successfully managed DAG membership, do one of the following:

- In the EAC, navigate to **Servers** > **Database Availability Groups**. The current DAG membership is displayed in the **Member Servers** column.
- In the Shell, run the following command to display DAG membership information.

```
Get-DatabaseAvailabilityGroup <DAGName> | Format-List
Servers
```

For more information

Add-DatabaseAvailabilityGroupServer

Remove-DatabaseAvailabilityGroupServer

Create a database availability group

network

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-02*

If needed, you can create additional networks for use in a database availability group (DAG). You can use the EAC or the Shell to create a DAG network.

Looking for other management tasks related to DAGs? Check out Managing database availability groups.

What do you need to know before you begin?

- Estimated time to complete: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.
- You can create a DAG network only when automatic network configuration has been disabled for a DAG. For detailed steps about how to disable automatic network configuration for a DAG, see Configure database availability group properties.
- When creating a DAG network, you must assign unique subnets that aren't in use by another DAG network. If you use subnets that are assigned to an existing DAG network, they will be removed from that DAG network and added to the newly created DAG network.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to create a database availability group network

1. In the EAC, go to **Servers** > **Database Availability Groups**.
2. Select the DAG you want to configure, and then click .
3. On the **new database availability group network** page, provide the following information:
   - **Database availability group network name** Use this field to type a name for the network

that's unique in the DAG.

- **Description** Use this field to provide a text description of the DAG network.
- **Subnets** Use this field to associate one or more subnets with the DAG network. Click **+** to add a subnet, click ✎ to edit a subnet, and click minus (-) to remove a subnet.

4. Click **Save** to create the DAG network.

Use the Shell to create a database availability group network

This example creates the network ReplicationDagNetwork02 with a subnet of 10.0.0.0 and a bitmask of 8 in the DAG DAG1. Replication is enabled for the network, and an optional description of the network is also being added.

```
New-DatabaseAvailabilityGroupNetwork -
DatabaseAvailabilityGroup DAG1 -Name
ReplicationDagNetwork02 -Description "Replication network
2" -Subnets 10.0.0.0/8 -ReplicationEnabled:$True
```

How do you know this worked?

To verify that you've successfully created a DAG network, do one of the following:

- In the EAC, navigate to **Servers** > **Database Availability Groups**. Select the appropriate DAG, and the newly created DAG network is displayed in the details pane.
- In the Shell, run the following command to verify the DAG network was created and to display DAG network configuration information.

```
Get-DatabaseAvailabilityGroupNetwork <DAGNetworkName> |
Format-List
```

For more information

Set-DatabaseAvailabilityGroupNetwork

Get-DatabaseAvailabilityGroupNetwork

New-DatabaseAvailabilityGroupNetwork

Remove-DatabaseAvailabilityGroupNetwork

Configure database availability group

network properties

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-31*

Each database availability group (DAG) network has several properties that you can configure, including the name of the DAG network, a description field for the DAG network, a list of subnets that are used by the DAG network, and whether the DAG network is enabled for replication.

Looking for other management tasks related to DAGs? Check out Managing database availability groups.

What do you need to know before you begin?

- Estimated time to complete: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.
- You can configure a DAG network only when automatic network configuration has been disabled for a DAG. For detailed steps about how to disable automatic network configuration for a DAG, see Configure database availability group properties.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure database availability group network properties

1. In the EAC, go to **Servers** > **Database Availability Groups**.
2. Select the DAG you want to configure, and in the Details pane, under the DAG network you want to configure, choose:
   - **Disable Replication** or **Enable Replication** Configures the replication settings for the DAG network.
   - **Remove** Removes a DAG network. Before you can remove a DAG network, you must first

remove all associated subnets from the DAG network.

- **View details** Configures DAG network properties, such as the name, description, and associated subnets for the DAG network. You can also view the network interfaces associated with those subnets, and enable or disable replication for the DAG network.

Use the Shell to configure database availability group network properties

This example adds a subnet of 10.0.0.0 and subnet mask of 255.0.0.0 to the DAG network MapiDagNetwork in the DAG DAG1.

```
Set-DatabaseAvailabilityGroupNetwork -Subnets 10.0.0.0/8 -
Identity DAG1\MapiDagNetwork
```

How do you know this worked?

To verify that you've successfully configured the DAG network, do the following:

- In the Shell, run the following command to display DAG network configuration settings and verify the DAG network was configured successfully.

```
Get-DatabaseAvailabilityGroupNetwork <DAGNetworkName> |
Format-List
```

For more information

Set-DatabaseAvailabilityGroupNetwork

Get-DatabaseAvailabilityGroupNetwork

New-DatabaseAvailabilityGroupNetwork

Remove-DatabaseAvailabilityGroupNetwork

Configure AutoReseed for a database availability group

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-04-15*

AutoReseed is a feature for quickly restoring database redundancy after a disk failure. If a disk fails, the database copies stored on that disk are automatically reseeded to a preconfigured spare disk on the Mailbox server. You can use the steps in this topic to configure AutoReseed for a database availability group (DAG).

Caution:

The AutoReseed feature doesn't perform any prerequisite configuration tasks for you. Installing disks correctly, adding spare disks to the system, replacing bad disks, and formatting new disks must be done manually by an administrator.

For additional management tasks related to DAGs, see Managing database availability groups.

What do you need to know before you begin?

- Estimated time to complete this task: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.
- A single logical disk/partition per physical disk must be created.
- The specific database and log folder structure described in the steps below must be used.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

How do you do this?

Step 1: Configure the root paths for databases and volumes

The first step involves configuring the root directories for the databases (*AutoDagDatabasesRootFolderPath*) and volumes (*AutoDagVolumesRootFolderPath*) used by the DAG. The defaults are C:\ExchangeDatabases, and C:\ExchangeVolumes, respectively. You can omit this step if you're using the default paths.

This example illustrates how to configure the root path for the databases.

```
Set-DatabaseAvailabilityGroup DAG1 -
AutoDagDatabasesRootFolderPath "C:\ExchDbs"
```

This example illustrates how to configure the root path for the storage volumes.

```
Set-DatabaseAvailabilityGroup DAG1 -
AutoDagVolumesRootFolderPath "C:\ExchVols"
```

How do you know this step worked?

To verify that you've successfully configured the root paths for databases and volumes, run the following command.

```
Get-DatabaseAvailabilityGroup DAG1 | Format-List *auto*
```

The output for *AutoDagDatabasesRootFolderPath* and *AutoDagVolumesRootFolderPath* should reflect the configured paths.

Step 2: Configure the number of databases per volume

Next, configure the number of databases per volume (*AutoDagDatabaseCopiesPerVolume*) for the DAG.

This example illustrates how to configure this AutoReseed setting for a DAG configured with 4 databases per volume.

```
Set-DatabaseAvailabilityGroup DAG1 -
AutoDagDatabaseCopiesPerVolume 4
```

How do you know this step worked?

To verify that you've successfully configured the number of databases per volume, run the following command.

```
Get-DatabaseAvailabilityGroup DAG1 | Format-List *auto*
```

The output for *AutoDagDatabaseCopiesPerVolume* should reflect the configured value.

Step 3: Create the root directories for databases and volumes

Next, create the directories that correspond to the root directories you configured in Step 1. This example shows how to create the default directories using the command prompt.

```
md C:\ExchangeDatabases
md C:\ExchangeVolumes
```

How do you know this step worked?

To verify that you've successfully configured the root directories for databases and volumes, run the following command.

```
Dir C:\
```

The created directories should appear in the output list.

Step 4: Mount the volume folders

For every volume that will be used for databases (including spare volumes), use the Windows Disk Management application (diskmgmt.msc) to mount each volume in a mounted folder under C:\ExchangeVolumes\. For example, if there are 2 volumes with databases and 1 spare volume, mount the volumes to the following mounted folders:

- C:\ExchangeVolumes\Volume1
- C:\ExchangeVolumes\Volume2
- C:\ExchangeVolumes\Volume3

The names of the mounted folders can be any folder name, as long as the folders are mounted under the root volume's path.

How do you know this step worked?

To verify that you've successfully mounted the volume folders, run the following command.

```
Dir C:\
```

The mounted volumes should appear in the output list.

Step 5: Create the database folders

Next, create the database directories under the root path C:\ExchangeDatabases. This example illustrates how to create directories for a storage configuration with 4 databases on each volume.

```
md c:\ExchangeDatabases\db001

md c:\ExchangeDatabases\db002

md c:\ExchangeDatabases\db003

md c:\ExchangeDatabases\db004
```

How do you know this step worked?

To verify that you've successfully mounted the database folders, run the following command.

```
Dir C:\ExchangeDatabases
```

The created directories should appear in the output list.

Step 6: Create the mount points for the databases

Create the mount points for each database and link the mount point to the correct volume. For example, the mounted folder for db001 should be at C:\ExchangeDatabases\db001. You can use diskmgmt.msc or mountvol.exe to do this. This example illustrates how to mount db001 to C:\ExchangeDatabases\db001 using mountvol.exe.

```
Mountvol.exe c:\ExchangeDatabases\db001 \\?\Volume (GUID)
```

How do you know this step worked?

To verify that you've successfully created the mount points for the database, run the following command.

```
Mountvol.exe C:\ExchangeDatabases\db001 /L
```

The mounted volume should appear in the mount point list.

Step 7: Create the database directory structure

Next, create two directories underneath the folders you created in Step 5, one for each database and one for each of the database's log stream that will be stored on the same volume. You must use the following format for your directory structure:

C:\< *DatabaseFolderName*>*DatabaseName*\<*DatabaseName*>.db

C:\< *DatabaseFolderName*>*DatabaseName*\<*DatabaseName*>.log

This example illustrates how to create directories for 4 databases that will be stored on Volume 1:

```
md c:\ExchangeDatabases\db001\db001.db

md c:\ExchangeDatabases\db001\db001.log

md c:\ExchangeDatabases\db002\db002.db

md c:\ExchangeDatabases\db002\db002.log

md c:\ExchangeDatabases\db003\db003.db
```

```
md c:\ExchangeDatabases\db003\db003.log

md c:\ExchangeDatabases\db004\db004.db

md c:\ExchangeDatabases\db004\db004.log
```

Repeat the preceding commands for databases on every volume.

How do you know this step worked?

To verify that you've successfully created the database directory structure, run the following command.

```
Dir C:\ExchangeDatabases /s
```

The created directories should appear in the output list.

Step 8: Create databases

Create databases with log and database paths configured with the appropriate folders. This example illustrates how to create a database that's stored in the newly created directory and mount point structure.

```
New-MailboxDatabase -Name db001 -Server MBX1 -LogFolderPath
C:\ExchangeDatabases\db001\db001.log -EdbFilePath C:
\ExchangeDatabases\db001\db001.db\db001.edb
```

How do you know this step worked?

To verify that you've successfully created databases in the appropriate folder, run the following command.

```
Get-MailboxDatabase db001 | Format List *path*
```

Database properties that are returned should indicate that the database file and log files are being stored in the above folders.

How do you know this task worked?

To verify that you've configured AutoReseed for a DAG, do the following:
1. Run the following command to verify the DAG is configured correctly.

```
Get-DatabaseAvailabilityGroup DAG1 | Format-List *auto*
```

2. Run the following command to verify the directory structure is configured correctly (below are

the default paths; if necessary, substitute the paths for the paths you're using).

```
Dir c:\ExchangeDatabases /s

Dir c:\ExchangeVolumes /s
```

Remove a database availability group

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-16*

Removing a database availability group (DAG) is a quick and easy task. You can use the EAC or the Shell to remove a DAG.

Looking for other management tasks related to DAGs? Check out Managing database availability groups.

What do you need to know before you begin?

- Estimated time to complete: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.
- Before you can remove a DAG, the DAG must be empty. If the DAG you want to remove contains any Mailbox servers, you must first remove the servers from the DAG. For detailed steps about how to remove a Mailbox server from a DAG, see Manage database availability group membership.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to remove a database availability group

1. Navigate to **Servers** > **Database availability groups**.

2. Select the DAG you want to remove and click **Delete** 🗑.
3. Click **Yes** to confirm the warning and remove the DAG.

Use the Shell to remove a database availability group

This example removes the DAG DAG1.

```
Remove-DatabaseAvailabilityGroup -Identity DAG1
```

How do you know this worked?

To verify that you've successfully removed the DAG, do one of the following:

- In the EAC, go to **Servers** > **Database Availability Groups**, and see if the DAG is still displayed.
- In the Shell, run the following command to see if the DAG still exists:

```
Get-DatabaseAvailabilityGroup <DAGName>
```

If the DAG was successfully deleted, the preceding command will produce an error message indicating the object could not be found.

Managing mailbox database copies

Exchange Server 2013 > High availability and site resilience > Managing high availability and site resilience >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-07*

After a database availability group (DAG) has been created, configured, and populated with Mailbox server members, you can use the Exchange Admin Center (EAC) or the Exchange Management Shell to add mailbox database copies in a flexible and granular way.

Managing database copies

After multiple copies of a database are created, you can use the EAC or the Shell to monitor the health and status of each copy and to perform other management tasks associated with database copies. Some of the management tasks you may need to perform include suspending or resuming a database copy, seeding a database copy, monitoring database copies, configuring database copy settings, and removing a database copy.

Suspending and resuming database copies

For a variety of reasons, such as performing planned maintenance, it may be necessary to suspend and resume continuous replication activity for a database copy. In addition, some administrative tasks, such as seeding, require you to first suspend a database copy. We recommend that all replication activity be suspended when the path for the database or its log files is being changed. You can suspend and resume database copy activity by using the EAC, or by running the **Suspend-MailboxDatabaseCopy** and **Resume-MailboxDatabaseCopy** cmdlets in the Shell. For detailed steps about how to suspend or resume continuous replication activity for a database copy, see Suspend or resume a mailbox database copy.

Seeding a database copy

Seeding, also known as *updating*, is the process in which a database, either a blank database or a copy of the production database, is added to the target copy location on another Mailbox server in the same DAG as the active database. This becomes the baseline database for the copy maintained by that server.

Depending on the situation, seeding can be an automatic process or a manual process that you initiate. When a database copy is added, the copy will be automatically seeded, provided that the target server and its storage are properly configured. If you want to manually seed a database copy and don't want automatic seeding to occur when creating the copy, you can use the *SeedingPostponed* parameter when running the Add-MailboxDatabaseCopy cmdlet.

Database copies rarely need to be reseeded after the initial seeding has occurred. But if reseeding is necessary, or if you want to manually seed a database copy instead of having the system automatically seed the copy, these tasks can be performed by using the Update Mailbox Database Copy wizard in the EAC or by using the Update-MailboxDatabaseCopy cmdlet in the Shell. Before seeding a database copy, you must first suspend the mailbox database copy. For detailed steps about how to seed a database copy, see Update a mailbox database copy.

After a manual seed operation has completed, replication for the seeded mailbox database copy is automatically resumed. If you don't want replication to automatically resume, you can use the *ManualResume* parameter when running the Update-MailboxDatabaseCopy cmdlet.

Choosing what to seed

When performing a seed operation, you can choose to seed the mailbox database copy, the content index catalog for the mailbox database copy, or both the database copy and the content index catalog copy. The default behavior of the Update Mailbox Database Copy wizard and the Update-MailboxDatabaseCopy cmdlet is to seed both the mailbox database copy and the content index catalog copy. To seed just the mailbox database copy without seeding the content index catalog, use the *DatabaseOnly* parameter when running the Update-MailboxDatabaseCopy cmdlet. To seed just the content index catalog copy, use the *CatalogOnly* parameter when running the Update-MailboxDatabaseCopy cmdlet.

Selecting the seeding source

Any healthy database copy can be used as the seeding source for an additional copy of that database. This is particularly useful when you have a DAG that has been extended across multiple physical locations. For example, consider a four-member DAG deployment, where two members (MBX1 and MBX2) are located in Portland, Oregon and two members (MBX3 and MBX4) are located in New York, New York. A mailbox database named DB1 is active on MBX1 and there are passive copies of DB1 on MBX2 and MBX3. When adding a copy of DB1 to MBX4, you have the option of using the copy on MBX3 as the source for seeding. In doing so, you avoid seeding over the wide area network (WAN) link between Portland and New York.

To use a specific copy as a source for seeding when adding a new database copy, you would do the following:

- Use the *SeedingPostponed* parameter when running the Add-MailboxDatabaseCopy cmdlet to add the database copy. If the *SeedingPostponed* parameter isn't used, the database copy will be explicitly seeded using the active copy of the database as the source.
- You can specify the source server you want to use as part of the Update Mailbox Database Copy wizard in the EAC, or you can use the *SourceServer* parameter when running the Update-MailboxDatabaseCopy cmdlet to specify the desired source server for seeding. In the preceding example, you would specify MBX3 as the source server. If the *SourceServer* parameter isn't used, the database copy will be explicitly seeded from the active copy of the database.

Seeding and networks

In addition to selecting a specific source server for seeding a mailbox database copy, you can also use the Shell to specify which DAG networks to use, and optionally override the DAG network's compression and encryption settings during the seed operation.

To specify the networks you want to use for seeding, use the *Network* parameter when running the Update-MailboxDatabaseCopy cmdlet and specify the DAG networks that you want to use. If you don't use the *Network* parameter, the system uses the following default behavior for selecting a network to use for the seeding operation:

- If the source server and target server are on the same subnet and a replication network has been configured that includes the subnet, the replication network will be used.
- If the source server and target server are on different subnets, even if a replication network that contains those subnets has been configured, the client (MAPI) network will be used for seeding.
- If the source server and target server are in different datacenters, the client (MAPI) network will be used for seeding.

At the DAG level, DAG networks are configured for encryption and compression. The default settings are to use encryption and compression only for communications on different subnets. If the source and target are on different subnets and the DAG is configured with the default values for *NetworkCompression* and *NetworkEncryption*, you can override these values by using the *NetworkCompressionOverride* and *NetworkEncryptionOverride* parameters, respectively, when running the Update-MailboxDatabaseCopy cmdlet.

Seeding process

When you initiate a seeding process by using the Add-MailboxDatabaseCopy or Update-MailboxDatabaseCopy cmdlets, the following tasks are performed:

1. Database properties from Active Directory are read to validate the specified database and servers, and to verify that the source and target servers are running Exchange 2013, they are both members of the same DAG, and that the specified database isn't a recovery database. The database file paths are also read.
2. Preparations occur for reseed checks from the Microsoft Exchange Replication service on the target server.
3. The Microsoft Exchange Replication service on the target server checks for the presence of database and transaction log files in the file directories read by the Active Directory checks in step 1.
4. The Microsoft Exchange Replication service returns the status information from the target server to the administrative interface from where the cmdlet was run.
5. If all preliminary checks have passed, you're prompted to confirm the operation before continuing. If you confirm the operation, the process continues. If an error is encountered during the preliminary checks, the error is reported and the operation fails.
6. The seed operation is started from the Microsoft Exchange Replication service on the target server.
7. The Microsoft Exchange Replication service suspends database replication for the active database copy.
8. The state information for the database is updated by the Microsoft Exchange Replication service to reflect a status of Seeding.
9. If the target server doesn't already have the directories for the target database and log files, they are created.
10. A request to seed the database is passed from the Microsoft Exchange Replication service on the target server to the Microsoft Exchange Replication service on the source server using TCP. This request and the subsequent communications for seeding the database occur on a DAG network that has been configured as a replication network.
11. The Microsoft Exchange Replication service on the source server initiates an Extensible Storage Engine (ESE) streaming backup via the Microsoft Exchange Information Store service interface.
12. The Microsoft Exchange Information Store service streams the database data to the Microsoft Exchange Replication service.
13. The database data is moved from the source server's Microsoft Exchange Replication service to the target server's Microsoft Exchange Replication service.
14. The Microsoft Exchange Replication service on the target server writes the database copy to a temporary directory located in the main database directory called *temp-seeding*.
15. The streaming backup operation on the source server ends when the end of the database is reached.
16. The write operation on the target server completes, and the database is moved from the temp-seeding directory to the final location. The temp-seeding directory is deleted.
17. On the target server, the Microsoft Exchange Replication service proxies a request to the Microsoft Exchange Search service to mount the content index catalog for the database copy, if

it exists. If there are existing out-of-date catalog files from a previous instance of the database copy, the mount operation fails, which triggers the need to replicate the catalog from the source server. Likewise, if the catalog doesn't exist on a new instance of the database copy on the target server, a copy of the catalog is required. The Microsoft Exchange Replication service directs the Microsoft Exchange Search service to suspend indexing for the database copy while a new catalog is copied from the source.

18. The Microsoft Exchange Replication service on the target server sends a seed catalog request to the Microsoft Exchange Replication service on the source server.
19. On the source server, the Microsoft Exchange Replication service requests the directory information from the Microsoft Exchange Search service and requests that indexing be suspended.
20. The Microsoft Exchange Search service on the source server returns the search catalog directory information to the Microsoft Exchange Replication service.
21. The Microsoft Exchange Replication service on the source server reads the catalog files from the directory.
22. The Microsoft Exchange Replication service on the source server moves the catalog data to the Microsoft Exchange Replication service on the target server using a connection across the replication network. After the read is complete, the Microsoft Exchange Replication service sends a request to the Microsoft Exchange Search service to resume indexing of the source database.
23. If there are any existing catalog files on the target server in the directory, the Microsoft Exchange Replication service on the target server deletes them.
24. The Microsoft Exchange Replication service on the target server writes the catalog data to a temporary directory called *CiSeed.Temp* until the data is completely transferred.
25. The Microsoft Exchange Replication service moves the complete catalog data to the final location.
26. The Microsoft Exchange Replication service on the target server resumes search indexing on the target database.
27. The Microsoft Exchange Replication service on the target server returns a completion status.
28. The final result of the operation is passed to the administrative interface from which the cmdlet was called.

Configuring database copies

After a database copy is created, you can view and modify its configuration settings when needed. You can view some configuration information by examining the **Properties** page for a database copy in the EAC. You can also use the Get-MailboxDatabase and Set-MailboxDatabaseCopy cmdlets in the Shell to view and configure database copy settings, such as replay lag time, truncation lag time, and activation preference order. For detailed steps about how to view and configure database copy settings, see Configure mailbox database copy properties.

Using replay lag and truncation lag options

Mailbox database copies support the use of a *replay lag time* and a *truncation lag time*, both of which are configured in minutes. Setting a replay lag time enables you to take a database copy back to a specific point in time. Setting a truncation lag time enables you to use the logs on a passive database copy to recover from the loss of log files on the active database copy. Because both of these features result in the temporary buildup of log files, using either of them will affect your storage design.

Replay lag time

Replay lag time is a property of a mailbox database copy that specifies the amount of time, in minutes, to delay log replay for the database copy. The replay lag timer starts when a log file has been replicated to the passive copy and has successfully passed inspection. By delaying the replay of logs to the database copy, you have the capability to recover the database to a specific point in time in the past. A mailbox database copy configured with a replay lag time greater than 0 is referred to as a *lagged mailbox database copy*, or simply, a *lagged copy*.

A strategy that uses database copies and the litigation hold features in Exchange 2013 can provide protection against a range of failures that would ordinarily cause data loss. However, these features can't provide protection against data loss in the event of logical corruption, which although rare, can cause data loss. Lagged copies are designed to prevent loss of data in the case of logical corruption. Generally, there are two types of logical corruption:

- **Database logical corruption** The database pages checksum matches, but the data on the pages is wrong logically. This can occur when ESE attempts to write a database page and even though the operating system returns a success message, the data is either never written to the disk or it's written to the wrong place. This is referred to as a *lost flush*. To prevent lost flushes from losing data, ESE includes a lost flush detection mechanism in the database along with a page patching feature (single page restore).
- **Store logical corruption** Data is added, deleted, or manipulated in a way that the user doesn't expect. These cases are generally caused by third-party applications. It's generally only corruption in the sense that the user views it as corruption. The Exchange store considers the transaction that produced the logical corruption to be a series of valid MAPI operations. The litigation hold feature in Exchange 2013 provides protection from store logical corruption (because it prevents content from being permanently deleted by a user or application). However, there may be scenarios where a user mailbox becomes so corrupted that it would be easier to restore the database to a point in time prior to the corruption, and then export the user mailbox to retrieve uncorrupted data.

The combination of database copies, hold policy, and ESE single page restore leaves only the rare but catastrophic store logical corruption case. Your decision on whether to use a database copy with a replay lag (a lagged copy) will depend on which third-party applications you use and your organization's history with store logical corruption.

If you choose to use lagged copies, be aware of the following implications for their use:

- The replay lag time is an administrator-configured value, and by default, it's disabled.
- The replay lag time setting has a default setting of 0 days, and a maximum setting of 14 days.

- Lagged copies aren't considered highly available copies. Instead, they are designed for disaster recovery purposes, to protect against store logical corruption.
- The greater the replay lag time set, the longer the database recovery process. Depending on the number of log files that need to replayed during recovery, and the speed at which your hardware can replay them, it may take several hours or more to recover a database.
- We recommend that you determine whether lagged copies are critical for your overall disaster recovery strategy. If using them is critical to your strategy, we recommend using multiple lagged copies, or using a redundant array of independent disks (RAID) to protect a single lagged copy, if you don't have multiple lagged copies. If you lose a disk or if corruption occurs, you don't lose your lagged point in time.
- Lagged copies aren't patchable with the ESE single page restore feature. If a lagged copy encounters database page corruption (for example, a -1018 error), it will have to be reseeded (which will lose the lagged aspect of the copy).

Activating and recovering a lagged mailbox database copy is an easy process if you want the database to replay all log files and make the database copy current. If you want to replay log files up to a specific point in time, it's a more difficult operation because you manually manipulate log files and run Exchange Server Database Utilities (Eseutil.exe).

For detailed steps about how to activate a lagged mailbox database copy, see Activate a lagged mailbox database copy.

Truncation lag time

Truncation lag time is a property of a mailbox database copy that specifies the amount of time, in minutes, to delay log deletion for the database copy after the log file has been replayed into the database copy. The truncation lag timer starts when a log file has been replicated to the passive copy, successfully passed inspection, and has been successfully replayed into the copy of the database. By delaying the truncation of log files from the database copy, you have the capability to recover from failures that affect the log files for the active copy of the database.

Database copies and log truncation

Log truncation works the same in Exchange 2013 as it did in Exchange 2010. Truncation behavior is determined by the replay lag time and truncation lag time settings for the copy.

The following criteria must be met for a database copy's log file to be truncated when lag settings are left at their default values of 0 (disabled):

- The log file must have been successfully backed up, or circular logging must be enabled.
- The log file must be below the checkpoint (the minimum log file required for recovery) for the database.
- All other lagged copies must have inspected the log file.
- All other copies (not lagged copies) must have replayed the log file.

The following criteria must be met for truncation to occur for a lagged database copy:

- The log file must be below the checkpoint for the database.

- The log file must be older than ReplayLagTime + TruncationLagTime.
- The log file must have been truncated on the active copy.

In Exchange 2013 log truncation doesn't occur on an active mailbox database copy when one or more passive copies are suspended. If planned maintenance activities are going to take an extended period of time (for example, several days), you may have considerable log file buildup. To prevent the log drive from filling up with transaction logs, you can remove the affected passive database copy instead of suspending it. When the planned maintenance is completed, you can re-add the passive database copy.

Exchange 2013 Service Pack 1 (SP1) introduces a new feature called *loose truncation*, which is disabled by default. During normal operations, each database copy keeps logs that need to be shipped to other database copies until all copies of a database confirm they have replayed (passive copies) or received (lagged copies) the log files. This is default log truncation behavior. If a database copy goes offline for some reason, the log files begin accumulating on the disks used by the other copies of the database. If the affected database copy remains offline for an extended period, this can cause the other database copies to run out of disk space.

When loose truncation is enabled, the truncation behavior is different. Each database copy tracks its own free disk space and applies loose truncation behavior if free space gets low. For the active copy, the oldest straggler (the passive database copy that is farthest behind in log replay) is ignored and truncation respects the oldest remaining passive copies. The active database copy is where global truncation is calculated. The passive copies will attempt to respect the truncation decision made on the active copy. Despite the implication of the name MinCopiesToProtect, Exchange will only ignore the oldest known straggler at the time truncation is run. For a passive copy, if space gets low, it will independently truncate its log files using the configured parameters described below.

When the offline database is brought back online, it will be missing log files that have been deleted from the other healthy copies, and its database copy status will be FailedAndSuspended. In this event, if Autoreseed is configured, the affected copy will be automatically reseeded. If Autoreseed is not configured, the database copy will need to be manually seeded by an administrator.

The required number of healthy copies, the free disk space threshold, and the number of logs to keep are all configurable parameters. By default, the free disk space threshold is 204800 MB (200 GB), and the number of logs to keep is 100,000 (100 GB) for passive copies, and 10,000 (10 GB) for active copies.

Enabling loose truncation and configuring loose truncation parameters is performed by editing the Windows registry on each DAG member. There are three registry values that can be configured, that are all stored under HKLM\Software\Microsoft\ExchangeServer\v15\BackupInformation. The BackupInformation key the following DWORD values do not exist by default and must be manually created. The DWORD registry values under BackupInformation are described in the following table:

Registry Value	Description	Default Value
LooseTruncation_MinCopiesTo	This key is used to enable loose	0

Protect	truncation. It represents the number of passive copies to protect from loose truncation on the active copy of a database. Setting the value of this key to 0 disables loose truncation.	
LooseTruncation_MinDiskFreeSpaceThresholdInMB	Available disk space (in MB) threshold for triggering loose truncation. If free disk space falls below this value, loose truncation is triggered.	If this registry value is not configured, the default value used by loose truncation is 200 GB.
LooseTruncation_MinLogsToProtect	The minimum number of log files to retain on healthy copies whose logs are being truncated. If this registry value is configured, then the configured value applies to both active and passive copies.	If this registry value is not configured, then default values of 100,000 for passive database copies and 10,000 for active database copies is used.

When using the LooseTruncation_MinLogsToProtect registry value, note that the behavior is different for active and passive database copies. On the active database copy, this is the number of extra logs that are retained preceding those that are required by the protected passive copies and the required range of the active copy.On a passive database copy, this is the number of logs maintained from the latest available log. One tenth of this number is also used to maintain logs prior to the required range of this passive copy. The two limits are in place to ensure that lagged database copies don't take up too much space, since their required range is typically very large.

Database activation policy

There are scenarios in which you may want to create a mailbox database copy and prevent the system from automatically activating that copy in the event of a failure, for example:

- If you deploy one or more mailbox database copies to an alternate or standby datacenter.
- If you configure a lagged database copy for recovery purposes.
- If you are performing maintenance or an upgrade of a server.

In each of the preceding scenarios, you have database copies that you don't want the system to activate automatically. To prevent the system from automatically activating a mailbox database copy, you can configure the copy to be blocked (suspended) for activation. This allows the system to maintain the currency of the database through log shipping and replay, but prevents the system from automatically activating and using the copy. Copies blocked for activation must be manually activated by an administrator. You can configure the database activation policy for an entire server by using the Set-MailboxServer cmdlet or an individual database copy by using the Set-MailboxDatabaseCopy cmdlet to set the *DatabaseCopyAutoActivationPolicy* parameter to Blocked.

For more information about configuring database activation policy, see Configure activation policy for a mailbox database copy.

Effect of mailbox moves on continuous replication

On a very busy mailbox database with a high log generation rate, there is a greater chance for data loss if replication to the passive database copies can't keep up with log generation. One scenario that can introduce a high log generation rate is mailbox moves. Exchange 2013 includes a Data Guarantee API that's used by services such as the Microsoft Exchange Mailbox Replication service (MRS) to check the health of the database copy architecture based on the value of the *DataMoveReplicationConstraint* parameter that was set by the system or an administrator. Specifically, the Data Guarantee API can be used to:

- **Check replication health** Confirms that the prerequisite number of database copies is available.
- **Check replication flush** Confirms that the required log files have been replayed against the prerequisite number of database copies.

When executed, the API returns the following status information to the calling application:

- **Retry** Signifies that there are transient errors that prevent a condition from being checked against the database.
- **Satisfied** Signifies that the database meets the required conditions or the database isn't replicated.
- **NotSatisfied** Signifies that the database doesn't meet the required conditions. In addition, information is provided to the calling application as to why the **NotSatisfied** response was returned.

The value of the *DataMoveReplicationConstraint* parameter for the mailbox database determines how many database copies should be evaluated as part of the request. The *DataMoveReplicationConstraint* parameter has the following possible values:

- `None` When you create a mailbox database, this value is set by default. When this value is set, the Data Guarantee API conditions are ignored. This setting should be used only for mailbox databases that aren't replicated.
- `SecondCopy` This is the default value when you add the second copy of a mailbox database. When this value is set, at least one passive database copy must meet the Data Guarantee API conditions.

- `SecondDatacenter` When this value is set, at least one passive database copy in another Active Directory site must meet the Data Guarantee API conditions.
- `AllDatacenters` When this value is set, at least one passive database copy in each Active Directory site must meet the Data Guarantee API conditions.
- `AllCopies` When this value is set, all copies of the mailbox database must meet the Data Guarantee API conditions.

Check Replication Health

When the Data Guarantee API is executed to evaluate the health of the database copy infrastructure, several items are evaluated.

If the *DataMoveReplicationConstraint* parameter is set to...	Then, for a given database...	Conditions
`SecondCopy`	At least one passive database copy for a replicated database must meet the conditions in the next column.	The passive database copy must: • Be healthy. • Have a replay queue within 10 minutes of the replay lag time. • Have a copy queue length less than 10 logs. • Have an average copy queue length less than 10 logs. The average copy queue length is computed based on the number of times the application has queried the database status.
`SecondDatacenter`	At least one passive database copy in another Active Directory site must meet the conditions in the next column.	
`AllDatacenters`	The active copy must be mounted, and a passive copy in each Active Directory site must meet the conditions in the next column.	
`AllCopies`	The active copy must be mounted, and all passive database copies must meet the conditions in the next column.	

Check Replication Flush

The Data Guarantee API can also be used to validate that a prerequisite number of database copies have replayed the required transaction logs. This is verified by comparing the last log replayed timestamp with that of the calling service's commit timestamp (in most cases, this is the timestamp of the last log file that contains required data) plus an additional five seconds (to deal with system time clock skews or drift). If the replay timestamp is greater than the commit timestamp, the *DataMoveReplicationConstraint* parameter is satisfied. If the replay timestamp is less than the commit timestamp, the *DataMoveReplicationConstraint* isn't satisfied.

Before moving large numbers of mailboxes to or from replication databases within a DAG, we recommend that you configure the *DataMoveReplicationConstraint* parameter on each mailbox database according to the following:

If you're deploying...	Set DataMoveReplicationConstraint to...
Mailbox databases that don't have any database copies	`None`
A DAG within a single Active Directory site	`SecondCopy`
A DAG in multiple datacenters using a stretched Active Directory site	`SecondCopy`
A DAG that spans two Active Directory sites, and you will have highly available database copies in each site	`SecondDatacenter`
A DAG that spans two Active Directory sites, and you will have only lagged database copies in the second site	`SecondCopy` This is because the Data Guarantee API won't guarantee data being committed until the log file is replayed into the database copy, and due to the nature of the database copy being lagged, this constraint will fail the move request, unless the lagged database copy ReplayLagTime value is less than 30 minutes.
A DAG that spans three or more Active Directory sites, and each site will contain highly available database copies	`AllDatacenters`

Balancing database copies

Due to the inherent nature of DAGs, as the result of database switchovers and failovers, active mailbox database copies will change hosts several times throughout a DAG's lifetime. As a result, DAGs can become unbalanced in terms of active mailbox database copy distribution. The following table shows an example of a DAG that has four databases with four copies of each database (for a total of 16 databases on each server) with an uneven distribution of active database copies.

DAG with unbalanced active copy distribution

Server	Number of active databases	Number of passive databases	Number of mounted databases	Number of dismounted databases	Preference count list
EX1	5	11	5	0	4, 4, 3, 5
EX2	1	15	1	0	1, 8, 6, 1
EX3	12	4	12	0	13, 2, 1, 0
EX4	1	15	1	0	1, 1, 5, 9

In the preceding example, there are four copies of each database, and therefore, only four possible values for activation preference (1, 2, 3, or 4). The **Preference count list** column shows the count of the number of databases with each of these values. For example, on EX3, there are 13 database copies with an activation preference of 1, two copies with an activation preference of 2, one copy with an activation preference of 3, and no copies with an activation preference of 4.

As you can see, this DAG isn't balanced in terms of the number of active databases hosted by each DAG member, the number of passive databases hosted by each DAG member, or the activation preference count of the hosted databases.

You can use the RedistributeActiveDatabases.ps1 script to balance the active mailbox databases copies across a DAG. This script moves databases between their copies in an attempt to have an equal number of mounted databases on each server in DAG. If required, the script also attempts to balance active databases across sites.

The script provides two options for balancing active database copies within a DAG:

- **BalanceDbsByActivationPreference** When this option is specified, the script attempts to move databases to their most preferred copy (based on activation preference) without regard to the Active Directory site.
- **BalanceDbsBySiteAndActivationPreference** When this option is specified, the script attempts to move active databases to their most preferred copy, while also trying to balance active databases within each Active Directory site.

After running the script with the first option, the preceding unbalanced DAG becomes balanced, as shown in the following table.

DAG with balanced active copy distribution

Server	Number of active databases	Number of passive databases	Number of mounted databases	Number of dismounted databases	Preference count list
EX1	4	12	4	0	4, 4, 4, 4
EX2	4	12	4	0	4, 4, 4, 4
EX3	4	12	4	0	4, 4, 4, 4
EX4	4	12	4	0	4, 4, 4, 4

As shown in the preceding table, this DAG is now balanced in terms of number of active and passive databases on each server and activation preference across the servers.

The following table lists the available parameters for the RedistributeActiveDatabases.ps1 script.

RedistributeActiveDatabases.ps1 script parameters

Parameter	Description
DagName	Specifies the name of the DAG you want to rebalance. If this parameter is omitted, the DAG of which the local server is a member is used.
BalanceDbsByActivationPreference	Specifies that the script should move databases to their most preferred copy without regard to the Active Directory site.
BalanceDbsBySiteAndActivationPreference	Specifies that the script should attempt to move active databases to their most preferred copy, while also trying to balance active databases within each Active Directory site.
ShowFinalDatabaseDistribution	Specifies that a report of current database distribution be displayed after redistribution is complete.
AllowedDeviationFromMeanPercentage	Specifies the allowed variation of active databases across sites, expressed as a percentage. The default is 20%. For example, if there were 99 databases distributed between

	three sites, the ideal distribution would be 33 databases in each site. If the allowed deviation is 20%, the script attempts to balance the databases so that each site has no more than 10% more or less than this number. 10% of 33 is 3.3, which is rounded up to 4. Therefore, the script attempts to have between 29 and 37 databases in each site.
ShowDatabaseCurrentActives	Specifies that the script produce a report for each database detailing how the database was moved and whether it's now active on its most-preferred copy.
ShowDatabaseDistributionByServer	Specifies that the script produce a report for each server showing its database distribution.
RunOnlyOnPAM	Specifies that the script run only on the DAG member that currently has the PAM role. The script verifies it's being run from the PAM. If it isn't being run from the PAM, the script exits.
LogEvents	Specifies that the script logs an event (MsExchangeRepl event 4115) containing a summary of the actions.
IncludeNonReplicatedDatabases	Specifies that the script should include non-replicated databases (databases without copies) when determining how to redistribute the active databases. Although non-replicated databases can't be moved, they may affect the distribution of the replicated databases.
Confirm	The Confirm switch can be used to suppress the confirmation prompt that appears by default when this script is run. To suppress the

	confirmation prompt, use the syntax -Confirm:$False. You must include a colon (:) in the syntax.

RedistributeActiveDatabases.ps1 examples

This example shows the current database distribution for a DAG, including preference count list.

```
RedistributeActiveDatabases.ps1 -DagName DAG1 -
ShowDatabaseDistributionByServer | Format-Table
```

This example redistributes and balances the active mailbox database copies in a DAG using activation preference without prompting for input.

```
RedistributeActiveDatabases.ps1 -DagName DAG1 -
BalanceDbsByActivationPreference -Confirm:$False
```

This example redistributes and balances the active mailbox database copies in a DAG using activation preference, and produces a summary of the distribution.

```
RedistributeActiveDatabases.ps1 -DagName DAG1 -
BalanceDbsByActivationPreference -
ShowFinalDatabaseDistribution
```

Monitoring database copies

A database copy is your first defense if a failure occurs that affects the active copy of a database. It's therefore critical to monitor the health and status of database copies to ensure that they will be available when needed. You can view a variety of information, including copy queue length, replay queue length, status, and content index state information, by examining the details of a database copy in the EAC. You can also use the **Get-MailboxDatabaseCopyStatus** cmdlet in the Shell to view a variety of status information for a database copy.

For more information about monitoring database copies, see Monitoring database availability groups.

Removing a database copy

A database copy can be removed at any time by using the EAC or by using the **Remove-MailboxDatabaseCopy** cmdlet in the Shell. After removing a database copy, you must manually delete any database and transaction log files from the server from which the database copy is being removed. For detailed steps about how to remove a database copy, see Remove a mailbox

database copy.

Database switchovers

The Mailbox server that hosts the active copy of a database is referred to as the mailbox database master. The process of activating a passive database copy changes the mailbox database master for the database and turns the passive copy into the new active copy. This process is called a database switchover. In a database switchover, the active copy of a database is dismounted on one Mailbox server and a passive copy of that database is mounted as the new active mailbox database on another Mailbox server. When performing a switchover, you can optionally override the database mount dial setting on the new mailbox database master.

You can quickly identify which Mailbox server is the current mailbox database master by reviewing the right-hand column under the **Database Copies** tab in the EAC. You can perform a switchover by using the **Activate** link in the EAC, or by using the **Move-ActiveMailboxDatabase** cmdlet in the Shell.

There are several internal checks that will be performed before activating a passive copy:

- The status of the database copy is checked. If the database copy is in a failed state, the switchover is blocked. You can override this behavior and bypass the health check by using the *SkipHealthChecks* parameter of the **Move-ActiveMailboxDatabase** cmdlet. This parameter allows you to move the active copy to a database copy in a failed state.
- The active database copy is checked to see if it's currently a seeding source for any passive copies of the database. If the active copy is currently being used as a source for seeding, the switchover is blocked. You can override this behavior and bypass the seeding source check by using the *SkipActiveCopyChecks* parameter of the **Move-ActiveMailboxDatabase** cmdlet. This parameter allows you to move an active copy that's being used as a seeding source. Using this parameter will cause the seeding operation to be cancelled and considered failed.
- The copy queue and replay queue lengths for the database copy are checked to ensure their values are within the configured criteria. Also, the database copy is verified to ensure that it isn't currently in use as a source for seeding. If the values for the queue lengths are outside the configured criteria, or if the database is currently used as a source for seeding, the switchover is blocked. You can override this behavior and bypass these checks by using the *SkipLagChecks* parameter of the **Move-ActiveMailboxDatabase** cmdlet. This parameter allows a copy to be activated that has replay and copy queues outside of the configured criteria.
- The state of the search catalog (content index) for the database copy is checked. If the search catalog isn't up to date, is in an unhealthy state, or is corrupt, the switchover is blocked. You can override this behavior and bypass the search catalog check by using the *SkipClientExperienceChecks* parameter of the **Move-ActiveMailboxDatabase** cmdlet. This parameter causes this search to skip the catalog health check. If the search catalog for the database copy you're activating is in an unhealthy or unusable state and you use this parameter to skip the catalog health check and activate the database copy, you will need to either crawl or seed the search catalog again.

When performing a database switchover, you also have the option of overriding the mount dial settings configured for the server that hosts the passive database copy being activated. Using the *MountDialOverride* parameter of the **Move-ActiveMailboxDatabase** cmdlet instructs the target server to override its own mount dial settings and use those specified by the *MountDialOverride* parameter.

For detailed steps about how to perform a switchover of a database copy, see Activate a mailbox database copy.

Add a mailbox database copy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-30*

When you add a copy of a mailbox database, continuous replication is automatically enabled between the existing database and the database copy. Database copies are automatically assigned an identity in the format of *<DatabaseName>**<HostMailboxServerName>*. For example, a copy of the database DB1 that's hosted on the server MBX3 would be DB1\MBX3.

Looking for other management tasks related to mailbox database copies? Check out Managing mailbox database copies.

What do you need to know before you begin?

- Estimated time to complete this task: 2 minutes, plus the time to seed the database copy, which depends on a variety of factors, such as the size of the database, the speed, available bandwidth and latency of the network, and storage speeds.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox database copies" entry in the High availability and site resilience permissions topic.
- The active copy of the database must be mounted.
- The specified Mailbox server must not already host a copy of the database.
- The path for the database copy and its log files must be available on the selected Mailbox server.
- The server hosting the active copy and the server that will host the passive copy must be in the same database availability group (DAG). The DAG must also have quorum and be healthy.
- If you're adding the second copy of a database (for example, creating the first passive copy of the database), circular logging must not be enabled for the specified mailbox database. If circular logging is enabled, you must first disable it. After the mailbox database copy has been added, circular logging can be enabled. After circular logging is enabled for a replicated mailbox

database, continuous replication circular logging (CRCL) is used instead of JET circular logging. If you're adding the third or subsequent copy of a database, CRCL can remain enabled.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to add a mailbox database copy

1. In the EAC, go to **Servers** > **Databases**.
2. Select the database that you want to copy, and then click .
3. On the **add mailbox database copy** page, click **browse...**, select the Mailbox server that will host the database copy, and then click **OK**.
4. Optionally, configure the **Activation preference number** for the database copy.
5. Click **More options...** to designate the database copy as a lagged database copy by configuring a replay lag time, or to postpone automatic seeding of the database copy.
6. Click **Save** to save the configuration changes and add the mailbox database copy.
7. Click **OK** to acknowledge any messages that appear.

Use the Shell to add a mailbox database copy

This example adds a copy of mailbox database DB1 to the Mailbox server MBX3. Replay lag time and truncation lag time are left at the default values of zero, and the activation preference is configured with a value of 2.

```
Add-MailboxDatabaseCopy -Identity DB1 -MailboxServer MBX3 -ActivationPreference 2
```

This example adds a copy of mailbox database DB2 to the Mailbox server MBX4. Replay lag time and truncation lag time are left at the default values of zero, and the activation preference is configured with a value of 5. In addition, seeding is being postponed for this copy so that it can be seeded using a local source server instead of the current active database copy, which is geographically distant from MBX4.

```
Add-MailboxDatabaseCopy -Identity DB2 -MailboxServer MBX4 -ActivationPreference 5 -SeedingPostponed
```

This example adds a copy of mailbox database DB3 to the Mailbox server MBX5. Replay lag time is set to 3 days, truncation lag time is left at the default value of zero, and the activation preference is configured with a value of 4.

```
Add-MailboxDatabaseCopy -Identity DB3 -MailboxServer MBX5 -
ReplayLagTime 3.00:00:00 -ActivationPreference 4
```

How do you know this worked?

To verify that you have successfully created a mailbox database copy, do one of the following:

- In the EAC, navigate to **Servers** > **Databases**. Select the database that was copied. In the Details pane, the status of the database copy and its content index are displayed, along with the current copy queue length.
- In the Shell, run the following command to verify the mailbox database copy was created and is healthy.

```
Get-MailboxDatabaseCopyStatus <DatabaseCopyName>
```

The Status and Content Index State should both be Healthy.

For more information

Mailbox database copies

Managing mailbox database copies

Configure mailbox database copy properties

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-01*

Each mailbox database copy has its own properties that you can configure. These include the amount of time, if any, for replay lag and truncation lag, and the activation preference number. For more information about replay lag, truncation lag and the activation preference number, see Managing mailbox database copies.

What do you need to know before you begin?

- Estimated time to complete this task: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To

see what permissions you need, see the "Mailbox database copies" entry in the High availability and site resilience permissions topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure mailbox database copy properties

1. In the EAC, go to **Servers** > **Databases**.
2. Select the database you want to configure.
3. In the Details pane, under **Database Copies**, click **View details** for the desired database copy, and then view or configure the following:
   - **Database** Displays the name of the selected database.
   - **Mailbox server** Displays the name of the Mailbox server that hosts the selected database copy.
   - **Content index state** Displays the current state of the content index for the selected database copy.
   - **Status** Displays the current status of the selected database copy.
   - **Copy queue length** Indicates the number of log files waiting to be copied to the selected database copy. This field is relevant only for passive database copies.
   - **Replay queue length** Indicates the number of log files waiting to be replayed into the selected database copy. This field is relevant only for passive database copies.
   - **Error messages** Displays any error messages for database copies that have a status of `Failed` or `Failed and Suspended`.
   - **Latest available log time** Displays the date and time stamp of the most recently generated log file on the active copy of the database. This field is relevant only for passive database copies. On active database copies (replicated and stand-alone), this field will display **never**.
   - **Last inspected log time** Displays the date and time stamp of the last log file that was inspected by the LogInspector on the selected database copy. This field is relevant only for passive database copies. On active database copies (replicated and stand-alone), this field will display **never**.
   - **Last copied log time** Displays the date and time stamp of the last log file that was copied by the LogCopier on the selected database copy. This field is relevant only for passive database copies. On active database copies (replicated and stand-alone), this field will display **never**.
   - **Last replayed log time** Displays the date and time stamp of the last log file that was replayed by the LogReplayer into the selected database copy. This field is relevant only for

passive database copies. On active database copies (replicated and stand-alone), this field will display **never**.

- **Activation preference number** Displays the activation preference number. This is used as part of Active Manager's best copy selection process and to balance the DAG by redistributing active mailbox databases throughout the DAG using the RedistributeActiveDatabases.ps1 script. The value for activation preference is a number equal to or greater than 1, where 1 is at the top of the preference order. The number can't be larger than the number of copies of the mailbox database.
- **Replay lag time (days)** Displays the amount of time that the Microsoft Exchange Information Store service should wait before replaying log files that have been copied by the Microsoft Exchange Replication service to the passive database copy. Setting this parameter to a value greater than 0 creates a lagged database copy. The default setting for this value is 0 days. The maximum allowable value for this setting is 14 days. The minimum allowable value is 0 days, and setting this value to 0 disables replay lag.

Use the Shell to configure mailbox database copy properties

This example configures a mailbox database copy with an activation preference number of 3.

```
Set-MailboxDatabaseCopy -Identity DB3\EX3 -
ActivationPreference 3
```

This example configures a copy of the database DB1 that's hosted on Server1 with a replay lag time and truncation lag time of 1 day, and an activation preference number of 2.

```
Set-MailboxDatabaseCopy -Identity DB1\Server1 -
ReplayLagTime 1.0:0:0 -TruncationLagTime 1.0:0:0 -
ActivationPreference 2
```

How do you know this worked?

To verify that you've successfully configured a mailbox database copy, do one of the following:

- In the EAC, navigate to **Servers** > **Databases**. Select the appropriate database, and in the Details pane, click **View details** to view the database copy properties.
- In the Shell, run the following command to display configuration information for a database copy.

```
Get-MailboxDatabaseCopyStatus <DatabaseCopyName> | Format-
List
```

For more information

Set-MailboxDatabaseCopy

Get-MailboxDatabaseCopyStatus

Get-MailboxDatabase

Move the mailbox database path for a mailbox database copy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-07*

After a mailbox database is created, you can move it to another volume, folder, location, or path by using either the EAC or the Shell. For step-by-step instructions about how to move the database path for a non-replicated mailbox database, see Move a mailbox database path.

If the mailbox database being moved is replicated to one or more mailbox database copies, you must follow the procedure in this topic to move the mailbox database path. All copies of a mailbox database must be located in the same path on each server that hosts a copy. For example, if database DB1 is located at C:\mountpoints\DB1 on server EX1, copies of DB1 on servers EX2, EX3, and so on, must also be located at C:\mountpoints\DB1.

Looking for other management tasks related to mailbox database copies? Check out Managing mailbox database copies.

What do you need to know before you begin?

- Estimated time to complete this task: 2 minutes, plus the time to move the data, which depends on a variety of factors, such as the size of the database, the speed, available bandwidth and latency of the network, and storage speeds.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox database copies" entry in the High availability and site resilience permissions topic.
- To perform the move operation, the database must be temporarily dismounted, making it inaccessible to all users. If the database is currently dismounted, it isn't remounted upon completion.
- To perform the move operation, replication for the database must be disabled for all copies. It's

not enough to suspend replication; you must disable it by using the Remove-MailboxDatabaseCopy cmdlet to remove the database copies.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to move a replicated mailbox database to a new path

Note:

You can't use the EAC to move a replicated mailbox database to a new path.

1. Note any replay lag or truncation lag settings for all copies of the mailbox database being moved. You can obtain this information by using the Get-MailboxDatabase cmdlet, as shown in this example.

```
Get-MailboxDatabase DB1 | Format-List *lag*
```

2. If circular logging is enabled for the database, it must be disabled before proceeding. You can disable circular logging for a mailbox database by using the Set-MailboxDatabase cmdlet, as shown in this example.

```
Set-MailboxDatabase DB1 -CircularLoggingEnabled $false
```

3. Remove all mailbox database copies for the database being moved. For detailed steps, see Remove a mailbox database copy. After all copies are removed, preserve the database and transaction log files from each server from which the database copy is being removed by moving them to another location. These files are being preserved so the database copies don't require re-seeding after they have been re-added.
4. Move the mailbox database path to the new location. For detailed steps, see Move a mailbox database path.

Important:

During the move operation, the database being moved must be dismounted. Until the move is complete, this process will cause an interruption in service and an outage for all users with mailboxes on the database being moved. After the move operation completes, the database is automatically mounted.

5. Create the necessary folder structure on each Mailbox server that previously contained a passive copy of the moved mailbox database. For example, if you moved the database to C:\mountpoints\DB1, you must create this same path on each Mailbox server that will host a mailbox database copy.
6. After creating the folder structure, move the passive copy of the mailbox database and its log

stream to the new location. These are the files that were left from and preserved after Step 3. Repeat this process for each database copy that was removed in Step 3.

7. Add all of the database copies that were removed in Step 3. For detailed steps, see Add a mailbox database copy.
8. On each server that contains a copy of the mailbox database being moved, run the following commands to stop and restart the content index services.

```
Net stop MSExchangeFastSearch
Net start MSExchangeFastSearch
```

9. Optionally, enable circular logging by using the Set-MailboxDatabase cmdlet, as shown in this example.

```
Set-MailboxDatabase DB1 -CircularLoggingEnabled $true
```

10. Reconfigure any previously set values for replay lag time and truncation lag time by using the Set-MailboxDatabaseCopy cmdlet, as shown in this example.

```
Set-MailboxDatabaseCopy DB1\MBX2 -ReplayLagTime 00:15:00
```

11. As each copy is added, we recommend that you verify the health and status of the copy prior to adding the next copy. You can verify the health and status by:
    a. Examining the event log for any error or warning events related to the database or the database copy.
    b. Using the Get-MailboxDatabaseCopyStatus cmdlet to check the health and status of continuous replication for the database copy.
    c. Using the Test-ReplicationHealth cmdlet to verify the health and status of the database availability group and continuous replication.

For detailed syntax and parameter information, see the following topics:

- Get-MailboxDatabase
- Set-MailboxDatabase
- Set-MailboxDatabaseCopy
- Get-MailboxDatabaseCopyStatus
- Test-ReplicationHealth

How do you know this worked?

To verify that you've successfully moved the path for a mailbox database copy, do one of the following:

- In the EAC, navigate to **Servers** > **Databases**. Select the database that was copied. In the Details pane, the status of the database copy and its content index are displayed, along with the current copy queue length. Verify that the status is Healthy.
- In the Shell, run the following command to verify the mailbox database copy was created and is healthy.

```
Get-MailboxDatabaseCopyStatus <DatabaseCopyName>
```

The Status and Content Index State should both be Healthy.

Configure activation policy for a mailbox database copy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-02*

Activation is the process of changing a mailbox database copy from a passive copy to an active copy. Activation occurs automatically by the system as part of a database or server failover operation, and it can be performed manually by an administrator as part of a database or server switchover operation. Blocking a database for activation prevents it from becoming the active copy during a database or server failover.

Looking for other management tasks related to mailbox database copies? Check out Managing mailbox database copies.

What do you need to know before you begin?

- Estimated time to complete this task: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox database copies" entry in the High availability and site resilience permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to configure the activation policy for a mailbox database copy

1. In the EAC, go to **Servers** > **Databases**.
2. Select the database that you want to configure.
3. In the Details pane, under **Database Copies**, locate the database copy you want to configure and click **Suspend**.
4. Optionally, add a comment, and select the check box that says **This copy can only be activated by manual intervention**.
5. Click **Save** to save the configuration changes for the mailbox database copy.

Use the Shell to suspend or resume a database copy for activation

This example blocks the copy of the database DB1 on the server MBX2 for activation.

```
Suspend-MailboxDatabaseCopy -Identity DB1\MBX2 -
ActivationOnly
```

This example resumes the copy of the database DB1 on the server MBX2 for activation.

```
Resume-MailboxDatabaseCopy -Identity DB1\MBX2
```

For detailed syntax and parameter information, see Suspend-MailboxDatabaseCopy or Resume-MailboxDatabaseCopy.

Use the Shell to configure the activation policy for a server

This example configures the database copies on server MBX2 as blocked for activation.

```
Set-MailboxServer -Identity MBX2 -
DatabaseCopyAutoActivationPolicy Blocked
```

This example configures the database copies on server MBX3 as blocked for out-of-site activation.

```
Set-MailboxServer -Identity MBX3 -
DatabaseCopyAutoActivationPolicy IntrasiteOnly
```

This example configures the database copies on server MBX4 as unblocked for activation.

```
Set-MailboxServer -Identity MBX4 -
DatabaseCopyAutoActivationPolicy Unrestricted
```

For detailed syntax and parameter information, see Suspend-MailboxDatabaseCopy, Resume-MailboxDatabaseCopy, or Set-MailboxServer.

How do you know this worked?

To verify that you've successfully configured the activation policy, do one of the following:

- In the Shell, run the following command to verify activation settings for a database copy.

```
Get-MailboxDatabaseCopyStatus <DatabaseCopyName> | Format-List ActivationSuspended
```

- In the Shell, run the following command to verify activation settings for a DAG member.

```
Get-MailboxServer <ServerName> | Format-List DatabaseCopyAutoActivationPolicy
```

Update a mailbox database copy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-02*

Updating, also known as *seeding*, is the process in which a copy of a mailbox database is added to another Mailbox server in a database availability group (DAG). The newly added copy becomes the baseline database for the passive copy into which log files copied from the active copy are replayed. Seeding is required under the following conditions:

- When a new passive copy of a database is created. Seeding can be postponed for a new mailbox database copy, but eventually, each passive database copy must be seeded to function as a redundant database copy.
- After a failover occurs in which data is lost as a result of the passive database copy having become diverged and unrecoverable.
- When the system has detected a corrupted log file that can't be replayed into the passive copy of the database.
- After an offline defragmentation of any copy of the database occurs.
- After the log generation sequence for the database has been reset back to 1.

You can perform seeding by using the following methods:

- **Automatic seeding** An automatic seed produces a passive copy of the active database on the target Mailbox server. Automatic seeding occurs during the creation of a database.
- **Seeding using the Update-MailboxDatabaseCopy cmdlet** You can use the Update-MailboxDatabaseCopy cmdlet in the Shell to seed a database copy at any time.
- **Seeding using the Update Mailbox Database Copy wizard** You can use the Update Mailbox

Database Copy wizard in the EAC to seed a database copy at any time.

- **Manually copying the offline database** You can dismount the active copy of the database and copy the database file to the same location on another Mailbox server in the same DAG. If you use this method, there will be an interruption in service because the process requires you to dismount the database.

Updating a database copy can take a long time, especially if the database being copied is large, or if there is high network latency or low network bandwidth. After the seeding process has started, don't close the EAC or the Shell until the process has completed. If you do, the seeding operation will be terminated.

A database copy can be seeded using either the active copy or an up-to-date passive copy as the source for the seed. When seeding from a passive copy, be aware that the seed operation will terminate with a network communication error under the following circumstances:

- If the status of the seeding source copy changes to Failed or FailedAndSuspended.
- If the database fails over to another copy.

Multiple database copies can be seeded simultaneously. However, when seeding multiple copies simultaneously, you must seed only the database file, and omit the content index catalog. You can do this by using the *DatabaseOnly* parameter with the Update-MailboxDatabaseCopy cmdlet.

Note:

If you don't use the *DatabaseOnly* parameter when seeding multiple targets from the same source, the task will fail with SeedInProgressException error FE1C6491.

Looking for other management tasks related to mailbox database copies? Check out Managing mailbox database copies.

What do you need to know before you begin?

- Estimated time to complete this task: 2 minutes, plus the time to seed the database copy, which depends on a variety of factors, such as the size of the database, the speed, available bandwidth and latency of the network, and storage speeds.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox database copies" entry in the High availability and site resilience permissions topic.
- The mailbox database copy must be suspended. For detailed steps, see Suspend or resume a mailbox database copy.
- The Remote Registry service must be running on the server hosting the passive database copy you're updating.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to update a mailbox database copy

1. In the EAC, go to **Servers** > **Databases**.
2. Select the mailbox database whose passive copy you want to update.
3. In the Details pane, under **Database Copies**, click **Suspend** under the passive database copy you want to seed. Provide any optional comments, and click **save**.
4. In the Details pane, under **Database Copies**, click **Update** under the passive database copy you want to seed.
5. By default, the active copy of the database is used as the source database for seeding. If you prefer to use a passive copy of the database for seeding, click **browse...** to select the server containing the passive database copy you want to use for the source.
6. Click **save** to update the passive database copy.

Use the Shell to update a mailbox database copy

This example shows how to seed a copy of the database DB1 on MBX1.

```
Update-MailboxDatabaseCopy -Identity DB1\MBX1
```

This example shows how to seed a copy of the database DB1 on MBX1 using MBX2 as the source Mailbox server for the seed.

```
Update-MailboxDatabaseCopy -Identity DB1\MBX1 -SourceServer
MBX2
```

This example shows how to seed a copy of the database DB1 on MBX1 without seeding the content index catalog.

```
Update-MailboxDatabaseCopy -Identity DB1\MBX1 -DatabaseOnly
```

This example shows how to seed the content index catalog for the copy of the database DB1 on MBX1 without seeding the database file.

```
Update-MailboxDatabaseCopy -Identity DB1\MBX1 -CatalogOnly
```

Manually copy an offline database

1. If circular logging is enabled for the database, it must be disabled before proceeding. You can disable circular logging for a mailbox database by using the Set-MailboxDatabase cmdlet, as shown in this example.

```
Set-MailboxDatabase DB1 -CircularLoggingEnabled $false
```

2. Dismount the database. You can use the Dismount-Database cmdlet, as shown in this example.

```
Dismount-Database DB1 -Confirm $false
```

3. Manually copy the database files (the database file and all log files) to a second location, such as an external disk drive or a network share.
4. Mount the database. You can use the Mount-Database cmdlet, as shown in this example.

```
Mount-Database DB1
```

5. On the server that will host the copy, copy the database files from the external drive or network share to the same path as the active database copy. For example, if the active copy database path is D:\DB1\DB1.edb and log file path is D:\DB1, you would copy the database files to D:\DB1 on the server that will host the copy.
6. Add the mailbox database copy by using the Add-MailboxDatabaseCopy cmdlet with the *SeedingPostponed* parameter, as shown in this example.

```
Add-MailboxDatabaseCopy -Identity DB1 -MailboxServer MBX3 -
SeedingPostponed
```

7. If circular logging is enabled for the database, enable it again by using the Set-MailboxDatabase cmdlet, as shown in this example.

```
Set-MailboxDatabase DB1 -CircularLoggingEnabled $true
```

How do you know this worked?

To verify that you've successfully seeded a mailbox database copy, do one of the following:

- In the EAC, navigate to **Servers** > **Databases**. Select the database that was seeded. In the Details pane, the status of the database copy and its content index are displayed, along with the current copy queue length.
- In the Shell, run the following command to verify the mailbox database copy was seeded successfully and is healthy.

```
Get-MailboxDatabaseCopyStatus <DatabaseCopyName>
```

The Status and Content Index State should both be Healthy.

Suspend or resume a mailbox database copy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-02*

You may need to suspend or resume a database copy for a variety of reasons, such as maintenance on the disk that contains a database copy, or suspend an individual database copy from activation for disaster recovery purposes.

Looking for other management tasks related to mailbox database copies? Check out Managing mailbox database copies.

What do you need to know before you begin?

- Estimated time to complete this task: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox database copies" entry in the High availability and site resilience permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to suspend a mailbox database copy

1. In the EAC, go to **Servers** > **Databases**.
2. Select the database whose copy you want to suspend.
3. In the Details pane, under **Database Copies**, click **Suspend** under the database copy you want to suspend.
4. In the **Comments** field, add an optional comment of up to 512 characters specifying the reason for the suspension.
5. To suspend the database copy from automatic activation, select the **This copy can only be activated by manual intervention** check box.
6. Click **save** to suspend the database copy.

Use the EAC to resume a mailbox database copy

1. In the EAC, go to **Servers** > **Databases**.
2. Select the database whose copy you want to resume.

3. In the Details pane, under **Database Copies**, click **Resume** under the database copy you want to resume.
4. Click **yes** to resume the database copy.

Use the Shell to suspend a mailbox database copy

This example suspends continuous replication for a copy of the database DB1 hosted on the server MBX1. An optional comment has also been specified.

```
Suspend-MailboxDatabaseCopy -Identity DB1\MBX1 -
SuspendComment "Maintenance on MBX1" -Confirm:$False
```

This example suspends activation for a copy of the database DB2 hosted on the server MBX2.

```
Suspend-MailboxDatabaseCopy -Identity DB2\MBX2 -
ActivationOnly -Confirm:$False
```

Use the Shell to resume a mailbox database copy

This example resumes a copy of the database DB1 on the server MBX1.

```
Resume-MailboxDatabaseCopy -Identity DB1\MBX1
```

This example resumes a copy of the database DB2 on the server MBX2 for replication only.

```
Resume-MailboxDatabaseCopy -Identity DB2\MBX2 -
ReplicationOnly
```

How do you know this worked?

To verify that you have successfully suspended or resumed a mailbox database copy, do one of the following:

- In the EAC, navigate to **Servers** > **Databases**. Select the appropriate database, and in the Details pane, click **View details** to view the database copy properties.
- In the Shell, run the following command to display status information for a database copy.

```
Get-MailboxDatabaseCopyStatus <DatabaseCopyName> | Format-
List
```

Activate a mailbox database copy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-01*

Activating a mailbox database copy is the process of designating a specific passive copy as the new active copy of a mailbox database. This process is referred to as a *database switchover*. A database switchover involves dismounting the current active database and mounting the database copy on the specified server as the new active mailbox database copy. The database copy that will become the active mailbox database must be healthy and current.

Looking for other management tasks related to mailbox database copies? Check out Managing mailbox database copies.

What do you need to know before you begin?

- Estimated time to complete this task: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox database copies" entry in the High availability and site resilience permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to move the active mailbox database

1. In the EAC, go to **Servers** > **Databases**.
2. Select the database whose copy you want to activate.
3. In the Details pane, under **Database Copies**, click **Activate** under the database copy you want to activate.
4. Click **yes** to activate the database copy.

Use the Shell to move the active mailbox database

This example activates and mounts a copy of the database DB4 hosted on MBX3 as the new active mailbox database. This command makes DB4 the new active mailbox database, and it doesn't override the database mount dial settings on MBX3.

```
Move-ActiveMailboxDatabase DB4 -ActivateOnServer MBX3 -
```

```
MountDialOverride:None
```

This example performs a switchover of the database DB2 to the Mailbox server MBX1. When the command completes, MBX1 hosts the active copy of DB2. Because the *MountDialOverride* parameter is set to `None`, MBX1 mounts the database using its own defined database auto mount dial settings.

```
Move-ActiveMailboxDatabase DB2 -ActivateOnServer MBX1 -
MountDialOverride:None
```

This example performs a switchover of the database DB1 to the Mailbox server MBX3. When the command completes, MBX3 hosts the active copy of DB1. Because the *MountDialOverride* parameter is specified with a value of `Good Availability`, MBX3 mounts the database using a database auto mount dial setting of *GoodAvailability*.

```
Move-ActiveMailboxDatabase DB1 -ActivateOnServer MBX3 -
MountDialOverride:GoodAvailability
```

This example performs a switchover of the database DB3 to the Mailbox server MBX4. When the command completes, MBX4 hosts the active copy of DB3. Because the *MountDialOverride* parameter isn't specified, MBX4 mounts the database using a database auto mount dial setting of *Lossless*.

```
Move-ActiveMailboxDatabase DB3 -ActivateOnServer MBX4
```

This example performs a server switchover for the Mailbox server MBX1. All active mailbox database copies on MBX1 will be activated on one or more other Mailbox servers with healthy copies of the active databases on MBX1.

```
Move-ActiveMailboxDatabase -Server MBX1
```

This example performs a switchover of the database DB4 to the Mailbox server MBX5. In this example, the database copy on MBX5 has a replay queue greater than 6. As a result, the *SkipLagChecks* parameter must be specified to activate the database copy on MBX5.

```
Move-ActiveMailboxDatabase DB4 MBX5 -SkipLagChecks
```

This example performs a switchover of the database DB5 to the Mailbox server MBX6. In this example, the database copy on MBX6 has a *ContentIndexState* of Failed. As a result, the *SkipClientExperienceChecks* parameter must be specified to activate the database copy on MBX6.

```
Move-ActiveMailboxDatabase DB5 MBX6 -
SkipClientExperienceChecks
```

How do you know this worked?

To verify that you've successfully activated a mailbox database copy, do one of the following:

- In the EAC, navigate to **Servers** > **Databases**. Select the appropriate database, and in the Details pane, click **View details** to view the database copy properties.
- In the Shell, run the following command to display status information for a database copy.

```
Get-MailboxDatabaseCopyStatus <DatabaseCopyName> | Format-List
```

For more information

Mailbox database copies

Configure mailbox database copy properties

Activate a lagged mailbox database copy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-28*

A lagged mailbox database copy is a mailbox database copy configured with a replay lag time value greater than 0. Activating and recovering a lagged mailbox database copy is a simple process if you want the database to replay all log files and make the database copy current. If you want to replay log files up to a specific point in time, it's a more difficult operation because you have to manually manipulate log files and run Eseutil.

Looking for other information related to lagged mailbox database copies? Check out Managing mailbox database copies.

Note:

The amount of time it takes to activate a lagged mailbox database copy directly depends on how many log files need to be replayed and how fast the hardware can replay them. At a minimum, you should experience a log replay rate of at least two logs per second per database.

What do you need to know before you begin?

- Estimated time to complete this task: 1 minute, plus the time it takes to duplicate the lagged copy, replay the necessary log files, and extract the data or mount the database for client activity.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox database copies" entry in the High availability and site resilience permissions topic.
- The mailbox database copy being activated must be configured with a replay lag time greater than 0.
- The mailbox database copy being activated must have all log files to the point in time to which you want to recover it. Keep in mind that database transactions can span multiple log files when determining the point in time to which you want to recover.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the Shell to activate a lagged mailbox database copy to a specific point in time

Note:

You can't use the EAC to activate a lagged mailbox database copy to a specific point in time. Instead, you perform a series of steps using the Shell and the command line.

1. This example suspends replication for the lagged copy being activated by using the Suspend-MailboxDatabaseCopy cmdlet.

```
Suspend-MailboxDatabaseCopy DB1\EX3 -SuspendComment
"Activate lagged copy of DB1 on Server EX3" -Confirm:$false
```

2. Optionally (to preserve a lagged copy), make a copy of the database copy and its log files.

Note:

At this point, continuing to perform this procedure on the existing volume would incur a copy on write performance penalty. If this isn't desirable, you can copy the database and log files to another volume to perform the recovery.

3. Determine which log files are required to replay into the database to meet your point-in-time requirement for this recovery (based on log file date and time, as shown in Windows Explorer). All logs created after this point should be moved to a different directory, until the recovery process is completed, and the logs are no longer needed.
4. Delete the checkpoint (.chk) file for the database.
5. This example uses Eseutil to perform the recovery operation.

```
Eseutil.exe /r eXX /a
```

Note:
In the preceding example, e*XX* is the log generation prefix for the database (for example, E00, E01, E02, and so on).

Important:
This step may take a considerable amount of time, depending on several factors, such as the length of the replay lag time, the number of log files generated during that period, and the speed at which your hardware can replay those logs into the database being recovered.

6. After log replay is finished, the database is in a clean shutdown state and can be copied and used for recovery purposes.
7. After the recovery process is complete, this example resumes replication for the database that was used as part of the recovery process.

```
Resume-MailboxDatabaseCopy DB1\EX3
```

For detailed syntax and parameter information, see Suspend-MailboxDatabaseCopy or Resume-MailboxDatabaseCopy.

Use the Shell to activate a lagged mailbox database copy by replaying all uncommitted log files

1. Optionally (to preserve a lagged copy), make a copy of the database copy and its log files.
   a. This example suspends replication for the lagged copy being activated by using the Suspend-MailboxDatabaseCopy cmdlet.

```
Suspend-MailboxDatabaseCopy DB1\EX3 -SuspendComment
"Activate lagged copy of DB1 on Server EX3" -Confirm:$false
```

   b. Optionally (to preserve a lagged copy), make a copy of the database copy and its log files.

Note:
At this point, continuing to perform this procedure on the existing volume would incur a copy on write performance penalty. If this isn't desirable, you can copy the database and log files to another volume to perform the recovery.

2. This example activates the lagged mailbox database copy using the Move-ActiveMailboxDatabase cmdlet with the *SkipLagChecks* parameter.

```
Move-ActiveMailboxDatabase DB1 -ActivateOnServer EX3 -
SkipLagChecks
```

Use the Shell to activate a lagged mailbox database copy

by using SafetyNet recovery

1. Optionally (to preserve a lagged copy), take a file system-based (non-Exchange aware) Volume Shadow Copy Service (VSS) snapshot of the volumes containing the database copy and its log files.
   a. This example suspends replication for the lagged copy being activated by using the Suspend-MailboxDatabaseCopy cmdlet.

```
Suspend-MailboxDatabaseCopy DB1\EX3 -SuspendComment
"Activate lagged copy of DB1 on Server EX3" -Confirm:$false
```

   b. Optionally (to preserve a lagged copy), make a copy of the database copy and its log files.

Note:
At this point, continuing to perform this procedure on the existing volume would incur a copy-on-write performance penalty. If this isn't desirable, you can copy the database and log files to another volume to perform the recovery.

2. Determine the required logs for the lagged database copy by looking for the "Log Required:" value in ESEUTIL database header output

```
Eseutil /mh <DBPath> | findstr /c:"Log Required"
```

Take note of the hexadecimal numbers in parentheses. The first number is the lowest generation required (referred to as LowGeneration), and the second number is the highest generation required (referred to as HighGeneration). Move all log generation files that have a generation sequence greater than HighGeneration to a different location so that they are not replayed into the database.

3. On the server hosting the active copy of database, either delete the log files for the lagged copy being activated from the active copy, or stop the Microsoft Exchange Replication service.
4. Perform a database switchover and activate the lagged copy. This example activates the database by using the Move-ActiveMailboxDatabase cmdlet with several parameters.

```
Move-ActiveMailboxDatabase DB1 -ActivateOnServer EX3 -
MountDialOverride BestEffort -SkipActiveCopyChecks -
SkipClientExperienceChecks -SkipHealthChecks -SkipLagChecks
```

5. At this point, the database will automatically mount and request redelivery of missing messages from SafetyNet.

How do you know this worked?

To verify that you've successfully activated a lagged mailbox database copy, do one of the following:

- In the EAC, navigate to **Servers** > **Databases**. Select the appropriate database, and in the Details pane, click **View details** to view the database copy properties.
- In the Shell, run the following command to display status information for a database copy.

```
Get-MailboxDatabaseCopyStatus <DatabaseCopyName> | Format-
List
```

Remove a mailbox database copy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-06*

These procedures show you how to remove a copy of a mailbox database. You can't use these procedures to remove the last copy of a mailbox database. For detailed steps about how to remove the last copy of a mailbox database, see Remove a mailbox database or Remove-MailboxDatabase.

Looking for other management tasks related to mailbox database copies? Check out Managing mailbox database copies.

What do you need to know before you begin?

- Estimated time to complete this task: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox database copies" entry in the High availability and site resilience permissions topic.
- Mailbox database copies can only be removed from a healthy database availability group (DAG). If the DAG isn't healthy (for example, the DAG's underlying cluster is down because quorum was lost), you won't be able to remove any mailbox database copies.
- If you're removing the last passive copy of the database, continuous replication circular logging (CRCL) must not be enabled for the specified mailbox database. If CRCL is enabled, you must first disable it. After the mailbox database copy has been removed, circular logging can be enabled. Once enabled for a non-replicated mailbox database, JET circular logging is used instead of CRCL. If you aren't removing the last passive copy of a database, CRCL can remain enabled.
- After removing a database copy, you must manually delete any database and transaction log files from the server from which the database copy is being removed.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

What do you want to do?

Use the EAC to remove a mailbox database copy

1. In the EAC, go to **Servers** > **Databases**.
2. Select the mailbox database whose copy you want to remove.
3. In the Details pane, locate the passive copy you want to remove and click **Remove**.
4. Confirm the removal on the warning dialog box by clicking **yes**.
5. Click **ok** to confirm the removal after reviewing any messages.
6. Manually delete any database and transaction log files from the server from which the database copy is being removed.

Use the Shell to remove a mailbox database copy

This example removes a copy of the mailbox database DB1 from the Mailbox server MBX1.

```
Remove-MailboxDatabaseCopy -Identity DB1\MBX1 -
Confirm:$False
```

How do you know this worked?

To verify that you've successfully removed a mailbox database copy, do one of the following:

- In the EAC, navigate to **Servers** > **Databases**. Select the appropriate database, and in the Details pane, the removed passive copy is no longer listed.
- In the Shell, run the following command to verify removal of the copy.

```
Get-MailboxDatabase <DatabaseName> | Format-List
DatabaseCopies
```

The removed passive copy is no longer listed.

Monitoring database availability groups

Exchange Server 2013 > High availability and site resilience > Managing high availability and site resilience >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-01*

You can use the details in this topic for monitoring the health and status of mailbox database

copies for database availability groups (DAGs), for gathering diagnostic information, and for configuring the low disk space monitoring threshold.

Get-MailboxDatabaseCopyStatus cmdlet

You can use the Get-MailboxDatabaseCopyStatus cmdlet to view status information about mailbox database copies. This cmdlet enables you to view information about all copies of a particular database, information about a specific copy of a database on a specific server, or information about all database copies on a server. The following table describes possible values for the copy status of a mailbox database copy.

Database copy status

Database copy status	Description
Failed	The mailbox database copy is in a Failed state because it isn't suspended, and it isn't able to copy or replay log files. While in a Failed state and not suspended, the system will periodically check whether the problem that caused the copy status to change to Failed has been resolved. After the system has detected that the problem is resolved, and barring no other issues, the copy status will automatically change to Healthy.
Seeding	The mailbox database copy is being seeded, the content index for the mailbox database copy is being seeded, or both are being seeded. Upon successful completion of seeding, the copy status should change to Initializing.
SeedingSource	The mailbox database copy is being used as a source for a database copy seeding operation.
Suspended	The mailbox database copy is in a Suspended state as a result of an administrator manually suspending the database copy by running the **Suspend-MailboxDatabaseCopy** cmdlet.

Healthy	The mailbox database copy is successfully copying and replaying log files, or it has successfully copied and replayed all available log files.
ServiceDown	The Microsoft Exchange Replication service isn't available or running on the server that hosts the mailbox database copy.
Initializing	The mailbox database copy is in an Initializing state when a database copy has been created, when the Microsoft Exchange Replication service is starting or has just been started, and during transitions from Suspended, ServiceDown, Failed, Seeding, or SinglePageRestore to another state. While in this state, the system is verifying that the database and log stream are in a consistent state. In most cases, the copy status will remain in the Initializing state for about 15 seconds, but in all cases, it should generally not be in this state for longer than 30 seconds.
Resynchronizing	The mailbox database copy and its log files are being compared with the active copy of the database to check for any divergence between the two copies. The copy status will remain in this state until any divergence is detected and resolved.
Mounted	The active copy is online and accepting client connections. Only the active copy of the mailbox database copy can have a copy status of Mounted.

Dismounted	The active copy is offline and not accepting client connections. Only the active copy of the mailbox database copy can have a copy status of Dismounted.
Mounting	The active copy is coming online and not yet accepting client connections. Only the active copy of the mailbox database copy can have a copy status of Mounting.
Dismounting	The active copy is going offline and terminating client connections. Only the active copy of the mailbox database copy can have a copy status of Dismounting.
DisconnectedAndHealthy	The mailbox database copy is no longer connected to the active database copy, and it was in the Healthy state when the loss of connection occurred. This state represents the database copy with respect to connectivity to its source database copy. It may be reported during DAG network failures between the source copy and the target database copy.
DisconnectedAndResynchronizing	The mailbox database copy is no longer connected to the active database copy, and it was in the Resynchronizing state when the loss of connection occurred. This state represents the database copy with respect to connectivity to its source database copy. It may be reported during DAG network failures between the source copy and the target database copy.
FailedAndSuspended	The Failed and Suspended states have been set simultaneously by the system because a failure

	was detected, and because resolution of the failure explicitly requires administrator intervention. An example is if the system detects unrecoverable divergence between the active mailbox database and a database copy. Unlike the Failed state, the system won't periodically check whether the problem has been resolved, and automatically recover. Instead, an administrator must intervene to resolve the underlying cause of the failure before the database copy can be transitioned to a healthy state.
SinglePageRestore	This state indicates that a single page restore operation is occurring on the mailbox database copy.

The **Get-MailboxDatabaseCopyStatus** cmdlet also returns details about the in-use replication networks, including *IncomingLogCopyingNetwork*, which is returned for passive database copies, and *OutgoingConnections*, which is returned for active databases that have more than one copy, as well as any database copy being used as a source for a database seeding operation. Outgoing connection information is provided for database copies that are in file mode replication. Outgoing connection information is not provided for database copies that are in block mode replication.

Get-MailboxDatabaseCopyStatus examples

The following examples use the **Get-MailboxDatabaseCopyStatus** cmdlet. Each example pipes the results to the **Format-List** cmdlet to display the output in list format.

This example returns status information for all copies of the database DB2.

```
Get-MailboxDatabaseCopyStatus -Identity DB2 | Format-List
```

This example returns the status for all database copies on the Mailbox server MBX2.

```
Get-MailboxDatabaseCopyStatus -Server MBX2 | Format-List
```

This example returns the status for all database copies on the local Mailbox server.

```
Get-MailboxDatabaseCopyStatus -Local | Format-List
```

For more information about using the **Get-MailboxDatabaseCopyStatus** cmdlet, see Get-MailboxDatabaseCopyStatus.

Test-ReplicationHealth cmdlet

You can use the Test-ReplicationHealth cmdlet to view continuous replication status information about mailbox database copies. This cmdlet can be used to check all aspects of the replication and replay status to provide a complete overview of a specific Mailbox server in a DAG.

The **Test-ReplicationHealth** cmdlet is designed for the proactive monitoring of continuous replication and the continuous replication pipeline, the availability of Active Manager, and the health and status of the underlying cluster service, quorum, and network components. It can be run locally on or remotely against any Mailbox server in a DAG. The **Test-ReplicationHealth** cmdlet performs the tests listed in the following table.

Test-ReplicationHealth cmdlet tests

Test name	Description
ClusterService	Verifies that the Cluster service is running and reachable on the specified DAG member, or if no DAG member is specified, on the local server.
ReplayService	Verifies that the Microsoft Exchange Replication service is running and reachable on the specified DAG member, or if no DAG member is specified, on the local server.
ActiveManager	Verifies that the instance of Active Manager running on the specified DAG member, or if no DAG member is specified, the local server, is in a valid role (primary, secondary, or stand-alone).
TasksRpcListener	Verifies that the tasks remote procedure call (RPC) server is running and reachable on the specified DAG member, or if no DAG member is specified, on the local server.
TcpListener	Verifies that the TCP log copy listener is

	running and reachable on the specified DAG member, or if no DAG member is specified, on the local server.
ServerLocatorService	Verifies the Active Manager client/server processes on DAG members and on the Client Access Server that perform lookups in Active Directory and Active Manager to determine where a user's mailbox database is active.
DagMembersUp	Verifies that all DAG members are available, running, and reachable.
ClusterNetwork	Verifies that all cluster-managed networks on the specified DAG member, or if no DAG member is specified, the local server, are available.
QuorumGroup	Verifies that the default cluster group (quorum group) is in a healthy and online state.
FileShareQuorum	Verifies that the witness server and witness directory and share configured for the DAG are reachable.
DatabaseRedundancy	Verifies that there is at least one healthy copy available of the databases on the specified DAG member, or if no DAG member is specified, on the local server.
DatabaseAvailability	Verifies that the databases have sufficient availability on the specified DAG member, or if no DAG member is specified, on the local server.
DBCopySuspended	Checks whether any mailbox database copies are in a state of Suspended on the specified

	DAG member, or if no DAG member is specified, on the local server.
DBCopyFailed	Checks whether any mailbox database copies are in a state of Failed on the specified DAG member, or if no DAG member is specified, on the local server.
DBInitializing	Checks whether any mailbox database copies are in a state of Initializing on the specified DAG member, or if no DAG member is specified, on the local server.
DBDisconnected	Checks whether any mailbox database copies are in a state of Disconnected on the specified DAG member, or if no DAG member is specified, on the local server.
DBLogCopyKeepingUp	Verifies that log copying and inspection by the passive copies of databases on the specified DAG member, or if no DAG member is specified, on the local server, are able to keep up with log generation activity on the active copy.
DBLogReplayKeepingUp	Verifies that replay activity for the passive copies of databases on the specified DAG member, or if no DAG member is specified, on the local server, is able to keep up with log copying and inspection activity.

Test-ReplicationHealth example

This example uses the **Test-ReplicationHealth** cmdlet to test the health of replication for the Mailbox server MBX1.

```
Test-ReplicationHealth -Identity MBX1
```

Crimson channel event logging

Windows includes two categories of event logs: Windows logs, and Applications and Services logs. The Windows logs category includes the event logs available in previous versions of Windows: Application, Security, and System event logs. It also includes two new logs: the Setup log and the ForwardedEvents log. Windows logs are intended to store events from legacy applications and events that apply to the entire system.

Applications and Services logs are a new category of event logs. These logs store events from a single application or component rather than events that might have system-wide impact. This new category of event logs is referred to as an application's crimson channel.

The Applications and Services logs category includes four subtypes: Admin, Operational, Analytic, and Debug logs. Events in Admin logs are of particular interest if you use event log records to troubleshoot problems. Events in the Admin log should provide you with guidance about how to respond to the events. Events in the Operational log are also useful, but may require more interpretation. Admin and Debug logs aren't as user friendly. Analytic logs (which by default are hidden and disabled) store events that trace an issue, and often a high volume of events are logged. Debug logs are used by developers when debugging applications.

Exchange 2013 logs events to crimson channels in the Applications and Services logs area. You can view these channels by performing these steps:

1. Open Event Viewer.
2. In the console tree, navigate to **Applications and Services Logs** > **Microsoft** > **Exchange**.
3. Under **Exchange**, select a crimson channel, such as **HighAvailability** or **MailboxDatabaseFailureItems** to see DAG and database copy-related events, or **ActiveMontoring** or **ManagedAvailability** to see events related to Managed Availability.

The HighAvailability channel contains events related to startup and shutdown of the Microsoft Exchange Replication service, and the various components that run within the Microsoft Exchange Replication service, such as Active Manager, the third-party synchronous replication API, the tasks RPC server, TCP listener, and Volume Shadow Copy Service (VSS) writer. The HighAvailability channel is also used by Active Manager to log events related to Active Manager role monitoring and database action events, such as a database mount operation and log truncation, and to record events related to the DAG's underlying cluster.

The MailboxDatabaseFailureItems channel is used to log events associated with any failures that affect a replicated mailbox database.

The ActiveMonitoring channel contains definition and result events for Managed Availability probes, monitors and responders.

The ManagedAvailability channel contains recovery action logs and results and related events.

Low Disk Space Monitor

Exchange 2013 Managed Availability monitors hundreds of system metrics and components every minute, including the amount of free disk space on volumes used by the Mailbox server role. Prior to Exchange 2013 Service Pack 1 (SP1), Exchange monitors available space on all local volumes, including volumes that don't contain any databases or log files. In SP1 and later, only volumes that contain Exchange databases and log files are monitored. In SP1, the default threshold for the low volume space monitor is 200 GB. In Exchange 2013 Cumulative Update 6 and later, the default threshold is 180 GB. In SP1 and later, you can configure the threshold by adding the following DWORD registry value (in MB) on each Mailbox server that you want to customize:

Path: **HKEY_LOCAL_MACHINE\Software\Microsoft\ExchangeServer\v15\Replay\Parameters**

Value: *SpaceMonitorLowSpaceThresholdInMB*

For example to configure the threshold to 100 GB, you would configure the following registry value:

REG_DWORD 186a0 (100000)

After configuring or modifying the above registry value, you must restart the Microsoft Exchange DAG Management service for the change to take effect.

CollectOverMetrics.ps1 script

Exchange 2013 includes a script called CollectOverMetrics.ps1, which can be found in the Scripts folder. CollectOverMetrics.ps1 reads DAG member event logs to gather information about database operations (such as database mounts, moves, and failovers) over a specific time period. For each operation, the script records the following information:

- Identity of the database
- Time at which the operation began and ended
- Servers on which the database was mounted at the start and finish of the operation
- Reason for the operation
- Whether the operation was successful, and if the operation failed, the error details

The script writes this information to .csv files with one operation per row. It writes a separate .csv file for each DAG.

The script supports parameters that allow you to customize the script's behavior and output. For example, the results can be restricted to a specified subset by using the *Database* or *ReportFilter* parameters. Only the operations that match these filters will be included in the summary HTML report. The available parameters are listed in the following table.

CollectOverMetrics.ps1 script parameters

Parameter	Description
DatabaseAvailabilityGroup	Specifies the name of the DAG from which you want to collect metrics. If this parameter is omitted, the DAG of which the local server is a

	member will be used. Wildcard characters can be used to collect information from and report on multiple DAGs.
Database	Provides a list of databases for which the report needs to be generated. Wildcard characters are supported, for example, `-Database:"DB1","DB2"` or `-Database:"DB*"`.
StartTime	Specifies the duration of the time period to report on. The script gathers only the events logged during this period. As a result, the script may capture partial operation records (for example, only the end of an operation at the start of the period or vice-versa). If neither *StartTime* nor *EndTime* is specified, the script defaults to the past 24 hours. If only one parameter is specified, the period will be 24 hours, either beginning or ending at the specified time.
EndTime	Specifies the duration of the time period to report on. The script gathers only the events logged during this period. As a result, the script may capture partial operation records (for example, only the end of an operation at the start of the period or vice-versa). If neither *StartTime* nor *EndTime* is specified, the script defaults to the past 24 hours If only one parameter is specified, the period will be 24 hours, either beginning or ending at the specified time.
ReportPath	Specifies the folder used to store the results of event processing. If this parameter is omitted,

	the Scripts folder will be used. When specified, the script takes a list of .csv files generated by the script and uses them as the source data to generate a summary HTML report. The report is the same one that's generated with the -GenerateHtmlReport option. The files can be generated across multiple DAGs at many different times, or even with overlapping times, and the script will merge all of their data together.
GenerateHtmlReport	Specifies that the script gather all the information it has recorded, group the data by the operation type, and then generate an HTML file that includes statistics for each of these groups. The report includes the total number of operations in each group, the number of operations that failed, and statistics for the time taken within each group. The report also contains a breakdown of the types of errors that resulted in failed operations.
ShowHtmlReport	Specifies that the HTML-generated report should be displayed in a Web browser after it's generated.
SummariseCsvFiles	Specifies that the script read the data from existing .csv files that were previously generated by the script. This data is then used to generate a summary report similar to the report generated by the *GenerateHtmlReport* parameter.
ActionType	Specifies the type of operational actions the script should collect. The values for this

	parameter are `Move`, `Mount`, `Dismount`, and `Remount`. The `Move` value refers to any time that the database changes its active server, whether by controlled moves or by failovers. The `Mount`, `Dismount`, and `Remount` values refer to times that the database changes its mounted status without moving to another computer.
ActionTrigger	Specifies which administrative operations should be collected by the script. The values for this parameter are `Admin` or `Automatic`. Automatic actions are those performed automatically by the system (for example, a failover when a server goes offline). Admin actions are any actions that were performed by an administrator using either the Exchange Management Shell or the Exchange Administration Center.
RawOutput	Specifies that the script writes the results that would have been written to .csv files directly to the output stream, as would happen with write-output. This information can then be piped to other commands.
IncludedExtendedEvents	Specifies that the script collects the events that provide diagnostic details of times spent mounting databases. This can be a time-consuming stage if the Application event log on the servers is large.
MergeCSVFiles	Specifies that the script takes all the .csv files containing data about each operation and merges them into a single .csv file.

ReportFilter	Specifies that a filter should be applied to the operations using the fields as they appear in the .csv files. This parameter uses the same format as a `where` operation, with each element set to `$_` and returning a Boolean value. For example: `{$_DatabaseName -notlike "Mailbox Database*"}` can be used to exclude the default databases from the report.

CollectOverMetrics.ps1 examples

The following example collects metrics for all databases that match DB* (which includes a wildcard character) in the DAG DAG1. After the metrics are collected, an HTML report is generated and displayed.

```
CollectOverMetrics.ps1 -DatabaseAvailabilityGroup DAG1 -
Database:"DB*" -GenerateHTMLReport -ShowHTMLReport
```

The following examples demonstrate ways that the summary HTML report may be filtered. The first uses the *Database* parameter, which takes a list of database names. The summary report then contains data only about those databases. The next two examples use the *ReportFilter* option. The last example filters out all the default databases.

```
CollectOverMetrics.ps1 -SummariseCsvFiles (dir *.csv) -
Database MailboxDatabase123,MailboxDatabase456
CollectOverMetrics.ps1 -SummariseCsvFiles (dir *.csv) -
ReportFilter { $_.DatabaseName -notlike "Mailbox Database*"
}
CollectOverMetrics.ps1 -SummariseCsvFiles (dir *.csv) -
ReportFilter { ($_.ActiveOnStart -like "ServerXYZ*") -and
($_.ActiveOnEnd -notlike "ServerXYZ*") }
```

CollectReplicationMetrics.ps1 script

CollectReplicationMetrics.ps1 is another health metric script included in Exchange 2013. This script provides an active form of monitoring because it collects metrics in real time, while the script is running. CollectReplicationMetrics.ps1 collects data from performance counters related to database replication. The script gathers counter data from multiple Mailbox servers, writes each server's data to a .csv file, and then reports various statistics across all of this data (for example, the

amount of time each copy was failed or suspended, the average copy or replay queue length, or the amount of time that copies were outside of their failover criteria).

You can either specify the servers individually, or you can specify entire DAGs. You can either run the script to first collect the data and then generate the report, or you can run it to just gather the data or to only report on data that's already been collected. You can specify the frequency at which data should be sampled and the total duration to gather data.

The data collected from each server is written to a file named **CounterData.<ServerName>.<TimeStamp>.csv**. The summary report will be written to a file named **HaReplPerfReport.<DAGName>.<TimeStamp>.csv**, or **HaReplPerfReport.<TimeStamp>.csv** if you didn't run the script with the *DagName* parameter.

The script starts Windows PowerShell jobs to collect the data from each server. These jobs run for the full period in which data is being collected. If you specify a large number of servers, this process can use a considerable amount of memory. The final stage of the process, when data is processed into a summary report, can also be quite time consuming for large amounts of data. It's possible to run the collection stage on one computer, and then copy the data elsewhere for processing.

The CollectReplicationMetrics.ps1 script supports parameters that allow you to customize the script's behavior and output. The available parameters are listed in the following table.

CollectReplicationMetrics.ps1 script parameters

Parameter	Description
DagName	Specifies the name of the DAG from which you want to collect metrics. If this parameter is omitted, the DAG of which the local server is a member will be used.
DatabaseNames	Provides a list of databases for which the report needs to be generated. Wildcard characters are supported for use, for example, `-DatabaseNames:"DB1","DB2"` or `-DatabaseNames:"DB*"`.
ReportPath	Specifies the folder used to store the results of event processing. If this parameter is omitted, the Scripts folder will be used.
Duration	Specifies the amount of time the collection process should run. Typical values would be one

	to three hours. Longer durations should be used only with long intervals between each sample or as a series of shorter jobs run by scheduled tasks.
Frequency	Specifies the frequency at which data metrics are collected. Typical values would be 30 seconds, one minute, or five minutes. Under normal circumstances, intervals that are shorter than these won't show significant changes between each sample.
Servers	Specifies the identity of the servers from which to collect statistics. You can specify any value, including wildcard characters or GUIDs.
SummariseFiles	Specifies a list of .csv files to generate a summary report. These files are the files named **CounterData.<CounterData>*** and are generated by the CollectReplicationMetrics.ps1 script.
Mode	Specifies the processing stages that the script executes. You can use the following values: • `CollectAndReport` This is the default value. This value signifies that the script should both collect the data from the servers and then process them to produce the summary report. • `CollectOnly` This value signifies that the script should just collect the data and not produce the report. • `ProcessOnly` This value signifies that the script should import data from a set of .csv files and process them to produce the summary report. The *SummariseFiles* parameter is used to provide the script with the list of files to process.
MoveFilestoArchive	Specifies that the script should move the files to a compressed folder after processing.

LoadExchangeSnapin	Specifies that the script should load the Shell commands. This parameter is useful when the script needs to run from outside the Shell, such as in a scheduled task.

CollectReplicationMetrics.ps1 example

The following example gathers one hour's worth of data from all the servers in the DAG DAG1, sampled at one minute intervals, and then generates a summary report. In addition, the *ReportPath* parameter is used, which causes the script to place all the files in the current directory.

```
CollectReplicationMetrics.ps1 -DagName DAG1 -Duration
"01:00:00" -Frequency "00:01:00" -ReportPath
```

The following example reads the data from all the files matching CounterData* and then generates a summary report.

```
CollectReplicationMetrics.ps1 -SummariseFiles (dir
CounterData*) -Mode ProcessOnly -ReportPath
```

Switchovers and Failovers

Applies to: *Exchange Server 2013 SP1, Exchange Server 2013*

Topic Last Modified: *2014-06-23*

Switchovers and failovers are the two forms of outages in Microsoft Exchange Server 2013:

- A *switchover* is a scheduled outage of a database or server that's explicitly initiated by an administrator, typically in preparation for performing a maintenance operation. Switchovers involve an administrator moving the active mailbox database copy to another server in the database availability group (DAG).
- A *failover* refers to unexpected events that result in the unavailability of services, data, or both. A failover involves the system automatically recovering from the failure by activating a passive mailbox database copy to make it the active mailbox database copy.

Exchange 2013 is specifically designed to handle both switchovers and failovers.

Looking for management tasks related to high availability and site resilience? See Managing high availability and site resilience.

Switchovers

There are three types of switchovers in Exchange 2013:

- Database switchovers
- Server switchovers
- Datacenter switchovers

Database Switchovers

A *database switchover* is the process by which an individual active database is switched over to another database copy (a passive copy), and that database copy is made the new active database copy. Database switchovers can happen both within and across datacenters. A database switchover can be performed by using the Exchange Admin Center (EAC) or the Shell. Regardless of which interface is used, the switchover process is as follows:

1. The administrator initiates a database switchover to move the current active mailbox database copy to another server.
2. The client used for the task makes an RPC call to the Microsoft Exchange Replication service on a DAG member.
3. If the DAG member doesn't hold the Primary Active Manager (PAM) role, the DAG member refers the task to the server that holds the PAM role.
4. The task makes an RPC call to the Microsoft Exchange Replication service on the server that holds the PAM role.
5. The PAM reads and updates the database location information that's stored in the cluster database for the DAG.
6. The PAM contacts the Microsoft Exchange Replication service on the DAG member whose passive copy is being activated as the new active mailbox database copy.
7. The Microsoft Exchange Replication service on the target server queries the Microsoft Exchange Replication services on all other DAG members to determine the best log source for the database copy.
8. The database is dismounted from the current server and the Microsoft Exchange Replication service on the target server copies the remaining logs to the target server.
9. The Microsoft Exchange Replication service on the target server requests a database mount.
10. The Microsoft Exchange Information Store service on the target server replays the log files and mounts the database.
11. Any error codes are returned to the Microsoft Exchange Replication service on the target server.
12. The PAM updates the database copy state information in the cluster database for the DAG.
13. Any error codes are returned by the Microsoft Exchange Replication service on the target server to the Microsoft Exchange Replication service on the PAM.
14. The Microsoft Exchange Replication service on the PAM returns any errors to the administrative interface where the task was called.
15. Remote PowerShell returns the results of the operation to the calling administrative interface.

For detailed steps about how to perform a database switchover, see Activate a mailbox database copy.

Server Switchovers

A server switchover is the process by which all active databases on a DAG member are activated on one or more other DAG members. Like database switchovers, a server switchover can occur both within a datacenter and across datacenters, and it can be initiated by using both the EAC and the Shell. Regardless of which interface is used, the server switchover process is as follows:

1. The administrator initiates a server switchover to move all current active mailbox database copies to one or more other servers.
2. The task performs the same steps described earlier in this topic for database switchovers (Steps 2 through 4) for each of the active databases on the current server.
3. The PAM reads and updates the database location information that's stored in the cluster database for the DAG.
4. The PAM contacts the Microsoft Exchange Replication service on each DAG member that has a passive copy being activated.
5. The Microsoft Exchange Replication service on the target servers query the Microsoft Exchange Replication services on all other DAG members to determine the best log source for the database copy.
6. The database is dismounted from the current server and the Microsoft Exchange Replication service on each target server copies the remaining logs.
7. The Microsoft Exchange Replication service on each target server requests a database mount.
8. The Microsoft Exchange Information Store service on each target server replays the log files and mounts the database.
9. Any error codes are returned to the Microsoft Exchange Replication service on the target server.
10. The PAM updates the database copy state information in the cluster database for the DAG.
11. Any error codes are returned by the Microsoft Exchange Replication service on the target server to the Microsoft Exchange Replication service on the PAM.
12. The Microsoft Exchange Replication service on the PAM returns any errors to the administrative interface where the task was called.
13. Remote PowerShell returns the results of the operation to the calling administrative interface.

For detailed steps about how to perform a server switchover, see Perform a Server Switchover.

Datacenter Switchovers

In a site resilient configuration, automatic recovery in response to a site-level failure can occur within a DAG, allowing the messaging system to remain in a functional state. This configuration requires at least three locations, as it requires deploying DAG members in two locations and the DAG's witness server in a third location.

If you don't have three locations, or even if you do have three locations, but you want to control

datacenter-level recovery actions, you can configure a DAG for manual recovery in the event of a site-level failure. In that event, you would perform a process called a *datacenter switchover*. As with many disaster recovery scenarios, prior planning and preparation for a datacenter switchover can simplify your recovery process and reduce the duration of your outage.

Because of the numerous architectural changes in Exchange 2013, including the consolidation of server roles, performing a datacenter switchover in Exchange 2013 is significantly easier than it was in Exchange 2010. For detailed steps to performing a datacenter switchover, see Datacenter Switchovers.

Failovers

A failover is an automatic activation process that can occur at the database, server, or datacenter level. Failovers occur in response to a failure that affects an individual database (for example, an isolated storage loss) an entire server (for example, a motherboard failure or a loss of power), or an entire site (for example, the loss of all DAG members in a site).

DAGs and mailbox database copies provide full redundancy and rapid recovery of both the data and the services that provide access to the data. The following table lists the expected recovery actions for a variety of failures. Some failures require the administrator to initiate the recovery, and other failures are automatically handled by the system.

Description	Automatic activation	Automatic repair action	State during repair: Active	State during repair: Passive	Repair actions	Comments
Extensible Storage Engine (ESE) soft database failure: The drives storing the database are returning errors on some reads (for example, a -1018 error).	Possible short outage. Possible automatic failover.	Automatic patching of bad page.	Manual switchover, automatic failover, or online repair.	Failed	RAID rebuild, database and database copy repair, restore and run recovery then page patching, or page patching from copy.	There may be other soft database failure codes. Doesn't include NTFS file system block failures. If failover or switchover is performed,

						host server is updated.
ESE "*semi-soft*" database failure: The drives storing the database are returning errors on some writes.	Short outage during automatic failover.	Automatic volume/disk rebuilt after possible drive replacement.	Dismounted if can't be recovered.	Failed	RAID rebuild may solve the problem. Copy and repair, restore and run recovery, or volume/disk rebuilt after possible replacement.	An ESE semi-soft write error means some writes are successful. Doesn't include an NTFS block failure.
ESE "semi-soft" log failure: The drives storing the log data are returning non-recovered errors on some reads or writes.	Short outage during automatic failover.	Automatic volume/disk rebuilt after possible drive replacement.	Dismounted if can't be recovered.	Failed	RAID rebuild may solve the problem. Copy and repair, restore and run recovery, or volume/disk rebuilt after possible replacement.	An ESE semi-soft read/write error means some reads/ writes are successful. If the database fails, automated recovery will occur before log data recovery processing starts.

ESE software error or resource exhaustion: An error where ESE terminates instance (for example, Event ID 1022, checkpoint depth too deep).	Short outage during automatic failover.	None.	Dismounted if can't be recovered.	Failed	Fix underlying resource issue.	This failure could be the surfaced error of other cases.
NTFS block failures: The drives storing the database or logs experiences a read or write error to an NTFS control structure.	Short outage during automatic failover.	Volume completely rebuilt after possible drive replacement.	Dismounted if can't be recovered.	Failed	RAID rebuild may solve the problem. NTFS utilities may solve the NTFS problems. Exchange recovery may be required.	This is more likely to occur when RAID isn't in use. If this impacts the active log volume, some recent log files will be lost. Doesn't include errors automatically corrected by NTFS or its underlying software or hardware

						stack.
Database or log drive failure: A drive storing the database or logs has completely failed and is inaccessible.	Short outage during automatic failover.	Drive reformatted or replaced, followed by complete volume rebuild.	Dismounted if can't be recovered.	Failed	Drive replacement followed by possible RAID rebuild. Drive replacement followed by complete volume rebuild. Complete volume rebuild.	Not applicable.
Database or log volume failure: The volume fails due to NTFS or lower level volume issues.	Short outage during automatic failover.	Drive reformatted or replaced.	Dismounted if can't be recovered.	Failed	Drive replacement followed by possible RAID rebuild. Drive replacement followed by complete volume rebuild. Complete volume rebuild.	Not applicable.

Database or log volume out of space: The NTFS file system with the database or log files is out of space.	Automatic failover if other copy isn't in similar state.	None.	Dismounted.	Failed	Run full or incremental backups, manually delete logs, let time pass, resume database copy, or repair failed database copy.	Not applicable.
Administrator dismounts the wrong database.	If automatic failover isn't blocked by the administrator, there will be a short outage. If automatic failover is prevented, there will be an outage until the database is mounted.	None.	Dismounted.	Not applicable	Administrator corrects the error.	Not applicable.
Administrator suspends the wrong database	Depending on configuratio	None.	Not applicable.	Suspended	Administrator corrects the error.	Not applicable.

copy.	n and impacted copy, auto recovery may be prevented.					
Administrator dismounts a database for storage, NTFS, or volume maintenance.	If automatic failover isn't blocked by the administrator, there will be a short outage. If automatic failover is blocked, there will be an outage until the administrator completes the task.	None.	Dismounted.	Not applicable	Administrator completes the task.	Not applicable.
Administrator suspends a database copy for storage, NTFS, or volume maintenance.	Depending on configuration and impacted copy, auto recovery may be prevented.	None.	Not applicable.	Suspended	Administrator completes the actions.	Not applicable.

Administrator dismounts a database for offline database maintenance.	Outage until repaired.	None.	Dismounted.	Suspended	Administrator completes the actions.	Active and passive database copies are diverged. Administrator must suspend copies.
Storage area network (SAN), disk, or storage controller failure.	Short outage during automatic failover.	None.	Dismounted.	Any	Repair hardware.	A passive database copy will be in the state that existed at the time when the system failed.
Server hardware maintenance.	Short outage during automatic failover (unless blocked by an administrator).	None.	Dismounted.	Any	Complete actions.	A passive database copy will be in the state that existed at the time when the system was shut down.
Server software maintenance.	Short outage during	None.	Dismounted.	Any	Complete actions.	A passive database copy will be

	automatic failover (unless blocked by an administrator).					in the state that existed at the time when the system was shut down.
Microsoft Exchange Information Store service is stopped or paused by an administrator.	None.	None.	Dismounted.	Any	Restart the Microsoft Exchange Information Store service.	A passive database copy will be in the state that existed at the time when the service was stopped.
Microsoft Exchange Information Store service fails; operating system is still running.	Short outage during automatic failover.	Service Control Manager restarts the Microsoft Exchange Information Store service.	Dismounted.	Any	Manually or automatically restart the Microsoft Exchange Information Store service.	A passive database copy will be in the state that existed when the Microsoft Exchange Information Store service failed.
Partial Microsoft Exchange Information Store service	Possible short outage during automatic	None.	Mounted and partially functional.	Any, but may be only partially functional	Restart server, operating system, or	Not applicable.

failure; some part of the Exchange store stops functioning, but it's not identified as completely failed.	failover.				Microsoft Exchange Information Store service.	
Server failure: The server fails for one of the following reasons: • Complete power failure • Unrecovered failure of the processor chip, motherboard, or backplane • Operating system stop error • Operating system stops responding • Complete communicatio n failure	Short outage during automatic failover.	Restart computer.	Dismounted.	Any	Restore power, change operating system settings, change hardware settings, replace hardware, restart operating system, service operating system, service hardware, or repair communicat ion problems.	Not applicable.

DAG experiences a quorum failure.	Outage until repaired.	None.	Dismounted.	Any	Repair failed quorum, assign new quorum, or restore the network that's causing quorum failure.	A passive database copy will be in the state that existed at the time when the system failed.
MAPI network communication failure: The server is no longer available on the MAPI network.	Short outage during automatic failover; must be lossless.	None. Communication continues to be attempted.	Dismounted.	Any	Fix communication problem by correcting hardware or software issues.	Not applicable.
Replication network communication failure: The server can't receive heartbeats, log copies, or seed through the failed replication network.	Possible short copying or seeding outage while the workload is switched to other network.	None. Communication continues to be attempted.	None.	Any	Fix communication problem by correcting hardware or software issues.	Resiliency impacted by failure.
Multiple network	Short outage	None. Communicat	Dismounted.	Any	Fix communicat	At least one network is

communication failure: The server can’t receive heartbeats, log copies, or seed through multiple networks.	during automatic failover; must be lossless.	ion continues to be attempted.			ion problem by correcting hardware or software issues.	still functional.
Partial failure of one or more networks: Networks experience high error rates.	Failure not detected; no action.	None.	Mounted, but possible performance issues.	Any	Fix communicat ion problem by correcting hardware or software issues.	Network experiences higher than normal error rates.
Undetected operating system hang: Operating system stops responding but it's not detected by monitoring or clustering.	None.	None.	Any.	Any	Restart or terminate the resources that aren't responding.	Hang isn't detected so no action is taken. Some functionality may be operational.
Operating system drive experiences a failure.	Short outage during automatic failover.	None.	Dismounted.	Any	Replace drive and rebuild server or rebuild	Not applicable.

					volume by using RAID.	
Operating system drive out of space.	Short outage during automatic failover.	None.	Dismounted.	Any	Manually free space on the volume.	Not applicable.
Drive containing Exchange binaries experiences a volume or drive failure.	Short outage during automatic failover.	None.	Dismounted.	Any	Replace drive and reinstall application or rebuild volume by using RAID.	Not applicable.
Drive containing the Exchange binaries is out of space.	Short outage during automatic failover.	None.	Dismounted.	Any	Manually free space on the volume.	Not applicable.
Invalid new log detected: The log sequence is disrupted by an existing file.	Short outage during automatic failover; assume other copies don't have the same problem.	None.	Dismounted.	Failed	Remove disruptive logs after determining source.	The disruptive logs shouldn't replicate.
Continuous	Not	Discard log.	Not	Failed	Discard	Not

replication detects invalid log: Replay detects an inappropriate log during copy or replay.	applicable.		applicable.		invalid log; move impacting log stream.	applicable.

Database Failovers

A database failover occurs when a database copy that was active is no longer able to remain active. The following occurs as part of a database failover:

1. The database failure is detected by the Microsoft Exchange Information Store service.
2. The Microsoft Exchange Information Store service writes failure events to the crimson channel event log.
3. The Active Manager on the server that contains the failed database detects the failure events.
4. The Active Manager requests the database copy status from the other servers that hold a copy of the database.
5. The other servers return the requested database copy status to the requesting Active Manager.
6. The PAM initiates a move of the active database to another server in the DAG using a best copy selection algorithm.
7. The PAM updates the database mount location in the cluster database to refer to the selected server.
8. The PAM sends a request to the Active Manager on the selected server to become the database master.
9. The Active Manager on the selected server requests that the Microsoft Exchange Replication service attempt to copy the last logs from the previous server and set the mountable flag for the database.
10. The Microsoft Exchange Replication service copies the logs from the server that previously had the active copy of the database.
11. The Active Manager reads the maximum log generation number from the cluster database.
12. The Microsoft Exchange Information Store service mounts the new active database copy.

Server Failovers

A server failover occurs when the DAG member is no longer able to service the MAPI network, or when the Cluster service on a DAG member is no longer able to contact the remaining DAG members. The following occurs as part of a server failover:

1. The Cluster service on the PAM sends a notification to the PAM for one of two conditions:
   a. **Node Down** The server is reachable but is unable to participate in DAG operations.

   b. **MAPI Network Down** The server can't be contacted over the MAPI network and therefore can't participate in DAG operations.
2. If the server is reachable, the PAM contacts the Active Manager on the affected server and requests that all databases be immediately dismounted.
3. For each affected database copy:
   a. The PAM requests the database copy status from all servers in the DAG.
   b. The PAM receives a response from all reachable and active DAG members.
   c. The PAM tries to determine the best log source among all responding servers by querying the most recent log generation number from each of the responders.
   d. Each of the servers responds with the log generation number.
4. The PAM retrieves the current search index catalog status from the cluster database.
5. Based on the log generation number and catalog health of each database copy, the PAM selects the best copies to activate.
6. The PAM updates the mounted location of the database in the cluster database.
7. The PAM initiates database failover by communicating with the Active Manager on one or more other servers.
8. The Active Manager on the selected servers requests that the Microsoft Exchange Replication service attempt to copy the last logs from the previous server and set the mountable flag.
9. When the database is mountable, the Active Manager on the servers mounts the databases.

For more information about Active Manager's best copy selection process, see Active Manager.

Datacenter Failovers

Significant changes have been made in Exchange 2013 that address the challenges with an Exchange 2010 site resilience configuration. With the namespace simplification, consolidation of server roles, separation of Client Access server and DAG recovery (in Exchange 2013, the namespace does not need to move with the DAG), and the changes around load balancing, Exchange 2013 provides new site resilience options, such as the ability to use a single global namespace. In addition, if you have more than two locations in which to deploy messaging service components, Exchange 2013 also provides the ability to configure the messaging service for automatic failover in response to failures that required manual intervention in Exchange 2010.

Site resilience has been operationally simplified in Exchange 2013. Exchange leverages fault tolerance built into the namespace through multiple IP addresses, load balancing (and if need be, the ability to take servers in and out of service). One of the most significant changes we made in Exchange 2013 was to leverage the clients' ability to cache multiple IP addresses returned from a DNS server in response to a name resolution request. Assuming the client has the ability to cache multiple IP addresses (which almost all HTTP clients do, and since almost all of the client access protocols in Exchange 2013 are HTTP based (Outlook, Outlook Anywhere, EAS, EWS, OWA, EAC, RPS, etc.), all supported HTTP clients have the ability to use multiple IP addresses), thereby providing failover on the client side. You can configure DNS to hand multiple IP addresses to a client during name resolution. The client asks for mail.contoso.com and gets back two IP addresses,

or four IP addresses, for example. However many IP addresses the client gets back will be used reliably by the client. This makes the client a lot better off because if one of the IP addresses fails, the client has one or more others to try to connect to. If a client tries one and it fails, it waits around 20 seconds and then tries the next one in the list. Thus, if you lose connectivity to your primary CAS array, and you have a second published IP address for a second CAS array, recovery for the clients happens automatically (and in about 21 seconds).

Modern HTTP clients (operating systems and Web browsers that are ten years old or less) simply work with this redundancy automatically. The HTTP stack can accept multiple IP addresses for an FQDN, and if the first IP it tries fails hard (e.g., cannot connect), it will try the next IP in the list. In a soft failure (connect lost after session established, perhaps due to an intermittent failure in the service where, for example, a device is dropping packets and needs to be taken out of service), the user might need to refresh their browser.

With the proper configuration, failover can happen at the client level and clients will be automatically redirected to a second datacenter that has operating Client Access servers, and those operating Client Access servers will proxy the communication back to the user's Mailbox server, which remains unaffected by the outage (because you don't do a switchover). Instead of working to recover service, the service recovers itself and you can focus on fixing the core issue (e.g., replacing a failed load balancer).

Since you can failover the namespace between datacenters, all that is needed to achieve a datacenter failover is a mechanism for failover of the Mailbox role across datacenters. To get automatic failover for the DAG, you simply architect a solution where the DAG is evenly split between two datacenters, and then place the witness server in a third location so that it can be arbitrated by DAG members in either datacenter, regardless of the state of the network between the datacenters that contain the DAG members. The key is that third location is isolated from network failures that affect the locations containing the DAG members.

Datacenter Switchovers

Applies to: *Exchange Server 2013 SP1, Exchange Server 2013*

Topic Last Modified: *2014-08-26*

In a site resilient configuration, automatic recovery in response to a site-level failure can occur within a DAG, allowing the messaging system to remain in a functional state. This configuration requires at least three locations, as it requires deploying DAG members in two locations and the DAG's witness server in a third location.

If you don't have three locations, or even if you do have three locations, but you want to control

datacenter-level recovery actions, you can configure a DAG for manual recovery in the event of a site-level failure. In that event, you would perform a process called a *datacenter switchover*. As with many disaster recovery scenarios, prior planning and preparation for a datacenter switchover can simplify your recovery process and reduce the duration of your outage.

There are four basic steps that you complete to perform a datacenter switchover, after making the initial decision to activate the second datacenter:

1. **Terminate a partially running datacenter** This step involves terminating Exchange services in the primary datacenter, if any services are still running. This is particularly important for the Mailbox server role because it uses an active/passive high availability model. If services in a partially failed datacenter aren't stopped, it's possible for problems from the partially failed datacenter to negatively affect the services during a switchover back to the primary datacenter.

Important:
If network or Active Directory infrastructure reliability has been compromised as a result of the primary datacenter failure, we recommend that all messaging services be off until these dependencies are restored to healthy service.

2. **Validate and confirm the prerequisites for the second datacenter** This step can be performed in parallel with step 1 because validation of the health of the infrastructure dependences in the second datacenter is largely independent of the first datacenter services. Each organization typically requires its own method for performing this step. For example, you may decide to complete this step by reviewing health information collected and filtered by an infrastructure monitoring application, or by using a tool that's unique to your organization's infrastructure. This is a critical step, because activating the second datacenter when its infrastructure is unhealthy and unstable is likely to yield poor results.
3. **Activate the Mailbox servers** This step begins the process of activating the second datacenter. This step can be performed in parallel with step 4 because the Microsoft Exchange services can handle database outages and recover. Activating the Mailbox servers involves a process of marking the failed servers from the primary datacenter as unavailable followed by activation of the servers in the second datacenter. The activation process for Mailbox servers depends on whether the DAG is in database activation coordination (DAC) mode. For more information about database activation coordination mode, see Datacenter Activation Coordination mode.

If the DAG is in DAC mode, you can use the Exchange site resilience cmdlets to terminate a partially failed datacenter (if necessary) and activate the Mailbox servers. For example, in DAC mode, this step is performed by using the Stop-DatabaseAvailabilityGroup cmdlet. In some cases, the servers must be marked as unavailable twice (once in each datacenter). Next, the Restore-DatabaseAvailabilityGroup cmdlet is run to restore the remaining members of the database availability group (DAG) in the second datacenter by reducing the DAG members to those that are still operational, thereby reestablishing quorum. If the DAG isn't in DAC mode, you must use the Windows Failover Cluster tools to activate the Mailbox servers. After either process is complete, the database copies that were previously passive in the second datacenter can become active and be mounted. At this point, Mailbox server recovery is complete.

4. **Activate the Client Access servers** This involves using the URL mapping information and the

Domain Name System (DNS) change methodology to perform all required DNS updates. The mapping information describes what DNS changes to perform. The amount of time required to complete the update depends on the methodology used and the Time to Live (TTL) settings on the DNS record (and whether the deployment's infrastructure honors the TTL).

Users should start to have access to messaging services sometime after steps 3 and 4 are completed. Steps 3 and 4 are described in greater detail later in this topic.

Contents

Terminating a Partially Failed Datacenter

If any DAG members in the failed datacenter are still running, they should be terminated.

When the DAG is in DAC mode, the specific actions to terminate any surviving DAG members in the primary datacenter are as follows:

1. The DAG members in the primary datacenter must be marked as stopped in the primary datacenter. *Stopped* is a state of Active Manager that prevents databases from mounting, and Active Manager on each server in the failed datacenter is put into this state by using the Stop-DatabaseAvailabilityGroup cmdlet. The *ActiveDirectorySite* parameter of this cmdlet can be used to mark all of the servers in the primary datacenter as stopped with a single command. This step may not be possible depending on the failure. This step should be taken if the state of the datacenter permits it. The **Stop-DatabaseAvailabilityGroup** cmdlet should be run against all servers in the primary datacenter. If the Mailbox server is unavailable but Active Directory is operating in the primary datacenter, the **Stop-DatabaseAvailabilityGroup** command with the *ConfigurationOnly* parameter must be run against all servers in this state in the primary datacenter, or the Mailbox server must be turned off. Failure to either turn off the Mailbox servers in the failed datacenter or to successfully perform the **Stop-DatabaseAvailabilityGroup** command against the servers will create the potential for split-brain syndrome to occur across the two datacenters. You may need to individually turn off computers through power management devices to satisfy this requirement.
2. The second datacenter must now be updated to represent which primary datacenter servers are stopped. This is done by running the same **Stop-DatabaseAvailabilityGroup** command with the *ConfigurationOnly* parameter using the same *ActiveDirectorySite* parameter and specifying the name of the Active Directory site in the failed primary datacenter. The purpose of this step is to inform the servers in the second datacenter about which mailbox servers are available to use when restoring service.

When the DAG isn't in DAC mode, the specific actions to terminate any surviving DAG members in the primary datacenter are as follows:

1. The DAG members in the primary datacenter must be forcibly evicted from the DAG's underlying cluster by running the following commands on each member:

```
net stop clussvc
cluster <DAGName> node <DAGMemberName> /forcecleanup
```

2. The DAG members in the second datacenter must now be restarted and then used to complete the eviction process from the second datacenter. Stop the Cluster service on each DAG member in the second datacenter by running the following command on each member:

```
net stop clussvc
```

3. On a DAG member in the second datacenter, force a quorum start of the Cluster service by running the following command:

```
net start clussvc /forcequorum
```

4. Open the Failover Cluster Management tool and connect to the DAG's underlying cluster. Expand the cluster, and then expand **Nodes**. Right-click each node in the primary datacenter, select **More Actions**, and then select **Evict**. When you're done evicting the DAG members in the primary datacenter, close the Failover Cluster Management tool.

If any Unified Messaging services are in use in the failed datacenter, they must be disabled to prevent call routing to the failed datacenter. You can disable Unified Messaging services by using the Disable-UMService cmdlet (for example, `Disable-UMService EX1`). Alternatively, if you are using a Voice over IP (VoIP) gateway, you can also remove the server entries from the VoIP gateway, or change the DNS records for the failed servers to point to the IP address of the servers in the second datacenter if your VoIP gateway is configured to route calls using DNS.

Return to top

Activating Mailbox Servers

The steps needed to activate Mailbox servers during a datacenter switchover also depend on whether the DAG is in DAC mode. Before activating the DAG members in the second datacenter, we recommend that you validate that the infrastructure services in the second datacenter are ready for messaging service activation.

When the DAG is in DAC mode, the steps to complete activation of the mailbox servers in the second datacenter are as follows:

1. The Cluster service must be stopped on each DAG member in the second datacenter. You can use the **Stop-Service** cmdlet to stop the service (for example, `Stop-Service ClusSvc`), or use `net stop clussvc` from an elevated command prompt.
2. The Mailbox servers in the standby datacenter are then activated by using the Restore-

DatabaseAvailabilityGroup cmdlet. The Active Directory site of the standby datacenter is passed to the **Restore-DatabaseAvailabilityGroup** cmdlet to identify which servers to use to restore service and to configure the DAG to use an alternate witness server. If the alternate witness server wasn't previously configured, you can configure it by using the *AlternateWitnessServer* and *AlternateWitnessDirectory* parameters of the Restore-DatabaseAvailabilityGroup cmdlet. If this command succeeds, the quorum criteria are shrunk to the servers in the standby datacenter. If the number of servers in that datacenter is an even number, the DAG will switch to using the alternate witness server as identified by the setting on the DAG object.

3. The databases can now be activated. Depending on the specific configuration used by the organization, this may not be automatic. If the servers in the standby datacenter have an activation blocked setting, the system won't do an automatic failover from the primary datacenter to the standby datacenter of any database. If no failover restrictions are present for any of the database copies in the standby datacenter, the system will activate copies in the second datacenter assuming they are healthy. If databases are configured with an activation blocked setting that requires explicit manual action, there are two choices for action:
   a. Clear the setting that blocks activation. This will make the system return to its default behavior, which is to activate any available copy.
   b. Leave the setting unchanged and use the Move-ActiveMailboxDatabase cmdlet to complete the database activation in the second datacenter. To complete this step using the **Move-ActiveMailboxDatabase** cmdlet when activation blocked is set, you must explicitly identify the target of the move.
4. The last step is to review all error and warning messages from the tasks. Any indicated warnings should be followed up and corrected. The task design model for these commands is to only fail if they can't achieve the fundamental goal of their design. For example, the **Restore-DatabaseAvailabilityGroup** cmdlet will fail if it can't shrink the quorum of the DAG to allow a server in the second datacenter to be restarted for servicing without causing a quorum outage. However, each task's output is also used to identify the issues that require administrator follow-up. You're strongly encouraged to save all task output and review it for follow-up actions.

When the DAG isn't in DAC mode, the steps to complete activation of the mailbox servers in the second datacenter are as follows:

1. The quorum must be modified based on the number of DAG members in the second datacenter.
   a. If there's an odd number of DAG members, change the DAG quorum model from a Node a File Share Majority to a Node Majority quorum by running the following command:

```
cluster <DAGName> /quorum /nodemajority
```

   b. If there's an even number of DAG members, reconfigure the witness server and directory by running the following command in the Exchange Management Shell:

```
Set-DatabaseAvailabilityGroup <DAGName> -WitnessServer
<ServerName>
```

2. Start the Cluster service on any remaining DAG members in the second datacenter by running the following command:

```
net start clussvc
```

3. Perform server switchovers to activate the mailbox databases in the DAG by running the following command for each DAG member:

```
Move-ActiveMailboxDatabase -Server <DAGMemberinPrimarySite>
-ActivateOnServer <DAGMemberinSecondSite>
```

4. Mount the mailbox databases on each DAG member in the second site by running the following command:

```
Get-MailboxDatabase <DAGMemberinSecondSite> | Mount-
Database
```

Return to top

Activating Client Access Servers

Clients connect to service endpoints (for example Outlook Web App, Autodiscover, Exchange ActiveSync, Outlook Anywhere, POP3, IMAP4, and the RPC Client Access array) to access Exchange services and data. Therefore, activating Client Access servers involves changing the mapping of the DNS records for these service endpoints from IP addresses in the primary datacenter to the IP addresses in the second datacenter that are configured as the new service endpoints. Depending on your DNS configuration, the DNS records that need to be modified may or may not be in the same DNS zone.

Activating Client Access Servers

Clients will then automatically connect to the new service endpoints in one of two ways:

- Clients will continue to try to connect, and should automatically connect after the TTL has expired for the original DNS entry, and after the entry is expired from the client's DNS cache. Users can also run the `ipconfig /flushdns` command from a command prompt to manually clear their DNS cache.
- Clients starting or restarting will perform a DNS lookup on startup and will get the new IP address for the service endpoint, which will be a Client Access server or array in the second datacenter.

Assuming that all appropriate configuration changes have been completed to define and configure the services in the second datacenter to function as they were in the primary datacenter, and assuming that the established DNS configuration is correct, no further changes should be needed to activate Client Access servers.

Activating Transport Services

Clients and other servers that submit messages typically identify those servers using DNS. Activating

transport services in the second datacenter involves changing DNS records to point to the IP addresses of the Mailbox servers in the second datacenter. Clients and sending servers will then automatically connect to the servers in the second datacenter in one of two ways:

- Clients will continue to try to connect, and should automatically connect after the TTL has expired for the original DNS entry, and after the entry is expired from the client's DNS cache. Users can also run the `ipconfig /flushdns` command from a command prompt to manually clear their DNS cache.
- Clients starting or restarting will perform a DNS lookup on startup and will get the new IP address for the SMTP endpoint, which will be a Mailbox server in the second datacenter.

Assuming that all appropriate configuration changes have been completed to define and configure the services in the second datacenter to function as they were in the primary datacenter, and assuming that the established DNS configuration is correct, no further changes should be needed to activate transport services.

Activating Unified Messaging Services

Unified Messaging (UM) services connect to an organization's PBX system and phone lines. The logical connection between the PBX system and the Unified Messaging service is provided by an IP gateway. IP gateways include high availability functionality and are able to switch between multiple Unified Messaging services when a failure is detected.

If there are Unified Messaging services in the second datacenter that were in a disabled state because they are dedicated to the site resilience solution, they can be enabled by using the Enable-UMService cmdlet (for example, `Enable-UMService EX4`).

Assuming the IP gateways are associated with Unified Messaging services by using DNS servers, activating Unified Messaging services therefore involves changing DNS records to point to the new IP addresses that will be configured for the Unified Messaging service in the second datacenter. Assuming that all appropriate configuration changes have been completed to define and configure the services in the second datacenter to function as they were in the primary datacenter, and assuming that the established DNS configuration is correct, no further changes should be needed to activate Unified Messaging services.

If the IP gateway in use doesn't support the use of DNS names to resolve the Unified Messaging services, additional configuration steps will be necessary to manually point the IP gateway to the IP addresses of the Unified Messaging services in the second datacenter.

Activating Edge Transport Servers

The steps to activate the Edge Transport server role will vary, depending on the specific configuration. Edge Transport servers in two datacenters can be configured in either an active/passive or an active/active configuration. In an active/passive configuration, the Edge Transport server in the second datacenter is idle until the second datacenter is activated. In an active/active

configuration, Edge Transport servers in both datacenters are delivering mail at all times.

In an active/active configuration, no steps are necessary to activate the second datacenter's Edge Transport servers because they are already running. In an active/passive configuration, the DNS MX resource record for each SMTP domain needs to be updated as part of the switchover from the primary datacenter to the standby datacenter. Although the active/active configuration provides a simple datacenter switchover solution, it has the drawback of requiring careful load monitoring to make sure that after the datacenter switchover, the Edge Transport servers in the second datacenter can provide sufficient capacity to support the increased load now flowing through it, as a result of the Edge Transport servers in the primary datacenter being unavailable.

Even with an active/active configuration, it may be appropriate to update the MX resource records for your Edge Transport servers during a datacenter switchover. Allowing the MX resource record for the failed datacenter to continue to point at the failed datacenter means that when the datacenter starts recovering, it could start experiencing connection attempts to its Edge Transport servers. This could happen while the Edge Transport services are in an unstable state (for example, because dependent services in the datacenter are being restored).

Assuming the DNS records are under the control of the organization, activating Edge Transport servers involves updating the MX resource record for each SMTP domain hosted by the server.

Note:

If the MX resource record used by your organization isn't hosted by a DNS server under your organization's control, you might consider referencing a CNAME record in the MX resource record and using a CNAME record under the organization's control that can then be updated.

DNS updates enable incoming traffic, and outgoing traffic is handled by the activation of the mailbox databases in a site that has functioning Edge Transport servers:

- When incoming SMTP connections are initiated using the updated name resolution information, SMTP clients will connect to the Edge Transport servers in the second datacenter. Traffic will be appropriately routed by the Edge Transport server, and no further changes are required.
- When outgoing SMTP connections are initiated, they will try the locally available Edge Transport server, and those messages will be queued or immediately sent based on the status of the receiving server.

Return to top

Restoring Service to the Primary Datacenter

Generally, datacenter failures are either temporary or permanent. With a permanent failure, such as an event that has caused the permanent destruction of a primary datacenter, there's no expectation that the primary datacenter will be activated. However, with a temporary failure (for example, an extended power loss or extensive but repairable damage), there's an expectation that the primary datacenter will eventually be restored to full service.

The process of restoring service to a previously failed datacenter is referred to as a *switchback*. The

steps used to perform a datacenter switchback are similar to the steps used to perform a datacenter switchover. A significant distinction is that datacenter switchbacks are scheduled, and the duration of the outage is often much shorter.

It's important that switchback not be performed until the infrastructure dependencies for Exchange have been reactivated, are functioning and stable, and have been validated. If these dependencies aren't available or healthy, it's likely that the switchback process will cause a longer than necessary outage, and the process could fail altogether.

Mailbox Server Role Switchback

The Mailbox server role should be the first role that's switched back to the primary datacenter. The following steps detail the Mailbox server role switchback process:

1. As part of the datacenter switchover process, the Mailbox servers in the primary datacenter were put into a stopped state. When the environment (such as primary datacenter, Exchange dependencies, and wide area network (WAN) connectivity) is ready, the first step is to put the Mailbox servers in the restored primary datacenter into a started state and incorporate them into the DAG. The way in which this is done depends on whether the DAG is in DAC mode.
   a. If the DAG is in DAC mode, you can reincorporate the DAG members in the primary site by using the Start-DatabaseAvailabilityGroup cmdlet. Then, to make sure that the proper quorum model is being used by the DAG, run the Set-DatabaseAvailabilityGroup cmdlet against the DAG without specifying any parameters.
   b. If the DAG isn't in DAC mode, you can reincorporate the DAG members by using the Add-DatabaseAvailabilityGroupServer cmdlet.
2. After the Mailbox servers in the primary datacenter have been incorporated into the DAG, they will need some time to synchronize their database copies. Depending on the nature of the failure, the length of the outage, and actions taken by an administrator during the outage, this may require reseeding the database copies. For example, if during the outage, you remove the database copies from the failed primary datacenter to allow log file truncation to occur for the surviving active copies in the second datacenter, reseeding will be required. Each database can individually proceed from this point forward. After a replicated database copy in the primary datacenter is healthy, it can proceed to the next step.

Note:

This process doesn't require that all databases be moved at the same time. You are encouraged to move the majority of your organization's databases at one time, but some databases many linger in the second datacenter if there are issues associated with the database copies in the primary datacenter.

3. After a majority of the databases are in a healthy state in the primary datacenter, the switchback outage can be scheduled. When the scheduled time arrives, the following steps must be taken:
   a. During the datacenter switchover process, the DAG was configured to use an alternate witness server. The DAG must be reconfigured to use a witness server in the primary datacenter. If you are using the same witness server and witness directory that was used prior to the primary datacenter outage, you can run the `Set-DatabaseAvailabilityGroup -Identity DAGName`

command. If you plan on using a witness server or witness directory that is different from the original witness server and directory, use the Set-DatabaseAvailabilityGroup command to configure the witness server and witness directory parameters with the appropriate values.

b. The databases being reactivated in the primary datacenter should be dismounted in the second datacenter. You can use the Dismount-Database cmdlet to dismount the databases.

c. After the databases have been dismounted, the Client Access server URLs should be moved from the second datacenter to the primary datacenter. This is accomplished by changing the DNS record for the URLs to point to the Client Access server or array in the primary datacenter. This will result in the system acting as though a database failover has occurred for each database being moved.

Important:
Don't proceed to the next step until the Client Access server URLs have been moved and the DNS TTL and cache entries have expired. Activating the databases in the primary datacenter prior to moving the Client Access server URLs to the primary datacenter will result in an invalid configuration (for example, a mounted database that has no Client Access servers in its Active Directory site).

d. Because each database in the primary datacenter is in a healthy state, it can be activated in the primary datacenter by performing database switchovers. This is accomplished by using the Move-ActiveMailboxDatabase cmdlet for each database that will be activated.

e. After each database is moved to the primary datacenter, it can be mounted by using the Mount-Database cmdlet.

After one or more databases are active and mounted in the primary datacenter, switchback procedures for the other server roles can be performed.

Client Access Server Switchback

As part of the switchover process, the internal and external DNS records used by clients, other servers, and IP gateways to resolve the service endpoints for Client Access servers, Transport and Unified Messaging services, and Edge Transport servers were modified to point to the corresponding endpoints in the second datacenter. The switchback process for the other server roles involves modifying those records to point to the restored service endpoints in the primary datacenter.

As with the DNS changes that were made during the switchover to the second datacenter, clients, servers, and IP gateways will continue to try to connect, and should automatically connect after the TTL has expired for the original DNS entry, and after the entry is expired from their DNS cache.

Return to top

Reestablishing Site Resilience

After switchback to the primary datacenter is completed successfully, you can reestablish site resilience for the primary datacenter by verifying the health and status of each mailbox database

copy in the second datacenter. In addition, if any database copies in the second datacenter were originally blocked for activation, you can reconfigure those settings at this time.

Return to top

Perform a Server Switchover

Applies to: *Exchange Server 2013 SP1, Exchange Server 2013*

Topic Last Modified: *2014-06-23*

A server switchover is a task that you perform to move all active mailbox database copies from their current Mailbox server to one or more other Mailbox servers in a database availability group (DAG). This task is performed as part of preparation for a scheduled outage for the current Mailbox server.

What do you need to know before you begin?

- Estimated time to complete: 30 seconds per database
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the EAC to perform a server switchover

You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the"Mailbox database copies" entry in the High availability and site resilience permissions topic.

1. In the EAC, go to **Servers** > **servers**.
2. Select the Mailbox server you want to switchover.
3. In the details pane, select **Server Switchover**.
4. On the **Server Switchover** page, do one of the following:
   a. Accept the default setting of **Automatically choose a target server** (in which case, the system automatically selects the best Mailbox server for each database being switched over), and then click **save**.
   b. Click **Use the specified server as the target for switchover**, click **Browse** to select a Mailbox

server, and then click **save**.

5. When the switchover has completed, click **close** to exit the **Server Switchover** page.

Use the Shell to perform a server switchover

This example performs a server switchover for the server MBX1. The system automatically selects the best Mailbox server for each active database on MBX1.

```
Move-ActiveMailboxDatabase -Server MBX1
```

This example performs a server switchover of the Mailbox server MBX4. When the command completes, MBX5 hosts the active copy of the databases that were previously active on MBX4.

```
Move-ActiveMailboxDatabase -Server MBX4 -ActivateOnServer
MBX5
```

For detailed syntax and parameter information, see Move-ActiveMailboxDatabase.

Backup, restore, and disaster recovery

Exchange Server 2013 > High availability and site resilience >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-20*

As part of your data protection planning, it's important that you understand the ways in which data can be protected, and to determine which method best suits your organization's needs. Data protection planning is a complex process that relies on many decisions that you make during the planning phase of your deployment.

Traditionally, backups have been used for the following scenarios:

- **Disaster recovery** In the event of a hardware or software failure, multiple database copies in a DAG enable high availability with fast failover and little or no data loss. This eliminates downtime and the resulting lost productivity that's a significant cost of recovering from a past point-in-time backup to disk or tape. DAGs can be extended to multiple sites and can provide resilience against disk, server, network, and datacenter failures.
- **Recovery of accidentally deleted items** Historically, in a situation where a user deleted items that later needed to be recovered, it involved finding the backup media on which the data that needed to be recovered was stored, and then somehow obtaining the desired items and providing them to the user. With the new Recoverable Items folder in Exchange 2013 and the Hold Policy that can be applied to it, it's possible to retain all deleted and modified data for a specified period of time, so recovery of these items is easier and faster. This reduces the burden

on Exchange administrators and the IT help desk by enabling end users to recover accidentally deleted items themselves, thereby reducing the complexity and administrative costs associated with single item recovery. For more information, see Messaging policy and compliance and Data loss prevention.

- **Long-term data storage** Backups have also been used as an archive, and typically tape is used to preserve point-in-time snapshots of data for extended periods of time as governed by compliance requirements. The new archiving, multiple-mailbox search, and message retention features in Exchange 2013 provide a mechanism to efficiently preserve data in an end-user accessible manner for extended periods of time. This eliminates expensive restores from tape, and increases productivity. For more information, see In-Place Archiving, In-Place eDiscovery, and In-Place Hold.
- **Point-in-time database snapshot** If a past point-in-time copy of mailbox data is a requirement for your organization, Exchange provides the ability to create a lagged database copy in a DAG environment. This can be useful in the rare event that store logical corruption replicates to multiple database copies in the DAG, resulting in a need to return to a previous point in time. It may also be useful if an administrator accidentally deletes mailboxes or user data. Recovery from a lagged copy can be faster than restoring from a backup because lagged copies don't require a time-consuming copy process from the backup server to the Exchange server. This can significantly lower total cost of ownership by reducing downtime.

Because there are native Exchange 2013 features that meet each of these scenarios in an efficient and cost effective manner, you may be able to reduce or eliminate the use of traditional backups in your environment.

Exchange Native Data Protection

Supported Backup Technologies

Exchange 2013 VSS Writer

Server Recovery

Unified Contact Store Recovery

Recovery Database

Database Portability

Dial Tone Portability

Exchange Native Data Protection

Microsoft's preferred architecture for Exchange Server 2013 leverages a concept known as Exchange Native Data Protection. Exchange Native Data Protection relies on built-in Exchange features to protect your mailbox data, without the use of backups (although you can still use those features and make backups). Exchange 2013 includes several new features and core changes that, when deployed and configured correctly, can provide native data protection that eliminates the need to make traditional backups of your data. Using the high availability features built into

Exchange 2013 to minimize downtime and data loss in the event of a disaster can also reduce the total cost of ownership of the messaging system. By combining these features with other built-in features, such as Legal Hold, you can reduce or eliminate your use of traditional point-in-time backups and reduce the associated costs.

In addition to determining whether Exchange 2013 enables you to move away from traditional point-in-time backups, we recommend that you evaluate the cost of your current backup infrastructure. Consider the cost of end-user downtime and data loss when attempting to recover from a disaster using your existing backup infrastructure. Also, include hardware, installation, and license costs, as well as the management cost associated with recovering data and maintaining the backups. Depending on the requirements of your organization, it's quite likely that a pure Exchange 2013 environment with at least three mailbox database copies will provide lower total cost of ownership than one with backups.

There are several issues that you should consider before using the features built into Exchange 2013 as a replacement for traditional backups. There may also be considerations unique to your organization. Consider the following issues, and note that this isn't an exhaustive list:

- You should determine how many copies of the database need to be deployed. We strongly recommend deploying a minimum of three (non-lagged) copies of a mailbox database before eliminating traditional forms of protection for the database, such as Redundant Array of Independent Disks (RAID) or traditional VSS-based backups.
- You should clearly define the recovery time objective and recovery point objective goals, and you should establish that using a combined set of built-in features in lieu of traditional backups to enable you to meet these goals.
- You should determine how many copies of each database are needed to cover the various failure scenarios against which your system is designed to protect.
- You should determine whether eliminating the use of a DAG or some of its members captures sufficient costs to support a traditional backup solution. If so, you should determine whether that solution improves your recovery time objective or recovery point objective service level agreements (SLAs).
- You should determine whether you can afford to lose a point-in-time copy if the DAG member hosting the copy experiences a failure that affects the copy or the integrity of the copy.
- Exchange 2013 allows you to deploy much larger mailboxes, with a recommended maximum mailbox database size of 2 terabytes (when two or more highly available mailbox database copies are being used). Based on the larger mailboxes that most organizations are likely to deploy, you should determine your recovery point objective if you have to replay a large number of log files when activating a database copy or a lagged database copy.
- You should determine how you'll detect and prevent logical corruption in an active database copy from replicating to the passive copies of the database. This includes determining the recovery plan for this situation and how frequently this scenario has occurred in the past. If logical corruption occurs frequently in your organization, we recommend that you factor that scenario into your design by using one or more lagged copies, with a sufficient replay lag window to allow you to detect and act on logical corruption when it occurs, but before that corruption is

replicated to other database copies.

One of the functions performed at the end of a successful full or incremental backup is the truncation of transaction log files that are no longer needed for database recovery. If backups aren't being taken, log truncation won't occur. To prevent a buildup of log files, you enable circular logging for your replicated databases. When you combine circular logging with continuous replication, you have a new type of circular logging called continuous replication circular logging (CRCL), which is different from Extensible Storage Engine (ESE) circular logging. Whereas ESE circular logging is performed and managed by the Microsoft Exchange Information Store service, CRCL is performed and managed by the Microsoft Exchange Replication service. When enabled, ESE circular logging doesn't generate additional log files and instead overwrites the current log file when needed. However, in a continuous replication environment, log files are needed for log shipping and replay. As a result, when you enable CRCL, the current log file isn't overwritten and closed log files are generated for the log shipping and replay process.

Specifically, the Microsoft Exchange Replication service manages CRCL so that log continuity is maintained and logs aren't deleted if they're still needed for replication. The Microsoft Exchange Replication service and the Microsoft Exchange Information Store service communicate by using remote procedure calls (RPCs) regarding which log files can be deleted.

For truncation to occur on highly available (non-lagged) mailbox database copies, the following must be true:

- The log file has been backed up, or CRCL is enabled.
- The log file is below the checkpoint.
- The other non-lagged copies of the database agree with deletion.
- The log file has been inspected by all lagged copies of the database.

For truncation to occur on lagged database copies, the following must be true:

- The log file is below the checkpoint.
- The log file is older than ReplayLagTime + TruncationLagTime.
- The log file is deleted on the active copy of the database.

Return to top

Supported Backup Technologies

Exchange 2013 supports only Exchange-aware, VSS-based backups. Exchange 2013 includes a plug-in for Windows Server Backup that enables you to make and restore VSS-based backups of Exchange data. To back up and restore Exchange 2013, you must use an Exchange-aware application that supports the VSS writer for Exchange 2013, such as Windows Server Backup (with the VSS plug-in), Microsoft System Center 2012 - Data Protection Manager, or a third-party Exchange-aware VSS-based application.

For detailed steps about how to back up and restore Exchange data using Windows Server Backup, see Using Windows Server Backup to back up and restore Exchange data.

Return to top

Exchange 2013 VSS Writer

Exchange 2013 introduces a significant change from the VSS writer architecture used in Exchange 2010 and Exchange 2007. These earlier versions of Exchange included two VSS writers: one inside the Microsoft Exchange Information Store service (store.exe) and one inside the Microsoft Exchange Replication service (msexchangerepl.exe). In Exchange, the VSS writer functionality previously found in the Microsoft Exchange Information Store service has been moved to the Microsoft Exchange Replication service. The new writer, which is named Microsoft Exchange Writer, is now used by Exchange-aware VSS-based applications to back up active and passive database copies, and to restore backed up database copies. Although the new writer runs in the Microsoft Exchange Replication service, it requires the Microsoft Exchange Information Store service to be running for the writer to be advertised. As a result, both services are required to back up or restore Exchange databases.

Return to top

Exchange Server Recovery

Almost all of the configuration settings for Mailbox and Client Access servers are stored in Active Directory. As with previous versions of Exchange, Exchange 2013 includes a Setup parameter for recovering lost servers. This parameter, */m:RecoverServer*, is used to rebuild and re-create a lost server by using the settings and configuration information stored in Active Directory. However, be aware that there are several settings which are not restored, such as changes to local web.config and other configuration files. In addition, custom registry entries are not restored. We recommend that you use a reliable change management process to track and recreate these changes.

For detailed steps about how to perform a server recovery of a lost Exchange 2013 server, see Recover an Exchange Server. For detailed steps about how to recover a lost server that's a member of a database availability group (DAG), see Recover a database availability group member server.

Return to top

Unified Contact Store Recovery

When Microsoft Lync Server 2013 is used in an Exchange 2013 environment, the user's Lync contact information is stored in a special contact folder in the user's mailbox. This is referred to as the unified contact store (UCS). If you restore a UCS-migrated mailbox, the instant messaging contact list for the target user may be affected. If the user was migrated after the last backup, restoring the mailbox will result in a complete loss of the user's contact list. In less severe cases, modifications to the contact list made by the user since the last backup will be lost. To mitigate this potential data loss, ensure the user is migrated back to the instant messaging server prior to restoring the mailbox.

Return to top

Recovery Database

A recovery database is a special kind of mailbox database that allows you to mount a restored mailbox database and extract data from the restored database as part of a recovery operation. You can use the New-MailboxRestoreRequest cmdlet to extract data from a recovery database. After extraction, the data can be exported to a folder or merged into an existing mailbox. Recovery databases enable you to recover data from a backup or copy of a database without disturbing user access to current data.

Using a recovery database for a Mailbox database from any previous version of Exchange isn't supported. In addition, the target mailbox used for data merges and extraction must be in the same Active Directory forest as the database mounted in the recovery database.

For more information, see Recovery databases. For detailed steps about how to create a recovery database, see Create a recovery database. For detailed steps about how to use a recovery database, see Restore data using a recovery database.

Return to top

Database Portability

Database portability is a feature that enables an Exchange 2013 mailbox database to be moved to and mounted on any other Exchange 2013 Mailbox server in the same organization. By using database portability, reliability is improved by removing several error-prone, manual steps from the recovery processes. In addition, database portability reduces the overall recovery times for various failure scenarios.

For more information, see Database portability. For detailed steps to use database portability, see Move a mailbox database using database portability.

Return to top

Dial Tone Portability

Dial tone portability is a feature that provides a limited business continuity solution for failures that affect a mailbox database, a server, or an entire site. Dial tone portability enables a user to have a temporary mailbox for sending and receiving e-mail while the original mailbox is being restored or repaired. The temporary mailbox can be on the same Exchange 2013 Mailbox server or on any other Exchange 2013 Mailbox server in your organization. This allows an alternative server to host the mailboxes of users who were previously on a server that's no longer available. Clients that support Autodiscover, such as Microsoft Outlook, are automatically redirected to the new server without having to manually update the user's desktop profile. After the user's original mailbox data

has been restored, an administrator can merge a user's recovered mailbox and the user's dial tone mailbox into a single, up-to-date mailbox.

The process for using dial tone portability is called a *dial tone recovery*. A dial tone recovery involves creating an empty database on a Mailbox server to replace a failed database. This empty database, referred to as a *dial tone database*, allows users to send and receive e-mail while the failed database is recovered. After the failed database is recovered, the dial done database and the recovered database are swapped, and then the data from the dial tone database is merged into the recovered database.

For more information, see Dial tone portability. For detailed steps to perform a dial tone recovery, see Perform a dial tone recovery.

Return to top

Using Windows Server Backup to back up and restore Exchange data

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-23*

Microsoft's preferred architecture for Exchange Server 2013 leverages a concept known as Exchange Native Data Protection. Exchange Native Data Protection relies on native Exchange features to protect your mailbox data, without the use of traditional backups. But if you want to create backups, Exchange includes a plug-in for Windows Server Backup (WSB) that enables you to create Exchange-aware Volume Shadow Copy Service (VSS)-based backups of Exchange data. To take Exchange-aware backups, you must have the WSB feature installed.

The plug-in, WSBExchange.exe, runs as a service named Microsoft Exchange Server Extension for Windows Server Backup (the short name for this service is WSBExchange). This service is automatically installed and configured for manual startup on all Mailbox servers. The plug-in enables WSB to create Exchange-aware VSS backups.

Before using WSB to back up Exchange data, we recommend that you familiarize yourself with the following features and options for the plug-in:

- Backups taken with WSB occur at the volume level, and the only way to perform an application-level backup or restore is to select an entire volume. To back up a database and its log stream, you must back up the entire volume containing the database and logs, not just the individual folders. You can't back up any data without backing up the entire volume containing the data.
- The backup must be run locally on the server being backed up, and you can't use the plug-in to

take remote VSS backups. There is no remote administration of WSB or the plug-in. You can, however, use Remote Desktop Services or Terminal Services to remotely manage backups.
- The backup can be created on a local drive or on a remote network share.
- Only full backups should be taken. Log truncation will occur only after a successful completion of a VSS full backup of a volume or folders containing an Exchange database.
- When restoring data, it's possible to restore only Exchange data. This data can be restored to its original location or to an alternate location. If you restore the data to its original location, WSB and the plug-in automatically handle the recovery process, including dismounting any existing database and replaying logs into the restored database.
- The restore process doesn't support the Exchange recovery database (RDB). If you want to use an RDB, you must restore the data to an alternate location and then manually copy or move the restored data from that location into the RDB folder structure.
- When restoring Exchange data, all backed up databases must be restored together. You can't restore a single database.
- Bare metal restores are supported when using WSB; however, the recommended recovery approach for Exchange servers is to recover the Exchange server and then restore the data. If you are using a third-party backup application (e.g., non-Microsoft), then support for bare metal restores of Exchange may be available from your backup application vendor.

The following table describes the supportability of the backup and recovery options available for Exchange 2013 with WSB.

If you…	Then…
Back up the full server…	A VSS copy backup will be performed, and the transaction logs for the databases on the server will not be truncated.
Perform a custom backup and select one or more volumes to back up…	A VSS full backup can be selected, allowing the transaction logs for the databases on the selected volumes to be truncated at the completion of a successful backup.
Perform a custom backup and select one or more folders to back up…	A VSS full backup can be selected and the log files will be truncated; however, restoration of the backup will be limited to file restore, as an Application level restore will not be available as an option.

For detailed steps to back up Exchange using WSB, see Use Windows Server Backup to back up Exchange.

For detailed steps to restore data from a backup taken with WSB, see Use Windows Server Backup

to restore a backup of Exchange.

Use Windows Server Backup to back up Exchange

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-18*

You can use Windows Server Backup to back up and restore Exchange databases. Exchange includes a plug-in for Windows Server Backup that allows you to make Volume Shadow Copy Service (VSS)-based backups of Exchange data.

What do you need to know before you begin?

- Estimated time to complete: 1 minute, plus the time it takes to back up the data
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox recovery" entry in the Recipients Permissions topic.
- The Windows Server Backup feature must be installed on the local computer.
- During the backup operation, a consistency check of the Exchange data files is run to make sure that the files are in a good state and can be used for recovery. If the consistency check succeeds, Exchange data is available for recovery from that backup. If the consistency check fails, the Exchange data isn't available for recovery. Windows Server Backup runs the consistency check on the snapshot taken for the backup. As a result, before copying files from the snapshot to backup media, the consistency of the backup is known, and the user is notified of the consistency check results.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use Windows Server Backup to back up Exchange

1. Start Windows Server Backup.
2. Select **Local Backup**.
3. In the Actions pane, click **Backup Once...** to start the Backup Once Wizard.
4. On the **Backup Options** page, select **Different options**, and then click **Next**.
5. On the **Select Backup Configuration** page, select **Custom**, and then click **Next**.

6. On the **Select Items for Backup** page, click **Add Items** to select the volume(s) to be backed up, and then click **OK**.

Note:
Choose volumes and not individual folders. The only way to perform an application-level backup or restore is to select an entire volume.

7. Click **Advanced Settings**. On the **Exclusions** tab, click **Add Exclusion** to add any files or file types you want to exclude from the backup.

Note:
By default, volumes that contain operating system components or applications are included in the backup and can't be excluded.

8. On the **VSS Settings tab**, select **VSS full Backup**, and then click **OK**, and then click **Next**.
9. On the **Specify Destination Type** page, select the location where you want to store the backup, and then click **Next**.
    - If you choose **Local drives**, the **Select Backup Destination** page appears. Select an option from the **Backup destination** dropdown, and then click **Next**.
    - If you choose **Remote shared folder**, the **Specify remote folder** page appears. Specify a UNC path for the backup files, configure access control settings. Choose **Do not inherit** if you want the backup to be accessible only through a specific account. Provide a user name and password for an account that has write permissions on the computer hosting the remote folder, and then click **OK**. Alternatively, choose **Inherit** if you want the backup to be accessible by everyone who has access to the remote folder. Click **Next**.
10. On the **Confirmation** page, review the backup settings, and then click **Backup**.
11. On the **Backup Progress** page, you can view the status and progress of the backup operation.
12. Click **Close** to exit the **Backup Progress** page at any time. Any backup in progress will continue to run in the background.

How do you know this worked?

To verify that you've successfully backed up the data, do any of the following:

- On the server on which Windows Server Backup was run, the last backup status will be displayed, which should say Successful. You can also verify that the backup completed successfully by viewing the Windows Server Backup logs.
- Open Event Viewer and verify that a backup completion event was logged in the Application event log.
- Run the following command in the Exchange Management Shell to verify that each database on the selected volume(s) was backed up successfully:

```
Get-MailboxDatabase -Server <ServerName> -Status | fl
Name,*FullBackup
```

The *SnapshotLastFullBackup* and *LastFullBackup* properties of the database indicate when the last successful backup was taken, and if it was a VSS full backup.

Use Windows Server Backup to restore a backup of Exchange

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-18*

You can use Windows Server Backup to back up and restore Exchange databases. Exchange includes a plug-in for Windows Server Backup that allows you to make and restore Volume Shadow Copy Service (VSS)-based backups of Exchange data. For additional information, see Using Windows Server Backup to back up and restore Exchange data.

What do you need to know before you begin?

- Estimated time to complete: 1 minute, plus the time it takes to restore the data
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox recovery" entry in the Recipients Permissions topic.
- The Windows Server Backup feature must be installed on the local computer.
- When restoring a database to its original location, the database can remain in a dirty shutdown state and be mountable by the system. When restoring to an alternate location (for example, in preparation to use a recovery database), the database must be manually brought into a clean shutdown state by using Exchange Server Database Utilities (Eseutil.exe).

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use Windows Server Backup to restore a backup of Exchange

1. Start Windows Server Backup.
2. Select **Local Backup**.
3. In the Actions pane, click **Recover...** to start the Recovery Wizard.
4. On the **Getting Started** page, do either of the following:
   - If the data being recovered was backed up on the local server, select **This server (ServerName)**, and then click **Next**.
   - If the data being recovered is from another server, or if the backup being recovered is located

on another computer, select **Another server**, and then click **Next**. On the **Specify location type** page, select **Local drives** or **Remote shared folder**, and then click **Next**. If you select **Local drives**, select the drive containing the backup on the **Select backup location** page, and then click **Next**. If you select **Remote shared folder**, enter the UNC path for the backup data on the **Specify remote folder** page, and then click **Next**.

5. On the **Select Backup Date** page, select the date and time of the backup that you want to recover, and then click **Next**.
6. On the **Select Recovery Type** page, select **Applications**, and then click **Next**.

Note:

If **Applications** is not available as a selection, it indicates that the backup selected for restore was a folder-level backup, and not a volume level backup. You must perform backups at the volume level when backing up Exchange data with Windows Server Backup.

7. On the **Select Application** page, verify that Exchange is selected in the **Applications** field. Click **View Details** to view the application components of the backups. If the backup that you're recovering is the most recent, the **Do not perform a roll-forward recovery of the application database** check box is displayed. Select this check box if you want to prevent Windows Server Backup from rolling forward the database being recovered by committing all uncommitted transaction logs. Click **Next**.
8. On the **Specify Recovery Options** page, specify where you want to recover the data, and then click **Next**:
   - Choose **Recover to original location** if you want to restore the Exchange data directly to its original location. If you use this option, you can't choose which databases are restored; all backed up databases on the volume will be restored to their original locations.
   - Choose **Recover to another location** if you want to restore individual databases and their files to a specified location. Click **Browse** to specify the alternate location. If you use this option, you can choose which databases are restored. After being restored, the data files can then be moved into a recovery database, manually moved back to their original location, or mounted somewhere else in the Exchange organization using Database portability. When you restore a database to an alternate location, the restored database will be in a dirty shutdown state. After the restore process has completed, you will need to manually put the database into a clean shutdown state using Eseutil.exe.
9. On the **Confirmation** page, review the recovery settings, and then click **Recover**.
10. On the **Recovery Progress** page, you can view the status and progress of the recovery operation.
11. Click **Close** when the recovery operation has completed.

How do you know this worked?

The **Recovery Progress** page will indicate whether or not the recovery process completed successfully. To further verify that you've successfully restored the data, do any of the following:

- Examine the target directory of the backup and verify that the restored data exists.
- On the server on which Windows Server Backup was run, verify that the job completed successfully by viewing the backup logs.

- Open Event Viewer and verify that a restore completion event was logged in the Application event log.

Recover an Exchange Server

Exchange Server 2013 > High availability and site resilience > Backup, restore, and disaster recovery >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-12*

You can recover a lost server by using the **Setup /m:RecoverServer** switch in Microsoft Exchange Server 2013. Most of the settings for a computer running Exchange 2013 are stored in Active Directory. The */m:RecoverServer* switch rebuilds an Exchange server with the same name by using the settings and other information stored in Active Directory.

Recovering a lost Exchange server is often accomplished by using new hardware. However, you can also use an existing server.

This topic shows you how to recover a lost Exchange 2013 server that isn't a member of a database availability group (DAG). For detailed steps about how to recover a server that was a member of a DAG, see Recover a database availability group member server.

Note:

If Exchange is installed in a location other than the default location, you must use the */TargetDir* switch to specify the location of the Exchange binary files. If you don't use the */TargetDir* switch, the Exchange files are installed in the default location (%programfiles%\Microsoft\Exchange Server\V15).
To determine the install location, follow these steps:

1. Open ADSIEDIT.MSC or LDP.EXE.
2. Navigate to the following location: **CN=ExServerName,CN=Servers,CN=First Administrative Group,CN=Administrative Groups,CN=ExOrg Name,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=DomainName,CN=Com**
3. Right-click the Exchange server object, and then click **Properties**.
4. Locate the **msExchInstallPath** attribute. This attribute stores the current installation path.

Looking for other management tasks related to backing up and restoring data? Check out Backup, restore, and disaster recovery.

What do you need to know before you begin?

- Estimated time to complete: 20 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange infrastructure permissions" section in the

Exchange and Shell infrastructure permissions topic.

- The server on which recovery is being performed must be running the same operating system as the lost server. For example, you can't recover a server that was running Exchange 2013 and Windows Server 2008 R2 on a server running Windows Server 2012, or vice versa. Likewise, you can’t recover a server that was running Exchange 2013 and Windows Server 2012 on a server running Windows Server 2012 R2, or vice versa.
- The same disk drive letters on the failed server for mounted databases must exist on the server on which you're running recovery.
- The server on which recovery is being performed should have the same performance characteristics and hardware configuration as the lost server.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Recover a Lost Exchange Server

1. Reset the computer account for the lost server. For detailed steps, see Reset a Computer Account.
2. Install the proper operating system and name the new server with the same name as the lost server. Recovery won't succeed if the server on which recovery is being performed doesn't have the same name as the lost server.
3. Join the server to the same domain as the lost server.
4. Install the necessary prerequisites and operating system components. For details, see Exchange 2013 system requirements and Exchange 2013 prerequisites.
5. Log on to the server being recovered and open a command prompt.
6. Navigate to the Exchange 2013 installation files, and run the following command.

```
Setup /m:RecoverServer /IAcceptExchangeServerLicenseTerms
```

7. After Setup has completed, but before the recovered server is put into production, reconfigure any custom settings that were previously present on the server, and then restart the server.

How do you know this worked?

The successful completion of Setup will be the primary indicator that the recovery was successful. To further verify that you've successfully recovered a lost server, do the following:

- Open the Windows Services tool (services.msc) and verify that the Microsoft Exchange services have been installed and are running.

Recover a database availability group

member server

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-16*

If a Mailbox server that's a member of a database availability group (DAG) is lost or otherwise fails and is unrecoverable and needs replacement, you can perform a server recovery operation. Microsoft Exchange Server 2013 Setup includes the switch */m:RecoverServer* that can be used to perform the server recovery operation. Running Setup with the */m:RecoverServer* switch causes Setup to read the server's configuration information from Active Directory for a server with the same name as the server from which you're running Setup. After the server's configuration information is gathered from Active Directory, the original Exchange files and services are then installed on the server, and the roles and settings that were stored in Active Directory are then applied to the server.

Looking for other management tasks related to DAGs? Check out Managing database availability groups.

What do you need to know before you begin?

- Estimated time to complete: 30 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox database copies" entry in the High availability and site resilience permissions topic.
- If Exchange is installed in a location other than the default location, you must use the */TargetDir* Setup switch to specify the location of the Exchange program files. If you don't use the */TargetDir* switch, the Exchange program files will be installed in the default location (%programfiles%\Microsoft\Exchange Server\V15).

To determine the install location, follow these steps:

1. Open ADSIEDIT.MSC or LDP.EXE.
2. Navigate to the following location: **CN=ExServerName,CN=Servers,CN=First Administrative Group,CN=Administrative Groups,CN=ExOrg Name,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=DomainName,CN=Com**
3. Right-click the Exchange server object, and then click **Properties**.
4. Locate the **msExchInstallPath** attribute. This attribute stores the current installation path.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use Setup /m:RecoverServer to recover a server

1. Retrieve any replay lag or truncation lag settings for any mailbox database copies that exist on the server being recovered by using the Get-MailboxDatabase cmdlet:

```
Get-MailboxDatabase DB1 | Format-List *lag*
```

2. Remove any mailbox database copies that exist on the server being recovered by using the Remove-MailboxDatabaseCopy cmdlet:

```
Remove-MailboxDatabaseCopy DB1\MBX1
```

3. Remove the failed server's configuration from the DAG by using the Remove-DatabaseAvailabilityGroupServer cmdlet:

```
Remove-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer MBX1
```

Note:

If the DAG member being removed is offline and can't be brought online, you must add the *ConfigurationOnly* parameter to the preceding command. If you use the *ConfigurationOnly* switch, you must also manually evict the node from the cluster.

4. Reset the server's computer account in Active Directory. For detailed steps, see Reset a Computer Account.
5. Open a Command Prompt window. Using the original Setup media, run the following command:

```
Setup /m:RecoverServer
```

6. When the Setup recovery process is complete, add the recovered server to the DAG by using the Add-DatabaseAvailabilityGroupServer cmdlet:

```
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer MBX1
```

7. After the server has been added back to the DAG, you can reconfigure mailbox database copies by using the Add-MailboxDatabaseCopy cmdlet. If any of the database copies being added previously had replay lag or truncation lag times greater than 0, you can use the *ReplayLagTime* and *TruncationLagTime* parameters of the Add-MailboxDatabaseCopy cmdlet to reconfigure those settings:

```
Add-MailboxDatabaseCopy -Identity DB1 -MailboxServer MBX1
Add-MailboxDatabaseCopy -Identity DB2 -MailboxServer MBX1 -
ReplayLagTime 3.00:00:00
Add-MailboxDatabaseCopy -Identity DB3 -MailboxServer MBX1 -
ReplayLagTime 3.00:00:00 -TruncationLagTime 3.00:00:00
```

How do you know this worked?

To verify that you've successfully recovered the DAG member, do the following:

- In the Shell, run the following command to verify the health and status of the recovered DAG member.

```
Test-ReplicationHealth <ServerName>
```

```
Get-MailboxDatabaseCopyStatus -Server <ServerName>
```

All of the replication health tests should pass successfully, and the status of databases and their content indexes should be healthy.

Recovery databases

Exchange Server 2013 > High availability and site resilience > Backup, restore, and disaster recovery >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-02*

A recovery database (RDB) is a special kind of mailbox database that allows you to mount a restored mailbox database and extract data from the restored database as part of a recovery operation. You can use the New-MailboxRestoreRequest cmdlet to extract data from an RDB. After extraction, the data can be exported to a folder or merged into an existing mailbox. RDBs enable you to recover data from a backup or copy of a database without disturbing user access to current data.

Microsoft Exchange Server 2013 supports the ability to restore data directly to a recovery database. Mounting the recovered data as a recovery database allows the administrator to restore individual mailboxes or individual items in a mailbox. Restoring to a recovery database can be accomplished in two ways:

- If a recovery database already exists, the application can dismount the database, restore the data onto the recovery database and log files, and then remount the database.
- The database and log files can be restored to any disk location. Exchange analyzes the restored data and replays the transaction logs to bring the databases up to date, and then a recovery database can be configured to point to already recovered database files.

Difference between a mailbox database and a recovery database

RDBs are different from standard mailbox databases in several respects:

- An RDB is created by using the Exchange Management Shell.
- Mail can't be sent to or from an RDB. All client protocol access to an RDB (including SMTP, POP3, and IMAP4) is blocked. This design prevents using an RDB to insert mail into or remove mail from the messaging system.
- Client MAPI access using Microsoft Office Outlook or Outlook Web App is blocked. MAPI access is supported for an RDB, but only by recovery tools and applications. Both the mailbox GUID and the database GUID must be specified when using MAPI to log into a mailbox in an RDB.
- Mailboxes in an RDB can't be connected to user accounts. To allow a user to access the data in a mailbox in an RDB, the mailbox must be merged into an existing mailbox, or exported to a folder.
- System and mailbox management policies aren't applied. This design prevents items in an RDB from being deleted by the system during the recovery process.
- Online maintenance isn't performed for RDBs.
- Circular logging can't be enabled for RDBs.
- Only one RDB can be mounted at any time on a Mailbox server. The use of an RDB doesn't count against the database limit per Mailbox server.
- You can't create mailbox database copies of an RDB.
- An RDB can be used as a target for restore operations, but not backup operations.
- A recovered database mounted as an RDB isn't tied to the original mailbox in any way.

Using a recovery database

Before you can use an RDB, there are certain requirements that must be met. An RDB can be used for Exchange 2013 mailbox databases only. Mailbox databases from previous versions of Exchange aren't supported. In addition, the target mailbox used for data merges and extraction must be in the same Active Directory forest as the database mounted in the RDB.

An RDB can be used to recover data in several situations, such as:

- **Same server dial tone recovery** You can perform a recovery from an RDB after the original database has been restored from backup, as part of a dial tone recovery operation.
- **Alternate server dial tone recovery** You can use an alternate server to host the dial tone database, and then later recover data from an RDB after the original database has been restored from backup.
- **Mailbox recovery** You can recover an individual mailbox from backup when the deleted mailbox retention period has elapsed. You then extract data from the restored mailbox and copy it to a target folder or merge it with another mailbox.
- **Specific item recovery** You can restore from backup data that has been deleted or purged from a mailbox.

Note:

Folder access control lists (ACLs) aren't preserved when recovering content into an active mailbox. Because the recovery process typically involves recovering mailbox data and merging

the content back into the original database, there should be no need to recover or copy ACLs.

An RDB is designed for mailbox database recovery under the following conditions and scenarios:

- The logical information about the original database and the mailboxes in that database remains intact and unchanged in Active Directory.
- You need to recover a single mailbox or a single database. Recovery scenarios include:
  - Recovering or repairing a database while a dial tone database is in use, with the goal of merging the two databases.
  - Recovering a database on a server other than the original server for that database. If needed, you can then merge the recovered data back to the original server.
  - Recovering deleted items that users previously deleted from their mailbox, after the deleted item retention period has expired.

RDBs are generally not designed for scenarios in which you have to restore entire servers, when you have to restore multiple databases, or when you're in an emergency situation that requires changing or rebuilding your Active Directory topology.

For detailed steps about how to create an RDB, see Create a recovery database. For detailed steps about how to use an RDB, see Restore data using a recovery database.

Create a recovery database

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-21*

You can use the Shell to create a recovery database, a special kind of mailbox database that's used to mount and extract data from the restored database as part of a recovery operation. After you create a recovery database, you can move a recovered or restored mailbox database into the recovery database, and then use the New-MailboxRestoreRequest cmdlet to extract data from the recovered database. After extraction, the data can then be exported to a folder or merged into an existing mailbox. Using recovery databases, you can recover data from a backup or copy of a database without disrupting user access to current data.

Looking for other management tasks related to recovery databases? Check out Recovery databases.

What do you need to know before you begin?

- Estimated time to complete this task: 1 minute
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox recovery" entry in the Recipients Permissions

topic.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to create a recovery database

This example creates the recovery database RDB1 on the Mailbox server MBX2.

```
New-MailboxDatabase -Recovery -Name RDB1 -Server MBX2
```

This example creates the recovery database RDB2 on the Mailbox server MBX1 using a custom path for the database file and log folder.

```
New-MailboxDatabase -Recovery -Name RDB2 -Server MBX1 -
EdbFilePath "C:\Recovery\RDB2\RDB2.EDB" -LogFolderPath "C:
\Recovery\RDB2"
```

For detailed syntax and parameter information, see New-MailboxDatabase.

How do you know this worked?

To verify that you've successfully created a recovery database, do the following:

- In the Shell, run the following command to display configuration information for the recovery database.

```
Get-MailboxDatabase <RecoveryDatabaseName> | Format-List
```

Other Tasks

After you create a recovery database, you may also want to restore data using a recovery database. For detailed steps, see Restore data using a recovery database.

Restore data using a recovery database

High availability and site resilience > Backup, restore, and disaster recovery > Recovery databases >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-30*

A recovery database (RDB) is a special kind of mailbox database that allows you to mount and extract data from a restored mailbox database as part of a recovery operation. RDBs allow you to recover data from a backup or copy of a database without disrupting user access to current data.

After you create an RDB, you can restore a mailbox database into the RDB by using a backup application or by copying a database and its log files into the RDB folder structure. Then you can use the New-MailboxRestoreRequest cmdlet to extract data from the recovered database. Once extracted, the data can then be exported to a folder or merged into an existing mailbox.

For additional management tasks related to RDBs, see Recovery databases.

What do you need to know before you begin?

- Estimated time to complete this task: 1 minute, plus the time it takes to put the database into a clean shutdown state and to extract the data.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox recovery" entry in the Recipients Permissions topic.
- Some backup applications have the ability to restore Exchange data directly to a recovery database. Windows Server Backup can restore only file-level backups to a recovery database. It cannot be used to restore application-level backups to a recovery database.
- The database and log files containing the recovered data must be restored or copied into the RDB folder structure.
- The database must be in a clean shutdown state. Because an RDB is an alternate restore location for all databases, all restored databases will be in a dirty shutdown state. You must use **Eseutil /R** to put restored databases into a clean shutdown state.

Use the Shell to recover data using a recovery database

1. Copy a recovered database and its log files, or restore a database and it log files, to the location you will use for your recovery database.
2. Use Eseutil to bring that database into a clean shutdown state. In the following example, EXX is the log generation prefix for the database (for example, E00, E01, E02, and so on).

```
Eseutil /R EXX /l <RDBLogFilePath> /d <RDBEdbFolder>
```

The following example illustrates a log generation prefix of E01 and a recovery database and log file path of E:\Databases\RDB1:

```
Eseutil /R E01 /l E:\Databases\RDB1 /d E:\Databases\RDB1
```

3. Create a recovery database. Give the recovery database a unique name, but use the name and path of the database file for the EdbFilePath parameter, and the location of the recovered log

files for the LogFolderPath parameter.

```
New-MailboxDatabase -Recovery -Name <RDBName> -Server
<ServerName> -EdbFilePath <RDBPathandFileName> -
LogFolderPath <LogFilePath>
```

The following example illustrates creating a recovery database that will be used to recover DB1.edb and its log files, which are located at E:\Databases\RDB1.

```
New-MailboxDatabase -Recovery -Name <RDBName> -Server
<ServerName> -EdbFilePath "E:\Databases\RDB1\DB1.EDB" -
LogFolderPath "E:\Databases\RDB1"
```

4. Restart the Microsoft Exchange Information Store service:

```
Restart-Service MSExchangeIS
```

5. Mount the recovery database:

```
Mount-database <RDBName>
```

6. Verify that the mounted database contains the mailbox(es) you want to restore:

```
Get-MailboxStatistics -Database <RDBName> | ft -auto
```

7. Use the New-MailboxRestoreRequest cmdlet to restore a mailbox or items from the recovery database to a production mailbox.

The following example restores the source mailbox that has the MailboxGUID 1d20855f-fd54-4681-98e6-e249f7326ddd on mailbox database DB1 to the target mailbox with the alias Scott.

```
New-MailboxRestoreRequest -SourceDatabase DB1 -
SourceStoreMailbox 1d20855f-fd54-4681-98e6-e249f7326ddd -
TargetMailbox Scott
```

The following example restores the content of the source mailbox that has the display name Scott Schnoll on mailbox database DB1 to the archive mailbox for scott@contoso.com.

```
New-MaiboxRestoreRequest -SourceDatabase DB1 -
SourceStoreMailbox "Scott Schnoll" -TargetMailbox
scott@contoso.com -TargetIsArchive
```

8. Periodically check the status of the Mailbox restore request using Get-MailboxRestoreRequest. Once the restore has a status of Completed, remove the restore request using Remove-MailboxRestoreRequest. For example:

```
Get-MailboxRestoreRequest -Status Completed | Remove-
MailboxRestoreRequest
```

How do you know this worked?

To verify that you have successfully recovered the mailbox data, open the target mailbox using Outlook or Outlook Web App and verify that the recovered data is present.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Database portability

Exchange Server 2013 > High availability and site resilience > Backup, restore, and disaster recovery >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-21*

Database portability is a feature that enables a Microsoft Exchange Server 2013 mailbox database to be moved to or mounted on any other Mailbox server in the same organization running Exchange 2013 that has databases with the same database schema version. Mailbox databases from previous versions of Exchange can't be moved to a Mailbox server running Exchange 2013. By using database portability, reliability is improved by removing several error-prone, manual steps from the recovery processes. In addition, database portability reduces the overall recovery times for various failure scenarios.

For information about how to perform a database recovery using the database portability feature, see Move a mailbox database using database portability.

Move a mailbox database using database portability

High availability and site resilience > Backup, restore, and disaster recovery > Database portability >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-16*

You can use database portability to move a Microsoft Exchange Server 2013 mailbox database between Exchange 2013 Mailbox servers in the same organization. This can help reduce overall

recovery times for some failure scenarios. To learn more, see Database portability.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes, plus the time it takes to restore the data, move the database files, and wait for Active Directory replication to complete.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox recovery" entry in the Recipients Permissions topic.
- You can't use the EAC to move user mailboxes to a recovered or dial tone database using database portability.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to move user mailboxes to a recovered or dial tone database using database portability

1. Verify that the database to be moved is in a clean shutdown state. If the database isn't in a clean shutdown state, perform a soft recovery.

Note:

When you perform a soft recovery, any uncommitted log files are committed to the database. If you don't have all of the required log files, you can't complete the soft recovery process. Proceed to step 2.

To commit all uncommitted log files to the database, from a command prompt, run the following command.

```
ESEUTIL /R <Enn>
```

Note:

<E*nn*> specifies the log file prefix for the database into which you intend to replay the log files. The log file prefix specified by <E*nn*> is a required parameter for Eseutil /r.

2. Create a database on a server using the following syntax:

```
New-MailboxDatabase -Name <DatabaseName> -Server
<ServerName> -EdbFilePath <DatabaseFileNameandPath> -
LogFolderPath <LogFilesPath>
```

3. Set the *This database can be over written by restore* attribute using the following syntax:

```
Set-MailboxDatabase <DatabaseName> -AllowFileRestore $true
```

4. Move the original database files (.edb file, log files, and Exchange Search catalog) to the

database folder you specified when you created the new database above.

5. Mount the database using the following syntax:

```
Mount-Database <DatabaseName>
```

6. After the database is mounted, modify the user account settings with the Set-Mailbox cmdlet so that the account points to the mailbox on the new mailbox server. To move all of the users from the old database to the new database, use the following syntax.

```
Get-Mailbox -Database <SourceDatabase> |where
{$_.ObjectClass -NotMatch '(SystemAttendantMailbox|
ExOleDbSystemMailbox)'}| Set-Mailbox -Database
<TargetDatabase>
```

7. Trigger delivery of any messages remaining in queues using the following syntax.

```
Get-Queue <QueueName> | Retry-Queue -Resubmit $true
```

After Active Directory replication is complete, all users can access their mailboxes on the new Exchange server. Most clients are redirected via Autodiscover. Microsoft Office Outlook Web App users are also automatically redirected.

How do you know this worked?

To verify that you've successfully moved a mailbox, do the following:

- Open the mailbox using Outlook Web App.
- Open the mailbox using Microsoft Outlook.

Dial tone portability

Exchange Server 2013 > High availability and site resilience > Backup, restore, and disaster recovery >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-01-28*

Dial tone portability is a feature of Microsoft Exchange Server 2013 that provides a limited business continuity solution for failures that affect a mailbox database, a server, or an entire site. Dial tone portability enables users to have a temporary mailbox for sending and receiving email while their original mailbox is being restored or repaired. The temporary mailbox can be on the same Exchange 2013 Mailbox server or on any other Exchange 2013 Mailbox server in your organization that has databases with the same database schema version. This allows an alternative server to host the mailboxes of users who were previously on a server that is no longer available. Clients that

support Autodiscover are automatically redirected to the new server without having to manually update the user's desktop profile. After the user's original mailbox data has been restored, an administrator can merge a user's recovered mailbox and the user's dial tone mailbox into a single, up-to-date mailbox.

The process for using dial tone portability is called a *dial tone recovery*. A dial tone recovery involves creating an empty database on a Mailbox server to replace a failed database. This empty database, referred to as a *dial tone database*, allows users to send and receive email messages while the failed database is recovered.

There are three options for performing a dial tone recovery:

- **Dial tone recovery on the server with the failed database** If the server hosting the failed database is still functional, we recommend that you perform a dial tone recovery on that server. This means less downtime because you don't need to move database files between servers. In addition, you won't need to reconfigure messaging profiles for clients that don't support Autodiscover.
- **Dial tone recovery using an alternate server for the dial tone database** If a server fails and needs to be rebuilt, the most efficient way to give users basic mail functionality is to create a dial tone database on another server, and use database portability to move the users' mailbox configuration to that new server. Because this process involves moving the dial tone database back to the original (recovered) server, this option adds more time to the overall recovery process. In addition, this process is more complex than performing a dial tone recovery on the original server. When performing this process, the server hosting the dial tone database must have sufficient resources to support the added load of the additional users. In addition, if the users' client doesn't support Autodiscover, their messaging profile will need to be reconfigured to point to the dial tone server.
- **Dial tone recovery using and staying on an alternate server for the dial tone database** This is similar to the preceding option, except that you don't revert back to the original server. We recommend this option for situations in which it isn't possible or feasible to recover the failed server. In this scenario, users typically remain on an alternate server after the recovery operation has completed. When performing this process, the server hosting the dial tone database must have sufficient resources to support the added load of the additional users. In addition, if the users' client doesn't support Autodiscover, their messaging profile will need to be reconfigured to point to the dial tone server.

All three options follow the same basic steps:

1. Create an empty dial tone database to replace the failed database.

This new database will allow users who had mailboxes on the failed database to send and receive new messages. Dial tone portability allows you to point a user to a different database without moving the mailbox. If you created the dial tone database on a different server than the server that housed the failed database, you need to move the mailbox configuration to that new server.

2. Restore the old database.

Use the backup and recovery software you typically use to restore the failed database. If there is no backup of the failed database, recover the failed database using other means if possible. If you're

using the same server for dial tone recovery, you need to restore the database to a recovery database (RDB).

3. Swap the dial tone database with the restored database.

After the failed database is restored, swap it with the dial tone database. This gives the users the ability to send and receive email and access all the data in the restored database. If users were moved to a dial tone database on another server, you need to move the mailbox configuration back to the original server.

4. Merge the databases.

To get the data from the dial tone database into the restored database, you merge the data using the New-MailboxRestoreRequest cmdlet.

For detailed steps about how to perform a dial tone recovery, see Perform a dial tone recovery.

Perform a dial tone recovery

High availability and site resilience > Backup, restore, and disaster recovery > Dial tone portability >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-27*

Using dial tone portability, users can have a temporary mailbox for sending and receiving email while their original mailbox is being restored or repaired. The temporary mailbox can be on the same Exchange 2013 Mailbox server or on any other Exchange 2013 Mailbox server in your organization. The process for using dial tone portability is called a dial tone recovery, which involves creating an empty database on a Mailbox server to replace a failed database. To learn more, see Dial tone portability.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes, plus the time it takes to restore and move the data.
- You must have fewer than the maximum number of databases deployed to create a dial tone database. Exchange 2013 Standard Edition supports a maximum of five databases per server. Exchange 2013 Enterprise Edition supports a maximum of 50 databases per server in RTM and CU1, and 100 databases per server in CU2 and later.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Mailbox recovery" entry in the Recipients Permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to perform a dial tone recovery on a single server

Note:

You can't use the EAC to perform a dial tone recovery on a single server.

1. Make sure that any existing files for the database being recovered are preserved in case they're needed later for further recovery operations.
2. Use the New-MailboxDatabase cmdlet to create a dial tone database, as shown in this example.

```
New-MailboxDatabase -Name DTDB1 -EdbFilePath D:\DialTone
\DTDB1.EDB
```

3. Use the Set-Mailbox cmdlet to rehome the user mailboxes hosted on the database being recovered, as shown in this example.

```
Get-Mailbox -Database DB1 | Set-Mailbox -Database DTDB1
```

4. Use the Mount-Database cmdlet to mount the database so client computers can access the database and send and receive messages, as shown in this example.

```
Mount-Database -Identity DTDB1
```

5. Create a recovery database (RDB) and restore or copy the database and log files containing the data you want to recover into the RDB. For detailed steps, see Create a recovery database.
6. After the data is copied to the RDB, but before mounting the restored database, copy any log files from the failed database to the recovery database log folder so they can be played against the restored database.
7. Mount the RDB, and then use the Dismount-Database cmdlet to dismount it, as shown in this example.

```
Mount-Database -Identity RDB1
Dismount-Database -Identity RDB1
```

8. After the RDB is dismounted, move the current database and log files within the RDB folder to a safe location. This is done in preparation for swapping the recovered database with the dial tone database.
9. Dismount the dial tone database, as shown in this example. Note that your end users will experience an interruption in service when you dismount this database.

```
Dismount-Database -Identity DTDB1
```

10. Move the database and log files from the dial tone database folder into the RDB folder.

11.Move the database and log files from the safe location containing the recovered database into the dial tone database folder, and then mount the database, as shown in this example.

```
Mount-Database -Identity DTDB1
```

This ends the service interruption for your end users. They will be able to access their original production database and send and receive messages.

12.Mount the RDB, as shown in this example.

```
Mount-Database -Identity RDB1
```

13.Use the Get-Mailbox and New-MailboxRestoreRequest cmdlets to export the data from the RDB and import it into the recovered database, as shown in this example. This will import all the messages sent and received using the dial tone database into the production database.

```
$mailboxes = Get-Mailbox -Database DTDB1

$mailboxes | %{ New-MailboxRestoreRequest -
SourceStoreMailbox $_.ExchangeGuid -SourceDatabase RDB1 -
TargetMailbox $_ }
```

14.After the restore operation is complete, you can dismount and remove the RDB, as shown in this example.

```
Dismount-Database -Identity RDB1
Remove-MailboxDatabase -Identity RDB1
```

For detailed syntax and parameter information, see the following topics:

- New-MailboxDatabase
- Get-Mailbox
- Set-Mailbox
- Mount-Database
- Dismount-Database
- Remove-MailboxDatabase

How do you know this worked?

To verify that you've successfully moved a mailbox, do the following:

- Open the mailbox using Microsoft Office Outlook Web App.
- Open the mailbox using Microsoft Outlook.

Managed Availability

Applies to: *Exchange Server 2013 SP1, Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-07*

Ensuring that users have a good email experience has always been the primary objective for messaging system administrators. To help ensure the availability and reliability of your Microsoft Exchange Server 2013 organization, all aspects of the system must be actively monitored, and any detected issues must be resolved quickly. In previous versions of Exchange, monitoring critical system components typically involved using an external application such as Microsoft System Center 2012 Operations Manager to collect data, and to provide recovery action for problems detected as a result of analyzing the collected data. Exchange 2010 and previous versions included health manifests and correlation engines in the form of management packs. These components enabled Operations Manager to make a determination as to whether a particular component was healthy or unhealthy. In addition, Operations Manager also used the diagnostic cmdlet infrastructure built into Exchange 2010 to run synthetic transactions against various aspects of the system.

Exchange 2013 takes a new approach to monitoring and preserving the end user experience natively using a feature called *Managed Availability* that provides built-in monitoring and recovery actions.

Managed Availability

Managed availability, also known as *Active Monitoring* or *Local Active Monitoring*, is the integration of built-in monitoring and recovery actions with the Exchange high availability platform. It's designed to detect and recover from problems as soon as they occur and are discovered by the system. Unlike previous external monitoring solutions and techniques for Exchange, managed availability doesn't try to identify or communicate the root cause of an issue. It's instead focused on recovery aspects that address three key areas of the user experience:

- **Availability** Can users access the service?
- **Latency** How is the experience for users?
- **Errors** Are users able to accomplish what they want?

The server role consolidation and other architectural changes in Exchange 2013 require a new approach to the monitoring methodologies and health model used in previous versions of Exchange. Managed availability is designed to address these changes by providing a native health monitoring and recovery solution. It moves away from monitoring individual separate slices of the system to monitoring the end-to-end user experience, and protecting the end user's experience through recovery-oriented actions.

Managed availability is an internal process that runs on every Exchange 2013 server. It polls and analyzes hundreds of health metrics every second. If something is found to be wrong, most of the time it will be fixed automatically. But there will always be issues that managed availability won't

be able to fix on its own. In those cases, managed availability will escalate the issue to an administrator by means of event logging.

Managed availability implemented in the form of two services:

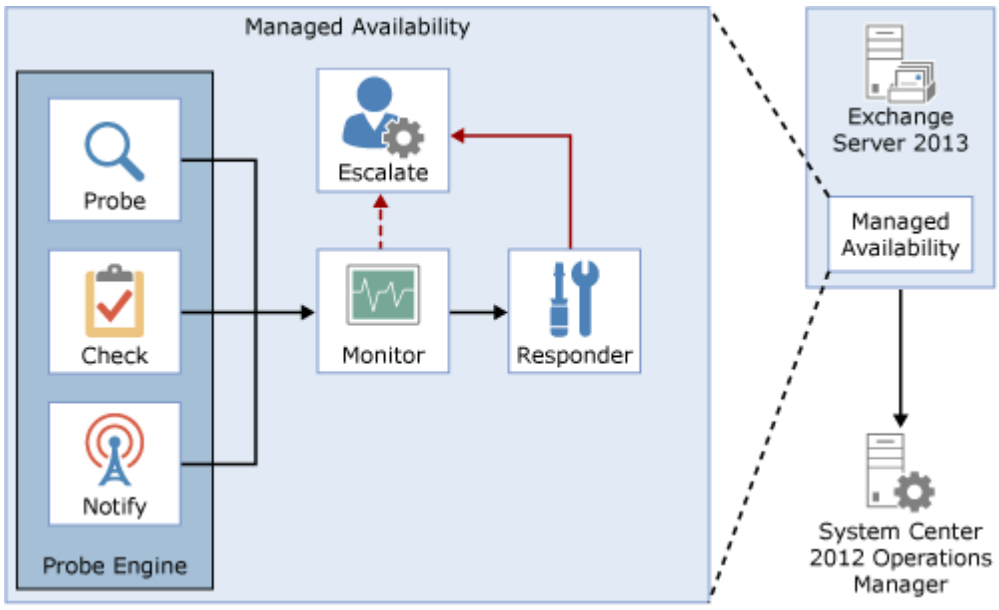
- **Exchange Health Manager Service (MSExchangeHMHost.exe)** This is a controller process used to manage worker processes. It's used to build, execute, and start and stop the worker process, as needed. It's also used to recover the worker process in case that process fails, to prevent the worker process from being a single point of failure.
- **Exchange Health Manager Worker process (MSExchangeHMWorker.exe)** This is the worker process responsible for performing run-time tasks within the managed availability framework.

Managed availability uses persistent storage to perform its functions:

- XML files in the \bin\Monitoring\config folder are used to store configuration settings for some of the probe and monitor work items.
- Active Directory is used to store global overrides.
- The Windows registry is used to store run-time data, such as bookmarks, and local (server-specific) overrides.
- The Windows crimson channel event log infrastructure is used to store the work item results.
- Health mailboxes are used for probe activity. Multiple health mailboxes will be created on each mailbox database that exists on the server.

As illustrated in the following drawing, managed availability includes three main asynchronous components that are constantly doing work.

Managed Availability Components



The first component is called a *Probe*. Probes are responsible for taking measurements on the server and collecting data. The results of those measurements flow into the second component, the *Monitor*. The monitor contains all of the business logic used by the system based on what is considered healthy on the data collected. Similar to a pattern recognition engine, the monitor looks for the various different patterns on all the collected measurements, and then it decides whether something is considered healthy. Finally, there are *Responders*, which are responsible for recovery

and escalation actions. When something is unhealthy, the first action is to attempt to recover that component. This could include multi-stage recovery actions; for example, the first attempt may be to restart the application pool, the second may be to restart the service, the third attempt may be to restart the server, and the subsequent attempt may be to take the server offline so that it no longer accepts traffic. If the recovery actions are unsuccessful, the system escalates the issue to a human through event log notifications.

There are three primary categories of probes: recurrent probes, notifications, and checks. Recurrent probes are synthetic transactions performed by the system to test the end-to-end user experience. Checks are the infrastructure that perform the collection of performance data, including user traffic, and measure the collected data against thresholds that are set to determine spikes in user failures. This enables the checks infrastructure to become aware when users are experiencing issues. Finally, the notification logic enables the system to take action immediately based on a critical event, without having to wait for the results of the data collected by a probe. These are typically exceptions or conditions that can be detected and recognized without a large sample set.

Recurrent probes run every few minutes and check some aspect of service health. These probes might transmit an email via Exchange ActiveSync to a monitoring mailbox, they might connect to an RPC endpoint, or they might verify Client Access-to-Mailbox connectivity.

All probes are defined on Health Manager service startup in the Microsoft.Exchange.ActiveMonitoring\ProbeDefinition crimson channel. Each probe definitions has many properties, but the most relevant properties are:

- **Name** The name of the probe, which begins with a *SampleMask* of the probe's monitor.
- **TypeName** The code object type of the probe that contains the probe's logic.
- **ServiceName** The name of the health set that contains this probe.
- **TargetResource** The object the probe is validating. This is appended to the name of the probe when it is executed to become a probe result *ResultName*
- **RecurrenceIntervalSeconds** How often the probe executes.
- **TimeoutSeconds** How long the probe will wait before failing.

There are hundreds of recurrent probes. Many of these probes are per-database, so as the number of databases increases, so does the number of probes. Most probes are defined in code and are therefore not directly discoverable.

The basics of a recurrent probe are as follows: start every *RecurrenceIntervalSeconds* and check (or probe) some aspect of health. If the component is healthy, the probe passes and writes an informational event to the Microsoft.Exchange.ActiveMonitoring\ProbeResult channel with a *ResultType* of 3. If the check fails or times out, the probe fails and writes an error event to the same channel. A *ResultType* of 4 means the check failed and a *ResultType* of 1 means that it timed out. Many probes will re-run if they timeout, up to the value of the *MaxRetryAttempts* property.

Note:

Note The ProbeResult crimson channel can get very busy with hundreds of probes running every few minutes and logging an event, so there can be a real impact on the performance of your Exchange server if you try expensive queries against the event logs in a production

environment.

Notifications are probes that are not run by the health manager framework, but by some other service on the server. These services perform their own monitoring, and then feed their data into the Managed Availability framework by directly writing probe results. You won't see these probes in the ProbeDefinition channel, as this channel only describes probes that will be run by the Managed Availability framework. For example, the ServerOneCopyMonitor Monitor is triggered by probe results written by the MSExchangeDAGMgmt service. This service performs its own monitoring, determines whether there is a problem, and logs a probe result. Most notification probes have the capability to log both a red event that turns the monitor unhealthy and a green event that makes the monitor healthy again.

Checks are probes that only log events when a performance counter passes above or below a defined threshold. They are really a special case of notification probes, as there is a service monitoring the performance counters on the server and logging events to the ProbeResult channel when the configured threshold is met.

To find the counter and threshold that is considered unhealthy, you can look at the monitor for this check. Monitors of the type *Microsoft.Office.Datacenter.ActiveMonitoring.OverallConsecutiveSampleValueAboveThresholdMonitor* or *Microsoft.Office.Datacenter.ActiveMonitoring.OverallConsecutiveSampleValueBelowThresholdMonitor* mean that the probe they watch is a check probe

Monitors query the data collected by probes to determine if action needs to be taken based on a predefined rule set. Depending on the rule or the nature of the issue, a monitor can either initiate a responder or escalate the issue to a human via an event log entry. In addition, monitors define how much time after a failure that a responder is executed, as well as the workflow of the recovery action. Monitors have various states. From a system state perspective, monitors have two states:

- **Healthy** The monitor is operating properly and all collected metrics are within normal operating parameters
- **Unhealthy** The monitor isn't healthy and has either initiated recovery through a responder or notified an administrator through escalation.

From an administrative perspective, monitors have additional states that appear in the Shell:

- **Degraded** When a monitor is in an unhealthy state from 0 through 60 seconds, it's considered Degraded. If a monitor is unhealthy for more than 60 seconds, it is considered Unhealthy.
- **Disabled** The monitor has been explicitly disabled by an administrator.
- **Unavailable** The Microsoft Exchange Health service periodically queries each monitor for its state. If it doesn't get a response to the query, the monitor state becomes Unavailable.
- **Repairing** An administrator sets the Repairing state to indicate to the system that corrective action is in process by a human, which allows the system and humans to differentiate between other failures that may occur at the same time corrective action is being taken (such as a database copy reseed operation).

Every monitor has a *SampleMask* property in its definition. As the monitor executes, it looks for

events in the ProbeResult channel that have a *ResultName* that matches the monitor's *SampleMask*. These events could be from recurrent probes, notifications, or checks. If the monitor's thresholds are achieved, it becomes Unhealthy. From the monitor's perspective, all three probe types are the same as they each log to the ProbeResult channel.

It is worth noting that a single probe failure does not necessarily indicate that something is wrong with the server. It is the design of monitors to correctly identify when there is a real problem that needs fixing. This is why many monitors have thresholds of multiple probe failures before becoming Unhealthy. Even then, many of these problems can be fixed automatically by responders, so the best place to look for problems that require manual intervention is in the Microsoft.Exchange.ManagedAvailability\Monitoring crimson channel. This will include the most recent probe error.

As their name implies, responders execute some sort of response to an alert that was generated by a monitor. Responders take a variety of recovery actions, such as resetting an application worker pool to restarting a server. There are several types of responders:

- **Restart Responder** Terminates and restarts a service
- **Reset AppPool Responder** Stops and restarts an application pool in Internet Information Services (IIS)
- **Failover Responder** Initiates a database or server failover
- **Bugcheck Responder** Initiates a bugcheck of the server, thereby causing a server reboot
- **Offline Responder** Takes a protocol on a server out of service (rejects client requests)
- **Online Responder** Places a protocol on a server back into production (accepts client requests)
- **Escalate Responder** Escalates the issue to an administrator via event logging

In addition to the above listed responders, some components also have specialized responders that are unique to their component.

All responders include throttling behavior, which provide a built-in sequencing mechanism for controlling responder actions. The throttling behavior is designed to ensure that the system isn't compromised or made worse as a result of responder recovery actions. All responders are throttled in some fashion. When throttling occurs, the responder recovery action may be skipped or delayed, depending on the responder action. For example, when the Bugcheck Responder is throttled, its action is skipped, and not delayed.

Health Sets

From a reporting perspective, managed availability has two views of health, one internal and one external. The internal view uses *health sets*. Each component in Exchange 2013 (for example, Outlook Web App, Exchange ActiveSync, the Information Store service, content indexing, transport services, etc.) is monitored by managed availability using probes, monitors, and responders. A group of probes, monitors and responders for a given component is called a *health set*. A health set is a group of probes, monitors, and responders that determine if that component is healthy. The current state of a health set (e.g., whether it is healthy or unhealthy) is determined by using the state

of the health set's monitors. If all of a health set's monitors are healthy, then the health set is in a healthy state. If any monitor is not in a healthy state, then the health set state will be determined by its least healthy monitor.

For detailed steps to view server health or health sets state, see Manage health sets and server health.

Health Groups

The external view of managed availability is composed of *health groups*. Health groups are exposed to System Center Operations Manager 2007 R2 and System Center Operations Manager 2012.

There are four primary health groups:

- **Customer Touch Points** Components that affect real-time user interactions, such as protocols, or the Information Store
- **Service Components** Components without direct, real-time user interactions, such as the Microsoft Exchange Mailbox Replication service, or the offline address book generation process (OABGen)
- **Server Components** The physical resources of the server, such as disk space, memory and networking
- **Dependency Availability** The server's ability to access necessary dependencies, such as Active Directory, DNS, etc.

When the Exchange 2013 Management Pack is installed, System Center Operations Manager (SCOM) acts as a health portal for viewing information related to the Exchange environment. The SCOM dashboard includes three views of Exchange server health:

- **Active Alerts** Escalation Responders write events to the Windows event log that are consumed by the monitor within SCOM. These appear as alerts in the Active Alerts view.
- **Organization Health** A rollup summary of the overall health of the Exchange organization health is displayed in this view. These rollups include displaying health for individual database availability groups, and health within specific Active Directory sites.
- **Server Health** Related health sets are combined into health groups and summarized in this view.

Overrides

Overrides provide an administrator with the ability to configure some aspects of the managed availability probes, monitors, and responders. Overrides can be used to fine tune some of the thresholds used by managed availability. They can also be used to enable emergency actions for unexpected events that may require configuration settings that are different from the out-of-box defaults.

Overrides can be created and applied to a single server (this is known as a *server override*), or they can be applied to a group of servers (this is known as a *global override*). Server override

configuration data is stored in the Windows registry on the server on which the override is applied. Global override configuration data is stored in Active Directory.

Overrides can be configured to last indefinitely, or they can be configured for a specific duration. In addition, global overrides can be configured to apply to all servers, or only servers running a specific version of Exchange.

When you configure an override, it will not take effect immediately. The Microsoft Exchange Health Manager service checks for updated configuration data every 10 minutes. In addition, global overrides will be dependent on Active Directory replication latency.

For detailed steps to view or configure server or global overrides, see Configure managed availability overrides.

Management Tasks and Cmdlets

There are three primary operational tasks that administrators will typically perform with respect to managed availability:

- Extracting or viewing system health
- Viewing health sets, and details about probes, monitors and responders
- Managing overrides

The two primary management tools for managed availability are the Windows Event Log and the Shell. Managed availability logs a large amount of information in the Exchange ActiveMonitoring and ManagedAvailability crimson channel event logs, such as:

- Probe, monitor, and responder definitions, which are logged in the respective *Definition event logs.
- Probe, monitor, and responder results, which are logged in the respective *Results event logs.
- Details about responder recovery actions, including when the recovery action is started, and it is considered complete (whether successful or not), which are logged in the RecoveryActionResults event log.

There are 12 cmdlets used for managed availability, which are described in the following table.

Cmdlet	Description
Get-ServerHealth	Used to get raw server health information, such as health sets and their current state (healthy or unhealthy), health set monitors, server components, target resources for probes, and timestamps related to probe or monitor start or stop times, and state transition times.
Get-HealthReport	Used to get a summary health view that

	includes health sets and their current state.
Get-MonitoringItemIdentity	Used to view the probes, monitors, and responders associated with a specific health set.
Get-MonitoringItemHelp	Used to view descriptions about some of the properties of probes, monitors, and responders.
Add-ServerMonitoringOverride	Used to create a local, server-specific override of a probe, monitor, or responder.
Get-ServerMonitoringOverride	Used to view a list of local overrides on the specified server.
Remove-ServerMonitoringOverride	Used to remove a local override from a specific server.
Add-GlobalMonitoringOverride	Used to create a global override for a group of servers.
Get-GlobalMonitoringOverride	Used to view a list of global overrides configured in the organization.
Remove-GlobalMonitoringOverride	Used to remove a global override.
Set-ServerComponentState	Used to configure the state of one or more server components.
Get-ServerComponentState	Used to view the state of one or more server components.

Manage health sets and server health

Applies to: *Exchange Server 2013 SP1, Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-12-02*

You can use the built-in health reporting cmdlets to perform a variety of tasks related to managed availability, such as:

- Viewing the health of a server or group of servers
- Viewing a list of health sets
- Viewing a list of probes, monitors, and responders associated with a particular health set
- View a list of monitors and their current health

What do you need to know before you begin?

- Estimated time to complete each procedure: 2 minutes
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

View Server Health

You can use the Shell to get a summary of the health of a server running Exchange 2013.

Use the Shell to View Server Health

Run either of the following commands to view the health sets and health information on a server running Exchange 2013.

```
Get-HealthReport -Identity <ServerName>
```

```
Get-ServerHealth -Identity <ServerName> | Format-Table
Server,CurrentHealthSetState,Name,HealthSetName,AlertValue,
HealthGroupName -Auto
```

Run any of the following commands to view the health sets on a server or database availability group running Exchange 2013.

```
Get-ExchangeServer | Get-HealthReport -RollupGroup
```

```
Get-ExchangeServer | Get-HealthReport -RollupGroup -
HealthSetName <HealthSet>
```

```
(Get-DatabaseAvailabiltyGroup <DAGName>).Servers | Get-HealthReport -RollupGroup
```

View a List of Health Sets

A health set is a group of monitors, probes and responders for a component that determine whether the component is healthy or unhealthy. You can use the Shell to view the list of health sets on a server running Exchange 2013.

Use the Shell to View a List of Health Sets

Run the following command to view the health sets on a server running Exchange 2013.

```
Get-HealthReport -Server <ServerName>
```

View the Probes, Monitors and Responders for a Health Set

A health set is a group of monitors, probes and responders for a component that determine whether the component is healthy or unhealthy. You can use the Shell to view the list of probes, monitors, and responders associated with a health set on a server running Exchange 2013.

Use the Shell to View the Probes, Monitors and Responders for a Health Set

Run the following command to view the probes, monitors and responders associated with a health set on a server running Exchange 2013.

```
Get-MonitoringItemIdentity -Server <ServerName> -Identity <HealthSetName> | Format-Table Identity,ItemType,Name -Auto
```

View a List of Monitors and Their Current Health

The health of a monitor is reported by using the "worst of" monitors in the health set. You can view the details of a health set to see which monitors are healthy and which ones are unhealthy.

Use the Shell to View a List of Monitors and Their Current Health

Run the following command to view a list of the monitors and their current health on a server running Exchange 2013.

```
Get-ServerHealth -HealthSet <HealthSetName> -Server <ServerName> | Format-Table Name, AlertValue -Auto
```

Configure managed availability overrides

Applies to: *Exchange Server 2013 SP1, Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-11-15*

Managed availability performs continuous probing to detect possible problems with Exchange components or their dependencies, and it performs recovery actions to make sure the end user experience is not impacted due to a problem with any of these components. However, there may be scenarios where the out-of-box settings may not be suitable for your environment. Managed availability probes, monitors and responders can be customized by creating an override.

There are two types of overrides: local and global. As their names imply, a local override is available only on the server on which it is created, and a global override is used to apply an override to multiple servers. Either type of override can be created for a specific duration or for a specific version of servers.

Note:

When you create an override, it does not take effect immediately. The Microsoft Exchange Health Management service checks for configuration changes every 10 minutes and loads any detected configuration changes. Alternatively, you can restart the service to make the override changes effective immediately.

For additional management tasks related to managed availability, see Manage health sets and server health.

What do you need to know before you begin?

- Estimated time to complete each procedure: 1 minute
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Create a Local Override

The following example shows how to create server-specific local overrides using the Shell.

Use the Shell to Create a Local Override

Use the Add-ServerMonitoringOverride cmdlet to create a server-specific local override.

```
Add-ServerMonitoringOverride -Server ServerName -Identity
<ItemtoOverride> -ItemType <ItemType> -PropertyName
<PropertyName> -PropertyValue <Value> -Duration
<DurationofOverride>
```

When using the ServerMonitoringOverride cmdlets, the Identity is in the form of HealthSetName \MonitoringItemName\TargetResource (for example, FrontEndTransport\OnPremisesInboundProxy or DataProtection\ServiceHealthMSExchangeReplEndpointEscalate\<ServerName>).

How do you know this worked?

To verify that you have successfully created a server override, use the Get-ServerMonitoringOverride cmdlet to view the list of server overrides:

```
Get-ServerMonitoringOverride -Server | Format-List
```

The override should appear in the list.

Remove a Local Override

The following example shows how to remove a server-specific local override using the Shell.

Use the Shell to Remove a Local Override

Use the Remove-ServerMonitoringOverride cmdlet to delete a server-specific local override.

```
Remove-ServerMonitoringOverride -Server <ServerName> -
Identity <ItemtoOverride> -ItemType <ItemType> -
PropertyName <PropertytoRemove>
```

How do you know this worked?

To verify that you have successfully removed a server override, use the Get-ServerMonitoringOverride cmdlet to view the list of server overrides:

```
Get-ServerMonitoringOverride -Server | Format-List
```

The removed override should not appear in the list.

Create a Global Override

The following example shows how to create a global override using the Shell.

Use the Shell to Create a Global Override

Use the Add-GlobalMonitoringOverride cmdlet to create a global override.

```
Add-GlobalMonitoringOverride -Identity <ItemtoOverride> -
ItemType <ItemType> -PropertyName <PropertytoOverride> -
PropertyValue <NewPropertyValue>
```

You can also create a global override that applies to a specific version of Exchange and/or for a specific time duration.

```
Add-GlobalMonitoringOverride -Identity <ItemtoOverride> -
ItemType <ItemType> -PropertyName <PropertytoOverride> -
PropertyValue <NewPropertyValue> -ApplyVersion
<ExchangeVersion> -Duration 45.00:00:00
```

How do you know this worked?

To verify that you have successfully created a global override, use the Get-GlobalMonitoringOverride cmdlet to view the list of global overrides:

```
Get-GlobalMonitoringOverride
```

The override should appear in the list.

Remove a Global Override

The following example shows how to remove a global override using the Shell.

Use the Shell to Remove a Global Override

Use the Remove-GlobalMonitoringOverride cmdlet to delete a global override.

```
Remove-GlobalMonitoringOverride -Identity <ItemtoOverride>
-ItemType <ItemType> -PropertyName <OverriddenProperty>
```

How do you know this worked?

To verify that you have successfully removed a global override, use the Get-GlobalMonitoringOverride cmdlet to view the list of global overrides:

```
Get-GlobalMonitoringOverride
```

The removed override should not appear in the list.

Exchange Management Shell

Exchange Server 2013 >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-11-06*

The Exchange Management Shell, built on Windows PowerShell technology, provides a powerful command-line interface for Microsoft Exchange Server 2013 that enables automation of administrative tasks. With the Shell, you can manage every aspect of Exchange. You can enable new email accounts, create Send and Receive connectors, configure database properties, manage distribution groups, and more. The Shell can perform every task that can be performed by the Exchange Administration Center (EAC) plus things that can't be done in the EAC. In fact, when you do something in the EAC, it's the Shell that's doing the work behind the scenes.

The Shell also provides a robust and flexible scripting platform that can reduce the complexity of current Microsoft Visual Basic scripts. Tasks that previously required many lines in Visual Basic scripts can now be done by using as little as one line of code in the Shell. The Shell provides this flexibility because it doesn't use text as the basis for interaction with the system, but uses an object model based on the Microsoft .NET platform. This object model enables the Shell cmdlets to apply the output from one command to subsequent commands when they are run.

If you want to start using the Shell immediately, see the “Exchange Management Shell basics” section later in this topic. For a list of cmdlets included with Exchange 2013, see Exchange 2013 cmdlets. Otherwise, continuing reading here for more information about the Shell in Exchange 2013.

Looking for a list of Shell topics? See Shell documentation.

Accessing the Exchange Management Shell

With Exchange 2013, you can connect to a remote session on a remote Exchange 2013 computer to perform commands on that remote computer. Whether you use the Shell to administer a local server or administer a server across the country, remote Shell is used to perform the operation in Exchange 2013. If the Exchange management tools are installed and you want to use the Shell, follow the procedure in Open the Shell.

In Exchange 2013, when you click the Shell shortcut, Windows PowerShell opens. Unlike in Microsoft Exchange Server 2007, which uses local Windows PowerShell, a Windows PowerShell snap-in for Exchange isn't loaded. Instead, Windows PowerShell connects to the closest Exchange 2013 server using a required component called Windows Remote Management 3.0, performs authentication checks, and then creates a remote session for you to use. When the remote session is created, you're given access only to the cmdlets and the parameters associated with the management role groups

and management roles you're assigned. For more information about how Exchange uses role groups and roles to manage who can do what tasks, see Permissions.

A benefit of remote Shell is that you don't need to install Exchange-specific tools on your computer. With Windows PowerShell, .NET Framework 4.5 and Windows Remote Management 3.0 installed on any computer running Windows 7, Windows 8, Windows Server 2008 R2 Service Pack 1 (SP1), or Windows Server 2012, you can connect to a remote Exchange 2013 computer to administer it. However, while it's possible to manage an Exchange 2013 server with just Windows PowerShell, .NET Framework 4.5, and Windows Remote Management 3.0, we recommend that you install the Exchange management tools on any computer that you use to manage Exchange 2013. Without the Exchange management tools installed, you need to connect to the remote Exchange 2013 server manually, and you don't have access to the additional capabilities that the Exchange management tools provide.

For more information about connecting to Exchange 2013 servers without the Exchange management tools installed, see Connect to Exchange using remote Shell.

Shell documentation

The following list provides links to topics that will help you learn about and use the Exchange Management Shell.

Exchange Management Shell basics

Open the Shell

Getting help

Cmdlets

Parameters

Identity

Syntax

Pipelining

WhatIf, Confirm, and ValidateOnly switches

Modifying multivalued properties

Working with command output

Comparison operators

Aliases

User-defined variables

Shell variables

Structured data

Arrays

Script security

Scripting with the Exchange Management Shell

Import and export files in the Exchange Management Shell

Manage Exchange Management Shell access

Cmdlet extension agents

Exchange Management Shell quick reference for Exchange 2013

Open the Shell

Exchange Server 2013 > Exchange Management Shell >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

When you open the Exchange Management Shell, you can perform administrative tasks against servers that run Microsoft Exchange Server 2013.

What do you need to know before you begin?

- Estimated time to complete this procedure: less than 1 minute
- The Exchange management tools have been installed on the computer you want to use to administer Exchange servers.
- The user must be enabled to connect to the Shell remotely.
- The user must be assigned at least one management role. For detailed steps, see Permissions.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Open the Shell

1. Click **Start** > **All Programs** > **Microsoft Exchange Server 2013**.
2. Click **Exchange Management Shell**.

Manage Exchange Management Shell access

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-19*

Remote Shell in Microsoft Exchange Server 2013 enables you to manage your server running Exchange Server 2013 from a remote computer, either on your network or from the Internet. You can enable or disable a user's ability to connect to an Exchange server using remote Shell. For more information about remote Shell, see Exchange Management Shell.

For additional management tasks related to remote Shell, see Connect to Exchange using remote Shell.

What do you need to know before you begin?

- Estimated time to complete each procedure: less than 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Remote Shell" entry in the Exchange and Shell infrastructure permissions topic.
- To use remote Shell, users must be a member of a management role group or be directly assigned a management role that enables the user to run Exchange cmdlets. For more information about role groups and management roles, see Permissions.
- You must use the Shell to perform these procedures.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Enable remote Shell for a user

To enable remote Shell for a user, set the *RemotePowerShellEnabled* parameter to `$True` on the **Set-User** cmdlet. This example enables remote Shell for the user David.

```
Set-User David -RemotePowerShellEnabled $True
```

For detailed syntax and parameter information, see Set-User.

After you enable remote Shell for a user, you may also want to:

- Manage role group members
- Open the Shell

Disable remote Shell for a user

To disable remote Shell for a user, set the *RemotePowerShellEnabled* parameter to `$False` on the **Set-User** cmdlet. This example disables remote Shell for the user David.

```
Set-User David -RemotePowerShellEnabled $False
```

For detailed syntax and parameter information, see Set-User.

Connect to Exchange using remote Shell

Exchange Server 2013 > Exchange Management Shell > Manage Exchange Management Shell access >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-02-11*

If you don't have the Exchange management tools installed, you can use Windows PowerShell on your local computer to create a remote Shell session to an Exchange server. It's a simple three-step process where you enter your Exchange credentials, provide the required connection settings, and then import the Exchange cmdlets into your local Windows PowerShell session so that you can use them.

Note:
For ease of management and to receive the benefits that come with them, we recommend that you install the Exchange management tools on any computer that's used to administer an Exchange 2013 server. For more information, see Install the Exchange 2013 management tools and Open the Shell.

For more information about the Exchange Management Shell, see Exchange Management Shell.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes
- You can use the following versions of Windows:
  - Windows 8 or Windows 8.1

- Windows Server 2012 or Windows Server 2012 R2
- Windows 7 Service Pack 1 (SP1)*
- Windows Server 2008 R2 SP1*

* You need to install the Microsoft .NET Framework 4.5 and then either the Windows Management Framework 3.0 or the Windows Management Framework 4.0. For more information, see Installing the .NET Framework 4.5, 4.5.1 and Windows Management Framework 3.0 or Windows Management Framework 4.0.

- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Connect to a remote Exchange server

1. On your local computer, open Windows PowerShell and run the following command.

```
$UserCredential = Get-Credential
```

In the **Windows PowerShell Credential Request** dialog box, type your Exchange user name and password, and then click **OK**.

2. Run the following command. You need to specify the FQDN of an Exchange 2013 Client Access server.

```
$Session = New-PSSession -ConfigurationName
Microsoft.Exchange -ConnectionUri http://<FQDN of Exchange
2013 Client Access server>/PowerShell/ -Authentication
Kerberos -Credential $UserCredential
```

3. Run the following command.

```
Import-PSSession $Session
```

Note:

Be sure to disconnect the remote PowerShell session when you're finished. If you close the Windows PowerShell window without disconnecting the session, you could use up all the remote PowerShell sessions available to you, and you'll need to wait for the sessions to expire. To disconnect the remote PowerShell session, run the following command.

```
Remove-PSSession $Session
```

How do you know this worked?

After Step 3, the Exchange cmdlets are imported into your local Windows PowerShell session as

tracked by a progress bar. If you don't receive any errors, you connected successfully. A quick test is to run an Exchange cmdlet—for example, **Get-Mailbox**—and see the results.

If you receive errors, check the following requirements:

- A common problem is an incorrect password. Run the three steps again and pay close attention to the user name and password you enter in Step 1.
- Windows PowerShell needs to be configured to run scripts. You only need to configure this setting once on your computer, not every time you connect. To enable Windows PowerShell to run signed scripts, run the following command in an elevated Windows PowerShell window (a Windows PowerShell window you opened by selecting **Run as administrator**).

```
Set-ExecutionPolicy RemoteSigned
```

- The account you use to connect to the Exchange server must be enabled for remote Shell. For more information, see Manage Exchange Management Shell access.
- TCP port 80 traffic needs to be open between your local computer and the Exchange server. It's probably open, but it's something to consider if your organization has a restrictive Internet access policy.

See also

The cmdlets that you use in this topic are Windows PowerShell cmdlets. For more information about these cmdlets, see the following topics.

- Get-Credential
- New-PSSession
- Import-PSSession
- Remove-PSSession
- Set-ExecutionPolicy

Basic concepts in Exchange Management Shell

Exchange Server 2013 > Exchange Management Shell >

Topic Last Modified: *2014-04-04*

Getting help

Getting help

Cmdlets and parameters

Cmdlets

Parameters

Structured data

Syntax

Running commands

Aliases

Arrays

Comparison operators

Identity

Import and export files in the Exchange Management Shell

Modifying multivalued properties

Pipelining

WhatIf, Confirm, and ValidateOnly switches

Working with command output

Running scripts

Scripting with the Exchange Management Shell

Script security

Shell variables

User-defined variables

Aliases

Exchange Server 2013 > Exchange Management Shell > Basic concepts in Exchange Management Shell >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-01*

You can assign an Exchange Management Shell cmdlet or Cmd.exe command to an administrator-

defined and easy-to-remember alias in Microsoft Exchange Server 2013. Such aliases can be handy when you frequently use certain cmdlets and want to reduce the typing that you must do.

When an alias is called from the command line, the rules that apply to the cmdlet that is represented by the alias are enforced exactly as when the cmdlet is called. You must supply any required parameters and their values exactly as if you had called the cmdlet name.

Contents

Built-in aliases

Many cmdlets that are used regularly have default, or built-in, aliases assigned to them. These built-in aliases help reduce the typing that you have to do when you administer Exchange 2013 by using the Shell.

For example, the **Get-ChildItem** cmdlet resembles the MS-DOS `Dir` command. Because you are familiar with the `Dir` command, you might want to use the `Dir` alias when you use the Shell instead of typing **Get-ChildItem** every time that you want to view the contents of a directory. The output from the **Get-ChildItem** cmdlet and the `Dir` alias is the same and can be used interchangeably.

You can view a list of built-in aliases by running the **Get-Alias** cmdlet in the Shell.

For more information about aliases, run the following command in the Shell.

```
Get-Help About_Alias
```

Return to top

Creating custom aliases

In addition to the default, or built-in, aliases, you can define and use custom aliases instead of the names of cmdlets that you frequently use. You can use the **Set-Alias** cmdlet to associate cmdlets to familiar command names that have the equivalent functionality in Cmd.exe. You can assign multiple aliases to a single command. However, each alias can only be assigned to a single command. For example, you can have three aliases `Alias1`, `Alias2`, and `Alias3` that are assigned to the **New-Mailbox** cmdlet. You could then use any of the three aliases to run the **New-Mailbox** cmdlet. However, each alias that you create can only be assigned to the **New-Mailbox** cmdlet. You can't,

for example, assign `Alias1` to both the **New-Mailbox** cmdlet and the **Get-Mailbox** cmdlet.

To create a new alias-cmdlet pairing, run the **Set-Alias** cmdlet and supply the name of the alias, together with the name of the cmdlet that you want to call when the alias is entered.

The following table shows several examples of how to create a new alias.

Examples of custom aliases

Alias description	Alias command
Retrieve the contents of a file	`Set-Alias Type Get-Content`
Retrieve the listing of a directory	`Set-Alias Dir Get-ChildItem`
Remove a file	`Set-Alias Erase Remove-Item`
Set **pad** as an alias for Microsoft WordPad	`Set-Alias Pad "${env:programfiles`
Display the list of all defined aliases	`Set-Alias Aliases Get-Alias`

Return to top

Removing an alias

To remove an alias, delete the alias from the alias drive. For example, an administrator creates the `Ls` alias by using the following command.

```
Set-Alias Ls Get-ChildItem
```

Later the administrator decides that the `Ls` alias is no longer needed and uses the following command to remove the `Ls` alias.

```
Remove-Item Alias Ls
```

Importing and exporting aliases

The **Export-Alias** cmdlet writes the current alias list to a file in comma-separated values (CSV) format. You can include the name of the file and its path in the command line. If the path doesn't exist, the cmdlet will create the path for you.

The **Import-Alias** cmdlet reads a text file that has CSV values and brings the list into the Shell as an object. By using the **Export-Alias** cmdlet and **Import-Alias** cmdlet, you can export a list of aliases from the Shell on one computer and import them to the Shell on another computer. Because existing predefined aliases exist on both computers, all alias name conflicts will be ignored and not

imported.

Alias persistence

Aliases that are created from the command line by using the **Set-Alias** cmdlet during a Shell session can be used when the session is active. After the session is closed, the alias definition is lost. To make a user-defined alias persistent and available every time that a new Shell session is opened, you must add the alias definition to your Shell profile. You can modify your Shell profile by running the command `Notepad $Profile`. If the profile directory doesn't exist, you might have to create it first. To find out the path of your profile, run the command `$Profile`.

Alias limitations

Although aliases can be defined for cmdlets and used instead of cmdlet names, you can't include parameters in the definition of the aliases that you define. You must provide parameters as needed when the alias is called, exactly as you would if you called the cmdlet.

Return to top

Arrays

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-04*

An array provides a data structure that can be used to store a collection of data elements of the same type. The Exchange Management Shell supports all kinds of data elements. See the following sections for information about:

Creating arrays

Reading arrays

Manipulating arrays

Associative arrays

For detailed information about how to use arrays, run the following command in the Shell.

```
Get-Help About_Array
```

Creating arrays

You can create and initialize arrays by assigning multiple values to a variable. The values that are stored in the array are delimited by using a comma and are separated from the variable name by the = assignment operator. For example, suppose you want to create an array that is named `$Example` that contains the following seven integer values: `22, 5, 10, 8, 12, 9, 80`. To create the array, enter the following command.

```
$Example = 22,5,10,8,12,9,80
```

In the array, the first data element is at index position `0`, the second is at position `1`, and so on.

Reading arrays

You can reference an array by its variable name, such as `$Example`. You can reference a specific value within the array by using the index number of the position in the array where the value is stored. For example, to reference the first data element in the `$Example` array, enter the following command.

```
Write-Host $Example[0]
```

The Shell will return the value `22` because that is stored in the first array element.

Manipulating arrays

To change the value of a single item in an array, specify the array name, the index you want to modify, the = assignment operator, and the new value that you want to use instead of the existing value. For example, to change the value of the second item in the `$Example` array, index position `1`, to `10`, enter the following command.

```
$Example[1] = 10
```

You can also use the **SetValue** method to change a value. The following example changes the second value, index position `1`, of an array named `$Example` to `500`:

```
$Example.SetValue(500,1)
```

You can append a value to the end of an existing array. For example, to add an additional integer, such as `200`, to the `$Example` array, enter the following command.

```
$Example += 200
```

Associative arrays

Associative arrays are the same as regular arrays. However, they enable the assignment of key-value pairs to a variable. For example, you may want to assign values to keys in an array to be called on when a command is being processed. The following example creates an associative array:

```
$Example = @{blue = 1; red = 2,3}
```

When you enter `$Example` on the command line, you see the following output:

```
Key                            Value
---                            -----
red                            {2, 3}
blue                           1
```

You can retrieve the information that is stored in the array by calling the array as follows:

```
$Example.blue
```

The previous example returns a value `1`.

Because multiple values were assigned to the `red` key, those values make up a nested array. You can reference the values in this nested array by using their index value. You can retrieve the information that is stored in the key's nested array by calling the associative array, `$Example`, with the `red` key and the index of the nested array location that you want to retrieve `1`, as follows:

```
$Example.red[1]
```

The previous example returns the value 3.

For more information about associative arrays, run the following command in the Shell.

```
Get-Help About_Associative_Array
```

Cmdlets

Exchange Server 2013 > Exchange Management Shell > Basic concepts in Exchange Management Shell >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-04*

A *cmdlet*, pronounced "command-let", is the smallest unit of functionality in the Exchange Management Shell. Cmdlets resemble built-in commands in other shells, for example, the `dir` command found in `cmd.exe`. Like these familiar commands, cmdlets can be called directly from the command line in the Shell and run under the context of the Shell, not as a separate process.

Note:

Since Microsoft Exchange Server 2007, there have been changes to how Exchange 2013 uses cmdlets internally due to the use of Windows PowerShell remoting functionality. These changes have little to no impact on how you need to use cmdlets, but they may offer additional flexibility in how you manage your Exchange servers.

Cmdlets are usually designed around repetitive administrative tasks, and, in the Shell, several hundred cmdlets are provided for Exchange-specific management tasks. These cmdlets are available in addition to the non-Exchange system cmdlets included in the basic Windows PowerShell shell design. For information about how to open the Exchange Management Shell, see Open the Shell.

All cmdlets in the Shell are presented in verb-noun pairs. The verb-noun pair is always separated by a hyphen (-) without spaces, and the cmdlet nouns are always singular. Verbs refer to the action that the cmdlet takes. Nouns refer to the object on which the cmdlet takes action. For example, in the **Get-SystemMessage** cmdlet, the verb is **Get**, and the noun is **SystemMessage**. All Shell cmdlets that manage a specific feature share the same noun. The following table provides examples of some verbs available in the Shell.

Note:

By default, if the verb is omitted, the Shell assumes the **Get** verb. For example, when you call **Mailbox**, you retrieve the same results as when you call **Get-Mailbox**.

Examples of verbs in the Exchange Management Shell

Verb	Description
Disable	**Disable** cmdlets set the `Enabled` status of the specified Exchange object to `$False`. This prevents the object from processing data even though the object exists.
Enable	**Enable** cmdlets set the Enabled status of the specified Exchange object to `$True`. This enables the object to process data.
Get	**Get** cmdlets retrieve information about a specific Exchange object. **Note:** Most **Get** cmdlets only return summary information when you run them. To tell the **Get** cmdlet to return verbose information when you run a command, pipe the command to the **Format-List** cmdlet. For more information about the **Format-List** command, see Working with command output. For more information about pipelining, see Pipelining.

Install	**Install** cmdlets install a new object or feature on an Exchange server.	
Move	**Move** cmdlets relocate the specified Exchange object from one container or server to another.	
New	**New** cmdlets create new Exchange object.	
Remove	**Remove** cmdlets delete the specified Exchange object.	
Set	**Set** cmdlets modify the properties of an existing Exchange object.	
Test	**Test** cmdlets test specific Exchange components and provide log files that you can examine.	
Uninstall	**Uninstall** cmdlets remove an object or feature from an Exchange server.	

The following list of cmdlets is an example of a complete cmdlet set. This cmdlet set is used to manage the delivery status notification (DSN) message and mailbox quota message features of Exchange 2013:

- **Get-SystemMessage**
- **New-SystemMessage**
- **Remove-SystemMessage**
- **Set-SystemMessage**

Comparison operators

Exchange Server 2013 > Exchange Management Shell > Basic concepts in Exchange Management Shell >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-05*

The Exchange Management Shell has a rich set of operators that enables comparisons of one object with another object or one object with a set of objects. For more information about comparison operators, run the following command in the Shell:

```
Get-Help About_Comparison_Operators
```

The following table lists the comparison operators that are available in the Shell. Some comparison operators are case-sensitive. If a comparison operator is case-sensitive, the case that is used in the strings that are being compared must match. For example, the string "Test" does not match "test" when you use a comparison operator that is case-sensitive.

Comparison operators that are available in the Exchange Management Shell

Operator	Definition
-eq	Equals (not case-sensitive)
-ieq	Equals (not case-sensitive)
-ceq	Equals (case-sensitive)
-ne	Not equal (not case-sensitive)
-ine	Not equal (not case-sensitive)
-cne	Not equal (case-sensitive)
-lt	Less than (not case-sensitive)
-ilt	Less than (not case-sensitive)
-clt	Less than (case-sensitive)
-gt	Greater than (not case-sensitive)
-igt	Greater than (not case-sensitive)
-cgt	Greater than (case-sensitive)
-le	Less than or equal to (not case-sensitive)
-ile	Less than or equal to (not case-sensitive)
-cle	Less than or equal to (case-sensitive)
-ge	Greater than or equal to (not case-sensitive)
-ige	Greater than or equal to (not case-sensitive)

-cge	Greater than or equal to (case-sensitive)
-contains	The elements in the left operand that is equal to the right operand (not case-sensitive)
-icontains	The elements in the left operand that is equal to the right operand (not case-sensitive)
-ccontains	The elements in the left operand that is equal to the right operand (case-sensitive)
-notcontains	The elements in the left operand that is equal to the right operand (not case-sensitive)
-inotcontains	The elements in the left operand that is equal to the right operand (not case-sensitive)
-cnotcontains	The elements in the left operand that is equal to the right operand (case-sensitive)
-band	Bitwise And
-bor	Bitwise Or
-bnot	Bitwise NOT
-and	Logical and
-or	Logical or
-not	Logical not
-match	Compare strings by using regular expressions (not case-sensitive)
-notmatch	Compare strings by using regular expressions (not case-sensitive)
-imatch	Compare strings by using regular expressions (not case-sensitive)

-inotmatch	Compare strings by using regular expressions (not case-sensitive)
-cmatch	Compare strings by using regular expressions (case-sensitive)
-cnotmatch	Compare strings by using regular expressions (case-sensitive)
-like	Compare strings by using wildcard rules
-notlike	Compare strings by using wildcard rules
-ilike	Compare strings by using wildcard rules (not case-sensitive)
-inotlike	Compare strings by using wildcard rules (not case-sensitive)
-clike	Compare strings by using wildcard rules (case-sensitive)
-cnotlike	Compare strings by using wildcard rules (case-sensitive)

Getting help

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-04*

In Microsoft Exchange Server 2013, the Exchange Management Shell provides many Help resources so that you can use it to its fullest potential. This topic provides the following sections, which describe Help resources and functionality:

- Exchange 2013 Help Exchange 2013 Help contains all the cmdlet Help topics in a role-based and task-based hierarchy. The cmdlet Help topics also link to procedural topics that tell you how to perform specific tasks.

- Help cmdlets The Shell has several Help cmdlets that enable you to find the appropriate information to accomplish your task.
- Help views Help in the Shell contains extensive information about the cmdlets available to you. Help views enable you to access the information that you need about a cmdlet.
- Tab completion You can use tab completion on cmdlet names and parameter names to reduce the amount of typing at a command prompt.

Exchange 2013 Help

Exchange 2013 Help contains the same cmdlet Help information available on each cmdlet in the Shell. However, in Exchange 2013 Help, the Help topics for all the cmdlets are organized by server role and administration task so that you can easily find specific cmdlets associated with the task that you want to perform. Also, cmdlet topics in Exchange 2013 Help are linked to topics that introduce you to the features that they manage, show you how to use the cmdlets to manage that feature, and provide specific details about the feature or common scenarios.

For more information about the cmdlet Help topics available in Exchange 2013 Help, see Exchange 2013 cmdlets.

Help cmdlets

The following tables provide examples of how to use the **Get-Help** and **Get-Command** cmdlets to access the Help information available for each cmdlet in the Shell.

Important:
To view a list of Exchange cmdlets that match a string that you specify, use the **Get-ExCommand** cmdlet. For more information, see the "Examples of how to use miscellaneous Help commands" table later in this section.

The following table provides examples of how the **Get-Help** cmdlet is used.

Examples of how to use the Get-Help cmdlet

Examples	Description
`Get-Help`	When you use the **Get-Help** cmdlet by itself, it gives you basic instructions on how to use the Shell Help system.
`Get-Help <cmdlet>`	When you give the **Get-Help** cmdlet a cmdlet as an argument, it displays the Help information for that cmdlet. For example, to retrieve the Help information for the **Get-SystemMessage** cmdlet, use the following

	command. `Get-Help Get-SystemMessage`
`Get-Help About_*`	The **Get-Help About_*** command provides a list of all general Shell Help topics to help you better understand and use the Shell. If you want to learn more about a topic in the list displayed, run the **Get-Help About_**<*feature*> command. For example, if you want to learn more about wildcards, use the following command. **Note:** You might need to run the **Update-Help** cmdlet to download the Windows PowerShell-provided About_* help files. `Get-Help About_wildcards`.
`Get-Help <cmdlet> -Detailed`	For a description, see the Help views section later in this topic.
`Get-Help <cmdlet> -Full`	For a description, see the Help views section later in this topic.
`Get-Help <cmdlet> -Examples`	For a description, see the Help views section later in this topic.
`Get-Help <cmdlet> -Parameter <parameter name>`	For a description, see the Parameters filter section later in this topic.
`Get-Help <cmdlet> -Online`	For a description, see the Online Help section later in this topic.

The following table provides examples of how the **Get-Command** cmdlet is used.

Examples of how to use the Get-Command cmdlet

Examples	Description
`Get-Command`	The **Get-Command** cmdlet provides a list of all the cmdlets available to the Shell. The **Get-Command** cmdlet allows for wildcard character expansion.

`Get-Command *<string>*`	When you enclose a string with wildcard characters (`*`), the **Get-Command** cmdlet returns a list of all cmdlets and functions that are available to you that contain that string. For example, to find all cmdlets or functions that contain the string "`mailbox`", use the command `Get-Command *mailbox*`. Exchange cmdlets are shown as functions in the output of the **Get-Command** cmdlet.
`Get-Command -Noun <CmdletNoun>`	The **Get-Command -Noun <***CmdletNoun***>** command lists all the cmdlets that exist with the specified noun. This command is useful when you want to view a list of all cmdlets associated with a particular feature. For example, the **Get-Command -Noun SystemMessage** command returns all the cmdlets available for the SystemMessage feature.
`Get-Command -Verb <CmdletVerb>`	The **Get-Command -Verb <***CmdletVerb***>** command lists all the cmdlets that exist with the specified verb. This command is useful when you want to view a list of all cmdlets associated with a particular action. For example, the **Get-Command -Verb Enable** command returns all cmdlets available that perform the enable action.

The following table provides examples of how to use miscellaneous Help commands.

Examples of how to use miscellaneous Help commands

Examples	Description
`Get-ExCommand`	The Get-ExCommand command returns a list of all Exchange cmdlets available to you.
`Get-ExCommand *<string>*`	When you enclose a string with wildcard

	characters (`*`), the **Get-ExCommand** command returns a list of all Exchange cmdlets that are available to you that contain that string. For example, to find all Exchange cmdlets that contain the string "`mailbox`", use the command `Get-ExCommand *mailbox*`. Exchange cmdlets are shown as functions in the output of the **Get-ExCommand** command.
`QuickRef`	The **QuickRef** command opens a link to a printable HTML chart that lists the most frequently used Shell cmdlets. This command works only if the Exchange management tools are installed. To open this chart directly, see Exchange Management Shell quick reference for Exchange 2013.
`<Cmdlet> -?`	Use the **<Cmdlet> -?** command together with any cmdlet to find the same Help information available when you use the **Get-Help** cmdlet. For example, type `Get-SystemMessage -?` to display detailed Help for the **Get-SystemMessage** cmdlet.
`Get-Tip`	The **Get-Tip** cmdlet generates a new Exchange Management Shell Tip of the Day. This cmdlet works only if the Exchange management tools are installed.
`Get-ExBlog`	The **Get-ExBlog** cmdlet opens your default browser to display the Exchange Team blog. This cmdlet works only if the Exchange management tools are installed.

Help views

When a cmdlet is specified as a parameter of the **Get-Help** cmdlet, the Help information for the specified cmdlet is displayed. In some cases, the information returned can be extensive, and you may only want to see specific information. Help views enable you to view specific information about a cmdlet without having to sort through information that you may not need.

The Shell has four views that present different types of information. You can also retrieve a specific parameter or set of similar parameters.

The following table shows the sections displayed in each view.

Help views in the Exchange Management Shell

Help view	Default	Detailed	Full	Examples
Synopsis	X	X	X	X
Syntax	X	X	X	
Description	X	X	X	
Parameters without metadata		X		
Parameters with metadata			X	
Inputs			X	
Outputs			X	
Errors			X	
Examples		X	X	X
Related links	X		X	
Remarks	X	X		

The following table describes each view and provides an example of a command that calls each view.

Examples of Exchange Management Shell Help views

Help view	Examples	Description

Default	`Get-Help Set-Mailbox`	The Default view is displayed when you use the command `Get-Help <cmdlet>`.
`Detailed`	`Get-Help Set-Mailbox -Detailed`	The `Detailed` view is displayed when you use the command `Get-Help <cmdlet> -Detailed`. The parameters returned in the `Parameters` section don't include parameter metadata. For more information, see Parameters.
`Full`	`Get-Help Set-Mailbox -Full`	The `Full` view is displayed when you use the command `Get-Help <cmdlet> -Full`. The parameters returned in the `Parameters` section include the following parameter metadata: • `Required?` • `Position?` • `Default value` • `Accept pipeline input?` • `Accept wildcard characters?` For more information, see Parameters.
`Examples`	`Get-Help Set-Mailbox -Examples`	The `Examples` view is displayed when you use the command `Get-Help <cmdlet> -Examples`.

Parameters filter

In addition to these four Help views, you can also access the description and metadata about a specific parameter or set of similar parameters. You can specify the parameter together with the `Get-Help <cmdlet>` command. The following example shows how you can display the description of the *ForwardingAddress* parameter on the **Set-Mailbox** cmdlet:

```
Get-Help Set-Mailbox -Parameter ForwardingAddress
```

You can also display a set of similar parameters that exist on a specific cmdlet if you specify the partial name of a parameter together with a wildcard character (*). The following example shows how you can display all the parameters on the **Set-Mailbox** cmdlet that contain the word Quota.

```
Get-Help Set-Mailbox -Parameter *Quota*
```

Note:

When you use the *Parameter* parameter with the **Get-Help** cmdlet to retrieve Help information for a cmdlet that has only one parameter, the **Get-Help** cmdlet doesn't return any results, even if you use the wildcard character (*). This is a known issue in Microsoft Windows PowerShell.

Online Help

If a cmdlet has many parameters, it may be difficult to read the Help information for that cmdlet in the Shell. With Exchange 2013, the *Online* switch has been made available. The *Online* switch tells the Shell to open your default Web browser and browse to the online Help topic for the cmdlet. The online Help topic is the same as the Help for the cmdlet in the Shell with the additional benefits of being able to view the topic in a larger window, to search the topic for terms, or to click related links embedded within the topic. For example, to view online Help for the **Set-Mailbox** cmdlet, use the following command:

```
Get-Help Set-Mailbox -Online
```

Using the *Online* switch requires that your computer has a connection to the Internet.

Tab completion

You can use tab completion to reduce typing when you use the Shell. After you type a partial cmdlet name, and then press the TAB key, the Shell completes the cmdlet name if a matching cmdlet is found. If multiple matching cmdlet names are found, each cmdlet name cycles through after you press the TAB key. When you use tab completion with cmdlet names, you must supply at least the verb and the hyphen (-). The following examples show how you can use tab completion when you enter a cmdlet name:

```
Get-Transport<Tab>
Enable-<Tab>
```

Each time you press the TAB key in the first example, the Shell cycles through all the cmdlet names that start with **Get-Transport**. In the second example, the Shell cycles through all cmdlets with the verb **Enable**.

As with cmdlet names, you can also use tab completion when you want the Shell to complete the

partial parameter name that you entered. When you use tab completion with parameter names, you must specify the full cmdlet name either by typing it or by using tab completion. The following examples show how you can use tab completion when you enter a parameter name:

```
Set-Mailbox -Email<Tab>
New-TransportRule -Cond<Tab>
```

Each time you press the TAB key in the first example, the Shell cycles through all the parameter names that start with *Email* on the **Set-Mailbox** cmdlet. In the second example, when you press the TAB key, the Shell completes the *Conditions* parameter on the **New-TransportRule** cmdlet.

Identity

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-04*

The *Identity* parameter is a special parameter that you can use with most cmdlets. The *Identity* parameter gives you access to the unique identifiers that refer to a specific object in Microsoft Exchange Server 2013. This capability lets you perform actions on a specific Exchange 2013 object.

The following sections describe the Identity parameter and provide examples of how you can use it effectively:

Characteristics of the Identity parameter

Wildcard characters in Identity

Examples of the Identity parameter

Characteristics of the Identity parameter

The primary unique identifier of an object in Exchange 2013 is always a GUID. A GUID is a 128-bit identifier, such as `63d64005-42c5-4f8f-b310-14f6cb125bf3`. This GUID never repeats and is therefore always unique. However, you don't want to type such GUIDs regularly. Therefore the *Identity* parameter typically also consists of the values of other parameters, or combined set of values from multiple parameters on a single object. These values are also guaranteed to be unique across that set of objects. You can specify the values of these other parameters, such as *Name* and *DistriguishedName*, or they can be system-generated. The additional parameters that are used, if any, and how they are populated, depend on the object you refer to.

The *Identity* parameter is also considered a positional parameter. The first argument on a cmdlet is

assumed to be the *Identity* parameter when no parameter label is specified. This reduces the number of keystrokes when you type commands. For more information about positional parameters, see Parameters.

The following example shows the use of the *Identity* parameter by using the Receive connector's unique *Name* parameter value. This example also shows how you can omit the *Identity* parameter name because *Identity* is a positional parameter.

```
Get-ReceiveConnector -Identity "From the Internet"
Get-ReceiveConnector "From the Internet"
```

Like all objects in Exchange 2013, this Receive connector can also be referred to by its unique GUID. For example, if the Receive connector named `"From the Internet"` is also assigned the GUID `63d64005-42c5-4f8f-b310-14f6cb125bf3`, you can also retrieve the Receive connector by using the following command:

```
Get-ReceiveConnector 63d64005-42c5-4f8f-b310-14f6cb125bf3
```

Return to top

Wildcard characters in Identity

Some **Get** cmdlets can accept a wildcard character (*) as part of the value you submit to *Identity* when you run the cmdlet. By using a wildcard with the *Identity* parameter, you can specify a partial name and retrieve a list of objects that match that partial name. You can place a wildcard character at the beginning or the end of the *Identity* value, but you can't place the character in the middle of a string. For example, the commands `Get-Mailbox David*` and `Get-Mailbox *anders*` are valid, but `Get-Mailbox Reb*ca` isn't a valid command.

Some **Get** cmdlets retrieve objects in Exchange 2013 that are organized in a hierarchical or parent and children relationship. That is, there may be a collection of parent objects that also contain their own child objects. Objects that have a parent and child relationship may have an *Identity* with the syntax of `<parent>\<child>`.

When an *Identity* parameter has a syntax of `<parent>\<child>`, some cmdlets enable you to use a wildcard character (*) to replace all or some of the parent or child names. For example, if you want to find all of the child objects named "Contoso" in all parent objects, you could use the syntax `"*\Contoso"`. Likewise, if you want to find all of the child objects with a partial name of "Auth" that exist under the `"ServerA"` parent object, you could use the syntax `"ServerA\Auth*"`.

Some, but not all, cmdlets allow you to specify just the child portion of the Identity parameter when you run a command. When you do this, the cmdlets default to the current parent object being accessed. For example, two receive connectors named "Contoso Receive Connector" exist on both MBX1 and MBX2. If you run the command `Get-ReceiveConnector "Contoso Receive Connector"` on MBX2, only the receive connector on the server MBX2 is returned.

The specific behavior of the Identity parameter and wildcard characters is dependent on the cmdlet that's being run. For more information about the cmdlet you're running, see the feature-specific content for that cmdlet.

Return to top

Examples of the Identity parameter

The examples described in this topic illustrate how the *Identity* parameter can accept different unique values to refer to specific objects in the Exchange 2013 organization. These examples also illustrate how the *Identity* parameter label can be omitted to reduce the number of keystrokes when you type commands.

DSN messages

The examples in this section refer to the delivery status notification (DSN) messages that can be configured in an Exchange 2013 organization. The first example shows how to retrieve DSN 5.4.1 by using the **Get-SystemMessage** cmdlet. In the **Get-SystemMessage** cmdlet, the *Identity* parameter consists of several pieces of data that are configured on each DSN message object. These pieces of data include the language that the DSN is written in, whether the DSN is internal or external in scope, and the DSN message code as in the following example:

```
Get-SystemMessage en\internal\5.4.1
```

You can also retrieve this DSN message by using its GUID as in the following example, because all objects in Exchange 2013 have a GUID:

```
Get-SystemMessage 82ca7bde-1c2d-4aa1-97e1-f298a6f10222
```

For more information about the makeup of the *Identity* parameter when it's used with the **SystemMessage** cmdlets, see DSN message identity.

Management role entries

The examples in this section refer to management role entries that make up management roles in Exchange 2013. Management roles are used to control the permissions that are granted to administrators and end users. Management role entries are made up of two parts: the management role they're associated with and a cmdlet. The Identity parameter is likewise made up of both the management role name and the cmdlet name. For example, the following is the role entry for the **Set-Mailbox** cmdlet on the `Mail Recipients` role:

```
Mail Recipients\Set-Mailbox
```

The `Mail Recipients\Set-Mailbox` role entry is one of several entries on the `Mail Recipients` role. To

view all the role entries on the `Mail Recipients` role, you can use the following command:

```
Get-ManagementRoleEntry "Mail Recipients\*"
```

To view all the role entries on the `Mail Recipients` role that contain the string "`Mailbox`", use the following command:

```
Get-ManagementRoleEntry "Mail Recipients\*Mailbox*"
```

To view all the management roles where **Set-Mailbox** is one of the role entries, use the following command:

```
Get-ManagementRoleEntry *\Set-Mailbox
```

With role entries you can use the wildcard character in a variety of ways to query Exchange 2013 for the information you're interested in.

For more information about management roles, see Understanding management roles.

Return to top

Import and export files in the Exchange Management Shell

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-10*

Microsoft Exchange Server 2013 uses Windows PowerShell command-line interface remoting to establish a connection between the server or workstation from which you're administering Exchange and the server running Exchange 2013 that you're administering. In Exchange 2013, this is called remote Exchange Management Shell, or remote Shell. Even if you're administering the local Exchange 2013 server, remote Shell is used to make the connection. For more information about local and remote Shell, see Exchange Management Shell.

How you import and export files to and from an Exchange server in Exchange 2013 is different than how you might have done it in Exchange Server 2007. This is due to the use of remote Shell in Exchange 2013. This topic discusses why this new process is required and how to import and export files between a local server or workstation and an Exchange 2013 server.

Windows PowerShell sessions

To understand why you need a special syntax to import and export files in remote Shell, you need to know how the Shell is implemented in Exchange 2013. The Shell uses Windows PowerShell sessions, which are the environments in which variables, cmdlets, and so on, can share information. Every time you open a new Shell window, you create a new session. The cmdlets that are run in each window can access variables and other information stored in that window, but can't access variables in other open Shell windows. This is because they're each contained within their own Windows PowerShell session. Windows PowerShell sessions can also be referred to as runspaces.

Remote Shell in Exchange 2013 has two sessions, the local session and the remote session. The local session is the Windows PowerShell session that's running on your local computer. This session contains all of the cmdlets that ship with Windows PowerShell. It also has access to your local file system.

The remote session is the Windows PowerShell session that's running on the remote Exchange server. This session is where all Exchange cmdlets are run. It has access to the Exchange server's file system.

When you connect to a remote Exchange server, a connection is made between your local session on your computer and the remote session on the Exchange server. This connection enables you to run Exchange cmdlets on the remote Exchange server in your local session even though your local computer doesn't have any Exchange cmdlets installed. Even though the Exchange cmdlets appear to be running on your local computer, they're actually running on the Exchange server.

Important:
Even if you open the Shell on an Exchange 2013 server, the same connection process takes place and two sessions are created. This means that you must use the same new syntax to import and export files whether you're opening the Shell on an Exchange 2013 server or from a remote client workstation.

The Exchange cmdlets that run in the remote session on the remote Exchange server don't have access to your local file system. This means that you can't use Exchange cmdlets, on their own, to import or export files from or to your local file system. Additional syntax needs to be used to transfer the files to and from your local file system so that the Exchange cmdlets running on the remote Exchange server can use the data. For more information about the required syntax, see "Importing and exporting files in remote Shell" later in this topic.

Importing and exporting files in remote Shell

Importing and exporting files requires a specific syntax because Mailbox and Client Access servers use remote Shell and don't have access to the local computer's file system.

Importing files in remote Shell

The syntax to import files in Exchange 2013 is used any time you want to send a file to a cmdlet running on an Exchange 2013 server from your local computer or server. Cmdlets that accept data from a file on your local computer will have a parameter called *FileData* (or something similar). To determine the correct parameter to use, see the Help information for the cmdlet you're using.

The Shell must know what file you want to send to the Exchange 2013 cmdlet, and what parameter will accept the data. To do so, use the following syntax.

```
<Cmdlet> -FileData ([Byte[]]$(Get-Content -Path <local path
to file> -Encoding Byte -ReadCount 0))
```

For example, the following command imports the file C:\MyData.dat into the *FileData* parameter on the **Import-SomeData** fictional cmdlet.

```
Import-SomeData -FileData (Byte[]]$(Get-Content -Path "C:
\MyData.dat" -Encoding Byte -ReadCount 0))
```

The following actions occur when the command is run:

1. The command is accepted by remote Shell.
2. Remote Shell evaluates the command and determines that there's an embedded command in the value being provided to the *FileData* parameter.
3. Remote Shell stops evaluating the **Import-SomeData** command and runs the **Get-Content** command. The **Get-Content** command reads the data from the MyData.dat file.
4. Remote Shell temporarily stores the data from the **Get-Content** command as a `Byte[]` object so that it can be passed to the **Import-SomeData** cmdlet.
5. Execution of the **Import-SomeData** command resumes. Remote Shell sends the request to run the **Import-SomeData** cmdlet to the remote Exchange 2013 server, along with the object created by the **Get-Content** cmdlet.
6. On the remote Exchange 2013 server, the **Import-SomeData** cmdlet is run, and the data stored in the temporary object created by the **Get-Content** cmdlet is passed to the *FileData* parameter. The **Import-SomeData** cmdlet processes the input and performs whatever actions are required.

Some cmdlets use the following alternate syntax that accomplishes the same thing as the preceding syntax.

```
[Byte[]]$Data = Get-Content -Path <local path to file> -
Encoding Byte -ReadCount 0
Import-SomeData -FileData $Data
```

The same process happens with this alternate syntax. The only difference is instead of performing the entire operation at once, the data retrieved from the local file is stored in a variable that can be referenced after it's created. The variable is then used in the import command to pass the contents of the local file to the **Import-SomeData** cmdlet. Using this two-step process is useful when you want to use the data from the local file in more than one command.

There are limitations that you must consider when importing files. For more information, see "Limitations on importing files" later in this topic.

For specific information about how to import data into Exchange 2013, see the Help topics for the feature you're managing.

Limitations on importing files

Limits must be set when importing data in remote Shell to preserve the integrity of the data that's being transferred. Transfers that are in progress can't be resumed if they're interrupted. Also, because data being transferred is stored in the remote server's memory, the server must be protected from memory exhaustion caused by excessively large amounts of data.

For these reasons, the amount of data that's transferred to a remote Exchange 2013 server from a local computer or server is limited to the following:

- 500 megabytes (MB) for each cmdlet that's run
- 75 MB for each object that's passed to a cmdlet

If you exceed either of the limits, the execution of the cmdlet and its associated pipeline will stop and you'll receive an error. Consider the examples in the following table to understand how these limits work.

Import data limit examples

Number of objects	Object size (MB)	Total size (MB)	Result of operation
10	40	400	The operation is successful because neither the size of the individual objects exceeds 75 MB nor the total amount of data passed to the cmdlet exceeds 500 MB.
5	80	400	The operation fails because, although the total amount of data passed to the cmdlet is only 400 MB, the size of each individual object exceeds the 75

			MB limit.
120	5	600	The operation fails because, although each individual object is only 5 MB, the total amount of data passed to the cmdlet exceeds the 500 MB limit.

Due to the size limits that have been placed on the amount of data that can be transferred between a remote Exchange 2013 server and a local computer, not all cmdlets that once supported importing support this method of data transfer. To determine whether a specific cmdlet supports this method, see the Help information for the specific cmdlet.

These limits should accommodate the majority of typical operations that can be performed on an Exchange 2013 server. If the limits are lowered, you may find that some normal operations fail because they exceed the new limits. If the limits are raised, the data being transferred could take longer to transfer and become more at risk to transient conditions that interrupt the data transfer. Also, you may exhaust the memory on the remote server if you haven't installed enough memory to allow the server to store the entire block of data during transfer. Each possibility could result in data loss and therefore we recommend you don't change the default limits.

Exporting files in remote Shell

The syntax to export files in Exchange 2013 is used any time you want to accept data from a cmdlet running on a remote Exchange 2013 server and store the data on your local computer or server. Cmdlets that provide data that you can save to a local file will output an object that will contain the **FileData** property (or something similar). Depending on the cmdlet, the **FileData** property is only populated on the object that's output in specific situations. To determine the correct property to use and when it can be used, see the Help information for the cmdlet you're using.

The Shell must know that you want to save the data stored in the **FileData** property to your local computer. To do so, use the following syntax.

```
<cmdlet> | ForEach { $_.FileData | Add-Content <local path
to file> -Encoding Byte }
```

For example, the following command exports the data stored in the **FileData** property on the object created by the **Export-SomeData** fictional cmdlet. The exported data is stored in a file you specify on the local computer, in this case MyData.dat.

Note:

This procedure uses the **ForEach** cmdlet, objects, and pipelining. For more information about each, see Pipelining and Structured data.

```
Export-SomeData | ForEach { $_.FileData | Add-Content C:
\MyData.dat -Encoding Byte }
```

The following actions occur when the command is run:

1. The command is accepted by remote Shell.
2. Remote Shell calls the **Export-SomeData** cmdlet on the remote Exchange 2013 server.
3. The output object created by the **Export-SomeData** cmdlet is passed back to the local Shell session via the pipeline.
4. The output object is then piped to the **ForEach** cmdlet, which has a script block.
5. Within the script block, the **FileData** property on the current object in the pipeline is accessed. The data contained within the **FileData** property is piped to the **Add-Content** cmdlet.
6. The **Add-Content** cmdlet saves the data piped from the **FileData** property to the file MyData.dat on the local file system.

For specific information about how to export data from Exchange 2013, see the Help topics for the feature you're managing.

Modifying multivalued properties

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

A multivalued property is a property that can contain more than one value. For example, the **BlockedRecipients** property on the **RecipientFilterConfig** object can accept multiple recipient addresses as in the following examples:

- john@contoso.com
- kim@northwindtraders.com
- david@adatum.com

Because the **BlockedRecipients** property can accept more than one value, it's called a multivalued property. This topic explains how to use the Exchange Management Shell to add values to and remove values from a multivalued property on an object.

For more information about objects, see Structured data. For more information about the Shell, see Exchange Management Shell.

Modifying a multivalued property vs. modifying a property that accepts only a single value

How you modify a multivalued property is slightly different from how you modify a property that accepts only one value. When you modify a property that accepts only a single value, you can assign a value directly to it, as in the following command.

```
Set-TransportConfig -MaxSendSize 12MB
```

When you use this command to provide a new value to the **MaxSendSize** property, the stored value is overwritten. This isn't a problem with properties that accept only one value. However, it becomes a problem with multivalued properties. For example, assume that the **BlockedRecipients** property on the **RecipientFilterConfig** object is configured to have the three values that are listed in the previous section. When you run the command `Get-RecipientFilterConfig | Format-List BlockedRecipients`, the following is displayed.

```
BlockedRecipients : {david@adatum.com,
kim@northwindtraders.com, john@contoso.com}
```

Now assume that you've received a request to add a new SMTP address to the blocked recipients list. You run the following command to add the new SMTP address.

```
Set-RecipientFilterConfig -BlockedRecipients
chris@contoso.com
```

When you run the `Get-RecipientFilterConfig | Format-List BlockedRecipients` command again, you will see the following.

```
BlockedRecipients : {chris@contoso.com}
```

This isn't what you expected. You wanted to add the new SMTP address to the existing list of blocked recipients, but instead the existing list of blocked recipients was overwritten by the new SMTP address. This unintended result exemplifies how modifying a multivalued property differs from modifying a property that accepts only a single value. When you modify a multivalued property, you must make sure that you append or remove values instead of overwriting the whole list of values. The following sections show you how to do exactly that.

Modifying multivalued properties

Modifying multivalued properties is similar to modifying single-valued properties. You just need to add some additional syntax to tell the Shell that you want to add or remove values to or from the multivalued property rather than replace everything that’s stored in the property. The syntax is

included, along with the value or values to add or remove to or from the property, as a value on a parameter when you run a cmdlet. The following table shows the syntax that you need to add to a parameter on a cmdlet to modify multivalued properties.

Multivalue property syntax

Action	Syntax
Add one or more values to a multivalued property	`@{Add="<value1>", "<value2>", "<value3>"}`
Remove one or more values from a multivalued property	`@{Remove="<value1>", "<value2>", "<value3>"}`

The syntax that you choose from the Multivalue property syntax table is specified as a parameter value on a cmdlet. For example, the following command adds multiple values to a multivalued property:

```
Set-ExampleCmdlet -Parameter @{Add="Red", "Blue", "Green"}
```

When you use this syntax, the values that you specify are added or removed from the list of values already present on the property. Taking the **BlockedRecipients** example earlier in this topic, we can now add chris@contoso.com without overwriting the rest of the values on this property by using the following command:

```
Set-RecipientFilterConfig -BlockedRecipients
@{Add="chris@contoso.com"}
```

If you wanted to remove david@adatum.com from the list of values, you would use this command:

```
Set-RecipientFilterConfig -BlockedRecipients
@{Remove="david@adatum.com"}
```

More complex combinations can be used, such as adding or removing values to and from a property at the same time. To do so, insert a semicolon (;) between `Add` and `Remove` actions. For example:

```
Set-RecipientFilterConfig -BlockedRecipients
```

```
@{Add="carter@contoso.com", "sam@northwindtraders.com",
"brian@adatum.com"; Remove="john@contoso.com"}
```

If we use the `Get-RecipientFilterConfig | Format-List BlockedRecipients` command again, we can see that the email addresses for Carter, Sam, and Brian have been added while the address for John has been removed.

```
BlockedRecipients : {brian@adatum.com,
sam@northwindtraders.com, carter@contoso.com,
chris@contoso.com, kim@northwindtraders.com}
```

Parameters

Exchange Server 2013 > Exchange Management Shell > Basic concepts in Exchange Management Shell >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-04*

Most cmdlets rely on parameters. Parameters are elements that provide information to the cmdlet, either identifying an object and its attributes to act upon, or controlling how the cmdlet performs its task. The name of the parameter is preceded by a hyphen (-) and followed by the value of the parameter as follows:

```
Verb-Noun -ParameterName <ParameterValue>
```

In this simple example, the hyphen in front of the parameter name tells the Exchange Management Shell that the word that immediately follows the hyphen is a parameter that is passed to the cmdlet and that the next separate word after the parameter is the value of the parameter.

This topic discusses the following parameters and their behavior in the Shell:

Positional parameters

Boolean parameters

Switch parameters

Common Windows PowerShell parameters

Positional parameters

A positional parameter is a parameter that lets you specify the parameter's value without specifying the parameter's name. A parameter is a positional parameter if the `Parameter Position` attribute is an integer. This integer indicates the position on the command line where the cmdlet can find the

parameter's value. For more information about the various attributes that make up a parameter, see the Parameter Details section later in this topic.

Most cmdlets only have one positional parameter, *Identity*. *Identity* is always in position `1` if it is available on a cmdlet. Some cmdlets have multiple positional parameters. With these cmdlets, you can specify the values for each positional parameter in the order specified by the `Parameter Position` attribute on each parameter. The values for each parameter must be in the correct position on the command line to work correctly.

If a parameter isn't a positional parameter, it's considered to be a `named` parameter. You must specify the parameter name and parameter value for `named` parameters.

The following two commands perform the same task of returning configuration information for a Receive connector that is named `"Contoso"`.

```
Get-ReceiveConnector -Identity "Contoso"
Get-ReceiveConnector "Contoso"
```

The following two commands perform the same task. The positional parameter values in the first command are placed in the exact order as required by the Parameter Position attribute on each parameter.

```
Set-ExampleCmdlet "Seattle Users" $True "Contoso.com"
Set-ExampleCmdlet -Name "Seattle Users" -Enabled $True -
Domain "Contoso.com"
```

Parameter details

Attributes, also called metadata, on each parameter are included in the PARAMETERS section of the Shell Help that is retrieved by the **Get-Help** cmdlet. The following example is from the **Get-Service** cmdlet.

```
PARAMETERS
    -ServiceName System.String[]
        Parameter required?           false
        Parameter position?           1
        Default value                 *
        Accept pipeline input?       true
        Accept wildcard characters?  True
```

This example from the **Get-Service** cmdlet includes some specific details about the value types that can be passed for the *ServiceName* parameter. Not all cmdlets include such details. However, most cmdlets do include some settings for each parameter as described in the following table.

Parameter settings

Setting	Description
Required?	Indicates whether the cmdlet will run if you don't supply the parameter. When *Required?* is set to `True`, the Shell prompts you for the value if the parameter isn't supplied on the command line.
Position?	Indicates whether you must put the parameter name in front of the parameter value. When *Position?* is set to `Named`, the parameter name is required. When *Position?* is set to an integer, the name isn't required, only the value.
Default value	Indicates the default value for this parameter if no other value is provided.
Accept pipeline input?	Indicates whether the parameter can receive its value as an input through a pipeline from another cmdlet.
Accept wildcard characters?	Indicates whether the parameter's value can contain wildcard characters and can be matched to multiple objects.

Boolean parameters

Boolean parameters are used in the Shell to determine, among other things, whether a feature or option is enabled, `$True`, or disabled, `$False`. The value that you assign to a Boolean parameter is stored in the configuration of the object that you're modifying. When you supply a value to a Boolean parameter, you must use the values `$True` or `1`, or `$False` or `0`. The dollar sign (`$`) must be included with `$True` and `$False`. You may notice that some commands insert a colon (:) between the Boolean parameter name and Boolean value. On Boolean parameters, this colon is optional. The following example disables the Receive connector "Contoso.com":

```
Set-ReceiveConnector "Contoso.com" -Enabled $False
```

Switch parameters

Switch parameters are commonly used to indicate whether the current command should proceed with additional prompting or to enable an alternate option for the command being run. This state isn't saved between commands. Switch parameters resemble Boolean parameters, but they serve different purposes and require different syntax. Switch parameters don't require a value. When you specify a switch parameter on a command line without a value, the parameter evaluates to `$True`.

On some cmdlets, the cmdlet may run as though the switch parameter was included on the command line, even if you didn't include it yourself. This behavior commonly occurs with the *Confirm* switch parameter on cmdlets that can cause data loss if they're inadvertently run. In the case of the *Confirm* switch parameter on such a cmdlet, the cmdlet will always prompt for confirmation before running unless you explicitly tell the cmdlet not to by overriding the switch parameter. You can override the switch parameter by including the *Confirm* switch parameter on the command line with a value of `:$False`. Unlike any other parameters, the colon character (`:`) is required between switch parameters and the value `$False`.

The first of the following examples instructs the **Start-EdgeSynchronization** cmdlet to display a confirmation prompt before it lets EdgeSync synchronization start. The second example instructs the **Remove-ReceiveConnector** cmdlet not to display a confirmation prompt before deleting the Receive connector "Contoso.com":

```
Start-EdgeSynchronization -Confirm
Remove-ReceiveConnector "Contoso.com" -Confirm:$False
```

Common Windows PowerShell parameters

There are several Windows PowerShell parameters that are automatically added to all commands by the Shell. These parameters perform functions that can be used with, or used by, the commands that they're run against. The following table lists all the common parameters that are available in the Shell. Three additional parameters, *WhatIf*, *Confirm*, and *ValidateOnly*, may also be added to cmdlets. For more information about these additional parameters, see WhatIf, Confirm, and ValidateOnly switches.

Common Windows PowerShell parameters in the Exchange Management Shell

Parameter name	Required	Type	Description
Debug	Optional	System.Boolean	The *Debug* parameter instructs the command to provide programmer-level detail about the operation.

ErrorAction	Optional	System.Enum	The *ErrorAction* parameter controls the behavior of the command when an error occurs. Values are as follows: • `Continue`, which is the default value • `Stop` • `SilentlyContinue` • `Inquire`, which asks the user what to do
ErrorVariable	Optional	System.String	The *ErrorVariable* parameter specifies the name of the variable that the command uses to store errors that are encountered during processing. This variable is populated in addition to `$ERROR`.
OutVariable	Optional	System.String	The *OutVariable* parameter specifies the name of the variable that the command uses for objects that are output from this command. This is equivalent to piping the command to `Set-Variable <name> -Passthru:$true`
Verbose	Optional	System.Boolean	The *Verbose* parameter instructs the command to provide detailed information about the

			operation. **Note:** Most **Get** cmdlets only return summary information that contains the most commonly used properties when you run them. To tell the **Get** cmdlet to return all of the properties on an object, pipe the command to the **Format-List** cmdlet. For more information, see Pipelining and Working with command output.

Pipelining

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-04*

Pipelining in the Exchange Management Shell is the act of one cmdlet using the output of another cmdlet when it performs an operation. Pipelining is accomplished by using the pipe "|" symbol. All verbs in the same noun-cmdlet set can use piped information from another command. Some noun-cmdlet sets also let you pass data through the pipeline to another noun cmdlet set.

See the following sections for information and examples of using pipelining in the Shell:

Using pipelining to perform multiple actions

Using pipelining to process data from another cmdlet

Using pipelining to pipe data between dissimilar nouns

Using pipelining to report errors

Using pipelining to perform multiple actions

The use of pipelining to string together the actions of two or more cmdlets gives the Shell the power of composition, which lets you take smaller components and convert them into something more powerful. For example, you can use one cmdlet to gather data, pass that data to a second

cmdlet to filter the data to a subset, and then pass that data to a third cmdlet to act on the subset only.

For example, the following command uses pipelining to move all the mailboxes on `Server1` to the `Executives` database on `Server2` by using the **New-MoveRequest** cmdlet, based on output that is piped from the **Get-Mailbox** cmdlet:

```
Get-Mailbox -Server Server1 | New-MoveRequest -
TargetDatabase Executives
```

Using pipelining to process data from another cmdlet

You can also use pipelining to process data that is output by a cmdlet. For example, for a list of all processes where the `HandleCount` property of the process is larger than `400`, you can run the following command:

```
Get-Process | Where { $_.HandleCount -gt 400 } | Format-
List
```

In this example, the **Get-Process** cmdlet passes objects to the **Where-Object** cmdlet. The **Where-Object** cmdlet picks out the objects that have a property called `HandleCount` with a value larger than `400`.

Note:

Where is an alias for the **Where-Object** cmdlet. For more information, see Aliases.

In this example, the `HandleCount` property is preceded by the `$_` variable. This variable is created automatically by the Shell to store the current pipeline object. The **Where-Object** cmdlet then sends these objects to the **Format-List** cmdlet to be displayed.

The use of structured objects, instead of text, is one of the most exciting capabilities of the Shell. The use of structured objects forms the basis of a powerful compositional model of administration. For more information about structured objects, see Structured data.

Using pipelining to pipe data between dissimilar nouns

Piping data between dissimilar nouns is useful in cases where you want to use the data from one cmdlet with another cmdlet, but the preceding cmdlet in the pipeline doesn't output an object that the next cmdlet can use to identify the object to act upon. This situation typically happens if you're trying to pipe an object from a cmdlet with a noun that's different than the cmdlet that's expecting the object. For more information about cmdlets, see Cmdlets.

To pass data between cmdlets that haven't been optimized to pass objects directly between each other, you need to pass the object through the **ForEach** cmdlet. When you use the **ForEach** cmdlet, you can access the object directly using the `$_` special variable and associate its properties with the

parameters on the second cmdlet.

In the following example, the **Get-Mailbox** cmdlet and the **New-InboxRule** cmdlet aren't optimized to send objects directly between each other. For the **New-InboxRule** cmdlet to take action on objects provided by the **Get-Mailbox** cmdlet, we need to manually associate the correct property on the mailbox object to the correct parameter on the **New-InboxRule** cmdlet. To do this, we use the following command:

```
Get-Mailbox | ForEach { New-InboxRule -Name "Mark as Read"
-Mailbox $_.Name -From john@contoso.com -MarkAsRead $True}
```

In this example, we know that the **New-InboxRule** cmdlet requires that you specify the mailbox on which to create the new inbox rule. We also know that the **Get-Mailbox** cmdlet outputs an object that contains the name of each mailbox being returned. By using the **ForEach** cmdlet, which contains the command to be run on each object it receives, we gain access to the `$_` special variable, which contains the current object in the pipeline. We can access the *Name* property of the current mailbox object using the syntax `$_.Name`. We provide `$_.Name` as an argument on the *Mailbox* parameter of the **New-InboxRule** cmdlet which provides the cmdlet with the information it needs to create the new inbox rule.

Note:

ForEach is an alias for the **ForEach-Object** cmdlet. For more information, see Aliases.

Using pipelining to report errors

To report errors, you can use the error pipeline. The error pipeline lets you report errors while a command runs. You don't have to wait until the command has finished running or to put the error information in the standard result pipeline. The **Write-Error** cmdlet writes its arguments to the error pipeline.

For more information about pipelining, run the following command in the Shell:

```
Get-Help About_Pipeline
```

Script security

Exchange Server 2013 > Exchange Management Shell > Basic concepts in Exchange Management Shell >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

Script security in the Exchange Management Shell helps prevent harmful or otherwise unwanted

scripts from running in your organization. Options are available to modify script security to meet the requirements of your organization.

You typically encounter scripts from several sources: yourself, another person in your organization, and script writers from outside your organization, such as the Internet. If you write a script, you trust the script to do what it's designed to do. If you share the script with other administrators in your organization, they too may trust the script because they trust you.

When scripts come from other sources, such as the Internet, script security is a concern. The only way that you can trust scripts from sources unknown to your organization is to inspect the script code directly and test it in an isolated lab environment. Although this process can be time consuming and tedious, we recommend this practice to prevent unintentional execution of malicious or destructive code.

The Shell supports the recommended use of digital signatures to make sure a script isn't altered after the script is created. For more information about digital signatures, see "Code-signing basics" later in this topic.

Script execution modes

Four modes of script execution in the Shell control how scripts are used, depending on how they are signed and if they are from known or unknown sources. The following table describes each script execution mode.

Important:
The remote Shell requires that the script execution mode be set to `RemotedSigned` or `Unrestricted`. For more information about the remote Shell, see Exchange Management Shell.

Script execution modes

Mode	Description
`Restricted` mode	No scripts will run, even if they are signed by a trusted publisher. This is the default script execution mode.
`AllSigned` mode	All scripts must be digitally signed by a trusted publisher before they will run.
`RemoteSigned` mode	All scripts locally created will run. Scripts downloaded from remote locations, such as the Internet, that can't be trusted, won't run.
`Unrestricted` mode	All scripts, regardless of whether they are digitally signed or trusted, will run. We don't

	recommend the `Unrestricted` mode unless you're running the script in a controlled test environment and not in a production environment.

To change the script execution mode from the default `Restricted` script execution mode, use the **Set-ExecutionPolicy** cmdlet in the Shell. This example changes the execution policy to `RemotedSigned` mode.

```
Set-ExecutionPolicy RemotedSigned
```

The Shell recognizes the change to the policy immediately.

Note:
If you have user access control enabled, you must open the Shell with elevated permissions to set the execution policy. To open the Shell with elevated permissions, right-click the Shell icon and select **Run as administrator**.

Large organizations that want to set a consistent script execution mode for all computers running the Shell should apply the script execution mode setting by using an Active Directory Group Policy. You configure the Active Directory Group Policy to set the `ExecutionPolicy` value located under the **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PowerShell\1\ShellIds\Microsoft.PowerShell** registry key to the desired script execution mode.

Caution:
Incorrectly editing the registry can cause serious problems that may require you to reinstall your operating system. Problems resulting from editing the registry incorrectly may not be able to be resolved. Before editing the registry, back up any valuable data.

Code-signing basics

Digital signatures are created by using a public-key signature algorithm that uses two different cryptographic keys called a key pair: the public key and the private key. The private key is known only to its owner, and the public key is available to anyone. In digital signatures, the private key generates the signature, and the corresponding public key validates the signature.

A *certificate* is a digital document that's generally used for authentication and to help secure information on open networks. A certificate securely binds a public key to the entity that holds the corresponding private key. Certificates are digitally signed by the issuing certification authority (CA). By using a code-signing certificate, the author of the script adds a digital signature to the script file. During this process, a one-way hash of the script is created and encrypted by using the private key. The encrypted hash is a digital signature string that's added to the script file. This digital signature string is commented out so that it doesn't interfere with script functionality.

When this script is run in a Shell environment where code signing is required, a new one-way hash

of the script file is produced. The one-way hash is compared to the encrypted hash included with the script file after it's decrypted by using the public key. If the script wasn't altered in any way after it was signed, the hashes will match. The computer then tries to verify that the signature is from a trusted publisher by building a certificate chain to a trusted CA. If the trust is verified, the script runs.

Whether a script is from a trusted source depends on the origin of the code-signing certificate used to digitally sign the script. There are generally two types of certificates:

- **Certificates issued by a trusted CA** The CA verifies the identity of the requestor before it issues a code-signing certificate. The issuing authority can be an external, public third party that sells certificates or an internal CA hosted by your organization. If you sign a script by using this kind of certificate, you can share the script with users on other computers that recognize and trust the CA that issued the certificate.
- **Self-signed certificates** For this kind of certificate, your computer is the authority that creates the certificate. The benefit of a self-signed certificate is you can write, sign, and run scripts on your computer. But you can't share your script to run on other computers because your computer isn't recognized as a trusted CA. If your computer isn't trusted, your self-signed signature can't be validated, and the script won't run.

Cmdlets for managing code signing

The Shell includes two cmdlets for managing code signing. The **Set-AuthenticodeSignature** cmdlet is used to add digital signatures to script files. The **Set-AuthenticodeSignature** cmdlet takes the name of the file to be signed as its first positional parameter. If the file isn't in the current working directory, you must provide the path of the file. The second input parameter for this cmdlet is the certificate used for signing. This certificate is stored in the local certificate store. You must provide this parameter in the form of a string that references the certificate. The certificate can be accessed through the Cert: drive.

The other cmdlet for managing code signing is the **Get-AuthenticodeSignature** cmdlet. Use the **Get-AuthenticodeSignature** cmdlet to check and confirm the current code-signing status for the file provided as a parameter input. If a problem occurs when you use a code-signed script, the output from the **Get-AuthenticodeSignature** cmdlet will provide useful troubleshooting information.

If you want to run scripts from outside sources, such as Microsoft, you must adapt the scripts according to the script execution mode of your environment. You can receive scripts as basic .txt files, rename them as .ps1 script files, and then after you apply any required signing, run these scripts as if you had written the scripts yourself.

For more information about digital signing and script execution policies, in the Shell, run the following command: `Get-Help About_Signing`. This command returns information that includes detailed instructions for digitally signing scripts.

Scripting with the Exchange Management Shell

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-17*

For most general tasks, running cmdlets one at a time or together through pipelines is sufficient. However, sometimes you may want to automate tasks. The Exchange Management Shell supports a rich scripting language, based on the Microsoft .NET Framework, which resembles the scripting language in other shells. The Shell lets you create scripts, from simple to complex. Language constructs for looping, conditional, flow control, and variable assignment are all supported.

Every organization has tasks that are in some way unique to that organization. With a library of script files to perform these tasks, you can save time by running these scripts on any computer that has the Shell installed.

For more information about how to use scripts, see Scripting with Windows PowerShell. Because the Shell is built on Microsoft Windows PowerShell technology, the scripting guidance for Windows PowerShell applies to the Exchange Management Shell.

Contents

Running a script inside the Shell

Those familiar with the Cmd.exe environment know how to run command shell scripts. These scripts are simply text files that have the .bat file name extension. Like batch files, you can create the Shell script files by using a text editor, such as Notepad. You can also use the Windows PowerShell Integrated Scripting Environment (ISE) to write scripts. The Windows PowerShell ISE provides a rich editing experience with debugging support, syntax coloring, selective execution, and more. The Shell script files use the .ps1 file name extension.

The Shell uses a root directory for script files when they are called. By default, the root directory is the *<root drive>*:\Program Files\Microsoft\Exchange Server\V15\bin directory. You can also verify the current PSHome directory on any computer running the Shell by running `$PSHome` at a command

prompt. Both of these directories are in the PATH environment variable.

If a script file is saved to the root directory, you can call it by using the script name. If the script file is located somewhere other than the current location, the path and script name must be used. If the script file is located in the current location, the script name must be prefixed by the period backslash (.\) characters.

These examples show the command syntax requirements for calling three different scripts. These examples all use the **Get-Date** cmdlet, from three different locations.

```
[PS] C:\>Get-Date-Script-A.ps1
Friday, January 20, 2006 3:13:01 PM
```

The script file Get-Date-Script-A.ps1 is located in the directory specified by `$PSHhome` and requires only the script name to run.

```
[PS] C:\>c:\workingfolder\Get-Date-Script-B.ps1
Friday, January 20, 2006 3:13:25 PM
```

The script file Get-Date-Script-B.ps1 is located in the C:\workingfolder directory so the full path must be supplied to run.

```
[PS] C:\>.\Get-Date-Script-C.ps1
Friday, January 20, 2006 3:13:40 PM
```

The script file Get-Date-Script-C.ps1 is located in the current location, C:\. Therefore, it must be prefixed with .\ to run.

```
[PS] C:\>Get-Date-Script-C.ps1
The term 'Get-Date-Script-C.ps1' is not recognized as the
name of a cmdlet, function, script file, or operable
program. Check the spelling
of the name, or if a path was included, verify that the
path is correct and try again.
At line:1 char:22
+ Get-Date-Script-C.ps1 <<<<
    + CategoryInfo          : ObjectNotFound: (Get-Date-
Script-C.ps1:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
```

In the last example, when this same script, Get-Date-Script-C.ps1, is called without the prefix .\, the expected results are shown.

As a best practice, always give script files a descriptive name and include comments in the script to describe its purpose and to identify each point of interest. Some information about the author

should also be included in case someone running the script has questions about its use. Use the number sign (#) to start comment lines inside the script body.

Running a script from Cmd.exe

If you want to run a script on a scheduled basis using the Windows Task Scheduler service, you can call the Shell and include the script that you want to run as a parameter. If you want to use Exchange cmdlets with your script, you must direct Windows PowerShell to connect to a server running Exchange and load the Exchange cmdlets you have access to. The shortcut you use to open the Shell does this automatically. To do this when you want to run a script that contains Exchange cmdlets, you must direct Windows PowerShell to run the scripts that make this connection. This syntax is required to open Windows PowerShell, connect to an Exchange server, and run your script from the Cmd.exe command.

```
PowerShell.exe -command ". 'D:\Program Files\Microsoft
\Exchange Server\V15\bin\RemoteExchange.ps1'; Connect-
ExchangeServer -auto; <path to your script>"
```

This example runs the script RetrieveMailboxes.ps1 from C:\My Scripts.

```
PowerShell.exe -command ". 'D:\Program Files\Microsoft
\Exchange Server\V15\bin\RemoteExchange.ps1'; Connect-
ExchangeServer -auto; C:\My Scripts\RetrieveMailboxes.ps1"
```

For additional options to use when you call the Shell from the Cmd.exe environment, type **PowerShell.exe /?**

Testing scripts

When you create scripts, you should always test them in a lab environment before you apply them in your production environment. As you test your scripts in your lab, and as you deploy them in your production environment, you can use the *WhatIf* parameter that's available on many cmdlets included in the Shell to verify that your script performs as expected. The *WhatIf* parameter instructs the command to which it is applied to run, but only to display which objects would be affected by running the command and what changes would be made to those objects, without actually changing any of those objects.

For more information about the *WhatIf* parameter, see WhatIf, Confirm, and ValidateOnly switches.

Troubleshooting scripts

Scripts may not work as expected for many reasons. It can be difficult to determine where the

problem is and what's wrong. The Shell can help you locate general syntax errors by reporting the line and character at the point of failure. When the syntax of a script is correct but its behavior is unexpected, it can be much more difficult to diagnose the problem. The Shell includes simple debugging functionality to troubleshoot script files by examining each step that the script makes as it executes. This functionality is called *tracing*.

To enable tracing and examine each command step in a script, use the **Set-PSDebug** cmdlet with the *Trace* parameter set to a value of 1. To examine each step and each variable assignment as they are made, set the *Trace* parameter to a value of 2. To turn tracing off, set the value of the *Trace* parameter to 0 (zero).

To examine each command in a script line by line, use the **Set-PSDebug** cmdlet with the *Step* parameter. At each step, you will be prompted to continue the operation. The following choices are available in step mode.

```
[Y] Yes (continue to the next step)
[A] Yes to All (continue to the end of the script)
[N] No (stop this step)
[L] No to All (stop all remaining steps)
[S] Suspend (suspend at this point and drop to a prompt)
```

Suspend lets you exit to a prompt where you can run any command, for example, to check or set values on an object before the script can access it. When you are ready to resume script execution, type **Exit**, and control immediately returns to the point at which the script was suspended.

Shell variables

Exchange Server 2013 > Exchange Management Shell > Basic concepts in Exchange Management Shell >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-09-19*

Shell variables are a set of variables that are created and declared automatically by the Exchange Management Shell. The variables are maintained throughout your session as part of the system state and are available to all commands, scripts, and applications that run in that session.

The Shell supports two types of shell variables:

- Automatic variables provide a mechanism for passing information to and from commands, scripts, and applications.
- Policy variables store information about the state of the Shell.

You can use shell variables as you would use any other type of variable. For example, the `$PSHome` shell variable stores the name of the directory where the Shell is installed, and the `$_` shell variable stores the current pipeline object. You can use these variables in a command to specify the location of the file and to call a property of the `Get-ChildItem` object, as shown in the following example:

```
Get-ChildItem $PSHome | Sort {$_.Name}
```

This command retrieves all items from the Shell installation directory, and it uses the name property of the object that is stored in the `$_` variable to sort the data when it is displayed.

Common Shell Variables

The following table lists several common automatic variables that are available for your use in the Shell.

Common automatic variables

Automatic variable	Description
`$$`	Contains the last token of the last line that is received by the Shell.
`$?`	Contains the success or fail status of the last operation.
`$^`	Contains the first token of the last line that is received by the Shell.
`$_`	Contains the current pipeline object that is used in script blocks, filters, and the `Where` statement.
`$Error`	Contains objects for which an error occurred when they are processed in a cmdlet.
`$ExBin`	Displays the full path of the <*root drive*>:`\Program Files\Microsoft\Exchange Server\V15\bin\` directory. This variable is only available if the Exchange management tools are installed.
`$ExScripts`	Displays the full path of the Exchange scripts

	directory. This variable is only available if the Exchange management tools are installed.
`$ForEach`	Refers to the enumerator in a `ForEach` loop.
`$Home`	Specifies the user's root directory. It is the equivalent of `%HomeDrive%%HomePath%`.
`$MaximumHistoryCount`	Specifies the maximum number of entries that can be saved in the command history.
`$PSHome`	Specifies the directory where the Shell is installed.

Structured data

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-04*

Each action that you take in the Exchange Management Shell must be done within the context of objects. The Shell uses structured collections of information called objects. These objects represent items in hierarchical data sources. When you call a cmdlet, one or more strongly typed structured objects are returned. Objects carry information about an item and about the object's structure. The object also acts as a proxy for the real item. For example, when you access a file from the Shell, you work with the object that represents that file, and not with the file itself.

Using objects gives the Shell an advantage over other traditional command shells. Traditional command shells have always supported the redirection of the output of one command to another in the form of a textual stream. This method has its disadvantages because parsing text has to be carefully controlled, usually by some kind of encoding to prevent unexpected behavior. By using objects, the Shell enables you to more easily choose the data you want to work with and use predefined methods to utilize and manipulate that data. You spend less time retrieving the data and more time using it.

The Exchange Management Shell uses this object model to pass information from one command to another by using pipelining. This avoids the problems caused by textual parsing in other command shells because the data that the Shell uses has a definite structure and is interpreted according to the object model.

For more information about pipelining, see Pipelining.

Structure of an object

An object consists of three types of data: the object's type, its properties, and its methods.

Object type

The data type of an object provides details about what kind of object it is. For example, an object that represents a mailbox is a **Mailbox** object. An object that represents a file is a **FileInfo** object. All objects have a distinct predefined type and namespace that the Shell can process.

To see what object types are accepted and returned by cmdlets, see Cmdlet Input and Output Types.

Object properties

A property is data associated with an object that specifies a particular state of that object. For example, a **Mailbox** object includes the property **EmailAddresses**. This object property represents the value of the actual attribute **ProxyAddresses** on mailbox-enabled Active Directory user accounts. This is the actual item represented by the **Mailbox** object.

The information about properties included with an object includes the current state and the definition of each property. This includes its name and the type of data that the property can take, such as Integer, Boolean, String, and so on.

Object methods

A method is a set of instructions that defines a particular action that you can take on an object. Methods are defined based on the object type. For example, an object that is of type System.String, or String, has several methods that enable you to manipulate the string. Using the **ToUpper()** method on a string enables you to raise all of the characters within the string to uppercase. Some methods don't take arguments and some require arguments. It depends on the particular method you're using.

To call the methods available to an object, specify the method you want to use after the variable that the object is stored in. The variable and the method are separated by a period. The following example stores a string in the *$Example* variable and then calls the **ToUpper()** method to raise the string to uppercase.

```
$Example = "This is a string"
$Example.ToUpper()
THIS IS A STRING.
```

Notice that if you run `$Example` again, the string itself hasn't been modified.

```
$Example
This is a string.
```

To update the variable with the output of the method, you need to assign the output to the variable as shown in the following example.

```
$Example = "This is a string"
$Example = $Example.ToUpper()
```

Now when you run `$Example`, the string has been changed to uppercase in the variable.

```
$Example
THIS IS A STRING.
```

If an object has properties, the properties can also have their own methods. As with objects, the type of the property defines what methods are available. The property type doesn't necessarily match the object type. To call a method on an object property, you use similar syntax to when you call an object method, but you include the property along with the object. For example, a Send connector object has a property called **MaxMessageSize**, which is of type ByteQuantifiedSize. One of the methods for the type ByteQuantifiedSize is **ToMB()**. The following command displays the value stored in **MaxMessageSize**.

```
$Connector = Get-ReceiveConnector "From Internet"
$Connector.MaxMessageSize
35 MB (36,700,160 bytes)
```

If you now call the **ToMB()** method, the value stored in **MaxMessageSize** is displayed in megabytes.

```
$Connector.MaxMessageSize.ToMB()
35
```

Syntax

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-01*

This topic explains how to read the Exchange Management Shell parameter sets and examples in the Exchange Help documentation and how to format a command so that the Shell can process the command. In the Shell and the Microsoft Exchange Server 2013 Help, parameter sets are displayed in the Syntax section of a cmdlet Help topic. For more information about cmdlet help, see Getting help.

Contents

Command conventions in the Exchange Management Shell

The Shell follows several command conventions that help you understand what information is required or optional when you run a command and how you must present the parameters and their values. See the "Parameter sets" section later in this topic for examples of how parameter sets are presented in the Shell Help and Exchange 2013 Help.

The following table lists these command conventions.

Exchange Management Shell command conventions

Symbol	Description
-	A hyphen indicates that the next word on the command line is a parameter. The most common parameter is *Identity*. For more information, see Parameters.
< >	Angle brackets are used to enclose parameter values. These values can be choices or names. For example, in `-Parameter1 <1 | 2 | 3>`, the numbers represent specific value choices. In `-Parameter2 <ServerName>`, `ServerName` represents the actual value.
[ ]	Square brackets are used to enclose an optional parameter and its value. A parameter

	and its value that are not enclosed in square brackets are required.
|	When the pipe symbol is used in a parameter value list, such as `-Parameter1 <1 | 2 | 3>`, it indicates a choice between available values. This convention applies to `System.Enum` parameters and `System.Boolean` parameters.

These command conventions help you understand how a command should be constructed. Don't type these conventions when you enter the command on the command line.

Parameter sets

In the Exchange Help documentation, all cmdlets display their associated parameters in parameter sets. Parameter sets are groupings of parameters that can be used with each other. Parameters that exist in one parameter set, but not in another parameter set, are mutually exclusive. They can't be used together.

Although all cmdlets have parameter sets, many only have one set of parameters. This means that all the parameters on that cmdlet can be used with each other. Other cmdlets may have several parameter sets. The following example displays the parameter sets that are available on the **New-SystemMessage** cmdlet:

```
New-SystemMessage -DsnCode <EnhancedStatusCode> -Internal
<$true | $false> -Language <CultureInfo> -Text <String>
<COMMON PARAMETERS>
New-SystemMessage -Language <CultureInfo> -QuotaMessageType
<WarningMailboxUnlimitedSize |
WarningPublicFolderUnlimitedSize | WarningMailbox |
WarningPublicFolder | ProhibitSendMailbox |
ProhibitPostPublicFolder | ProhibitSendReceiveMailBox> -
Text <String> <COMMON PARAMETERS>
COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

The **New-SystemMessage** cmdlet has two parameter sets (plus the COMMON PARAMETERS list; more on that later). The first parameter set contains the *DsnCode* parameter and *Internal* parameter, and the second parameter set contains the *QuotaMessageType* parameter. This means that the *DsnCode* parameter and *Internal* parameter can be used with each other. But, they can't be used with the *QuotaMessageType* parameter.

Parameter sets can indicate that a single cmdlet may have multiple uses. For example, you can use the **New-SystemMessage** cmdlet to configure customized delivery status notification (DSN) messages or configure customized mailbox quota limit messages. However, cmdlets typically have multiple parameter sets because one parameter may perform a function that is incompatible with another parameter. For example, the following example displays the parameter sets for the **Set-ManagementScope** cmdlet:

```
Set-ManagementScope [-RecipientRestrictionFilter <String>]
[-RecipientRoot <OrganizationalUnitIdParameter>] <COMMON
PARAMETERS>
Set-ManagementScope -ServerRestrictionFilter <String>
<COMMON PARAMETERS>
Set-ManagementScope -DatabaseRestrictionFilter <String>
<COMMON PARAMETERS>
Set-ManagementScope -
PartnerDelegatedTenantRestrictionFilter <String> <COMMON
PARAMETERS>
COMMON PARAMETERS: -Identity <ManagementScopeIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>]
[-Force <SwitchParameter>] [-Name <String>] [-WhatIf
[<SwitchParameter>]]
```

In the **Set-ManagementScope** cmdlet, the first parameter set lists the parameters that let you create a recipient filter-based management scope. The subsequent parameter sets let you create server and database scopes respectively (partner scopes are a Microsoft internal-only scope and are not publically available). Only the parameters that are listed in each parameter set can be used with each other. For example, you can't use the *RecipientRestrictionFilter* parameter with the *DatabaseRestrictionFilter* parameter in the same command because they're in different parameter sets.

However, you'll notice that each parameter set includes **<COMMON PARAMETERS>**. *Common parameters* can be used with parameters in any parameter set. The list of common parameters is included at the end of all the available parameter sets and begins with **COMMON PARAMETERS**:. You can, for example, use the *Identity* parameter in a command with the *RecipientRestrictionFilter* parameter and in another command with the *ServerRestrictionFilter* parameter, or any other parameter.

Use of quotation marks

Double quotation marks (") are most commonly used to enclose a value that has spaces when you pass that value to a parameter. For example, if you want to pass `Contoso Receive Connector` to the *Name* parameter of the **Set-ReceiveConnector** cmdlet, you must enclose `Contoso Receive`

`Connector` in quotation marks as in the following example:

```
Set-ReceiveConnector -Name "Contoso Receive Connector"
```

If you don't enclose the string in quotation marks, the Shell tries to interpret each word in the string as a new argument on the command line and displays an error.

In the Shell, double quotation marks and single quotation marks (') have different meanings. When you enclose a string in double quotation marks, the Shell replaces any variables with a matching value. For example, assume the value `serverName` is assigned to the variable `$server`. Then, assume the following command is entered on the command line:

```
"$Server Example"
```

The following output is displayed:

```
ServerName Example
```

The variable `$server` is replaced by the value `serverName` in the output.

When you enclose a string in single quotation marks, the Shell doesn't try to replace variables with a matching value. Assume the variable `$server` is still assigned the value `serverName`. Then assume the following command is entered on the command line:

```
'$Server-Example'
```

The following output is displayed:

```
$Server-Example
```

The variable `$server` has not been replaced with a value because the Shell doesn't interpret variables that are included in text that is enclosed in single quotation marks.

For more information about variables, see User-defined variables and Shell variables.

Escape character

You may also want to display some characters, such as the dollar sign (`$`), double or single quotation marks, or back quotation mark (` ). These characters have special meanings when you use them in the Shell. To instruct the Shell not to interpret these characters and to display them when they are included in a string that is enclosed with double quotation marks, you must use the back quotation mark escape character ( `). For example, type the following text on the command line:

```
"The price is `$23."
```

The following output is displayed:

```
The price is $23.
```

Because we used the back quotation escape character with the dollar sign (`$`), the Shell doesn't interpret the `$` as the beginning of a variable.

If you enclose a string in single quotation marks, you don't have to escape any character unless you want to display a single quotation mark in a string. If you want to display a single quotation mark in a string that is enclosed in single quotation marks, you must use two single quotation marks (`''`). For example, type the following on the command line:

```
'Don''t confuse two single quotation marks with a double quotation mark!'
```

The following output is displayed:

```
Don't confuse two single quotation marks with a double quotation mark!
```

Command operators in the Exchange Management Shell

Use the operators in the following table when you type commands in the Shell. Some of the operators may match some of the previously mentioned command conventions. However, they don't have the same meaning when they are typed on the command line. The following table shows the valid operators that you can use in a command.

Exchange Management Shell command operators

Operator	Description
=	The equal sign is used as an assignment character. The value on the right side of the equal sign is assigned to the variable on the left side of the equal sign. The following characters are also assignment characters: • **+=** Add the value on the right side of the equal sign to the current value that is contained in the variable on the left side of the equal sign. • **-=** Subtract the value on the right side of the equal sign from the current value that is contained in the variable on the left side of the equal sign. • ***=** Multiply the current value of the variable on the left side of the equal sign by the value that is specified on the right side

	of the equal sign. • **/=** Divide the current value of the variable on the left side of the equal sign by the value that is specified on the right side of the equal sign. • **%=** Modify the current value of the variable on the left side of the equal sign by the value that is specified on the right side of the equal sign.
:	A colon can be used to separate a parameter's name and the parameter's value, as in the following example: `-Enabled:$True`. The use of a colon is optional with all parameter types except switch parameters. For more information about switch parameters, see Parameters.
!	The exclamation point is a logical **NOT** operator. When it is used with the equal (**=**) sign, the combined pair means "not equal to."
[ ]	Brackets are used to specify the index value of an array position. For example, `$Red[9]` refers to the tenth index position in the array, `$Red`. It refers to the tenth index position because array indexes start at zero (`0`). Brackets can also be used to assign a type to a variable, as in the following example: `$A=[XML] "<Test><A>value</A></Test>"`. The following types are valid: `Array`, `Bool`, `Byte`, `Char`, `Char[]`, `Decimal`, `Double`, `Float`, `Int`, `Int[]`, `Long`, `Long[]`, `RegEx`, `Single`, `ScriptBlock`, `String`, `Type`, and `XML`.
{ }	Braces are used to include an expression in a command, as in the following example: `Get-Process | Where { $_.HandleCount -gt 400 }`
|	The pipe symbol is used when one cmdlet pipes a result to another cmdlet. For example, the following command pipes the results from the Get-Mailbox cmdlet to the Set-Mailbox cmdlet: `Get-Mailbox -Server SRV1 | Set-Mailbox -ProhibitSendQuota 2GB`

>	The right-angle bracket is used to send the output of a command to a file, as in the following example: `Get-TransportRulePredicate > c:\out.txt`. The destination file is overwritten.
>>	Double right-angle brackets are used to append the output of a command to a file, if the file exists. If the file doesn't exist, a new file is created. The following is an example of how to use double right-angle brackets: `Get-TransportRulePredicate >>c:\out.txt`
" "	Quotation marks are used to enclose a string that contains spaces.
$	A dollar sign indicates a variable. For example, `$Blue = 10` assigns the value `10` to the variable `$Blue`.
@	The @ symbol references an associative array. For more information, see Arrays.
$( )	A dollar sign (`$`) with parentheses indicates command substitution. You can use command substitution when you want to use the output of one command as an argument in another command, as in the following example: `Get-ChildItem $(Read-Host -Prompt "Enter FileName: ")`
..	Double-periods indicate a value range. For example, if an array contains several indexes, you can specify the following command to return the values of all indexes between the second and fifth indexes, as in the following example: `$Blue[2..5]`
+	The **+** operator adds two values together. For example, `6 + 6` equals `12`.
-	The **-** operator subtracts one value from another value. For example, `12 - 6` equals `6`. The **-** operator can also be used to represent a negative number,

	such as `-6`. For example, `-6 * 6` equals `-36`.	
*****	A wildcard character has several meanings. You can use wildcard characters to match strings, to multiply numeric values, or, if strings and numeric values are used together, to repeat the string value the number of times that is specified by the numeric value, as in the following example: `"Test" * 3` equals `TestTestTest`.	
/	The **/** operator divides one value by another. For example, `6 / 6` equals `1`.	
%	When used in a numerical evaluation, the **%** operator returns the remainder from a division operator. For example, `6 % 4` equals `2`. When used in a pipeline, the percent character (`%`) is shorthand for the `ForEach` cmdlet. For example, instead of the command `Import-Csv c:\MyFile.csv | ForEach { Set-Mailbox $_.Identity -Name $_.Name }`, you can use `Import-Csv c:\MyFile.csv | % { Set-Mailbox $_.Identity -Name $_.Name }`. For more information, see Pipelining.	
?	The question mark character (`?`) is shorthand for the **Where** cmdlet. For example, instead of `Get-Alias | Where {$_.Definition -eq "Clear-Host"}`, you can use `Get-Alias \| ? {$_.Definition -eq "Clear-Host"}`.	

User-defined variables

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-05*

A variable is a location to store information. Unlike in many programming environments, in the Exchange Management Shell, you don't have to declare a variable before you use it.

You designate a variable by prepending a string with a dollar sign (`$`). You must enclose the string in braces (`{ }`) if the string contains spaces or other special characters. By using the array reference notation (`[ ]`), you can address the elements of an array or hash table variable. For more information, see Arrays.

See the following sections for more information about user-defined variables in the Shell:

Using variables to store values

Storing the output of a command in a variable

Storing the output of the Dir command in a variable

Using variables to store values

Variables are very useful if you want to store a value. You can assign values to variables by using an assignment operator. For more information about operators, see Syntax.

For example, to assign a value of `8` to the variable `$Example`, use the following command:

```
$Example = 8
```

This command assigns the integer `8` to the variable `$Example`. You can then call the `$Example` variable later in another command to recall the value. The values that are specified in a variable are treated exactly as if the value that it contains was typed in the location that the variable is specified. For example, the following two commands are equivalent if `$Example2` is assigned the value `"Hello"`:

```
Write-Host $Example2
Write-Host "Hello"
```

Storing the output of a command in a variable

You can also store the output of commands in a variable for later use. When you assign a command to a variable, the command is evaluated at the time that command is run. The output of that command is assigned to the variable. For example, if you run `$CurrentDate = Get-Date` on the command line and then call `$CurrentDate` repeatedly over several seconds, the value that is reported is the same every time that the variable is called.

When you assign the output of a command to a variable, you can also access the properties and methods of the underlying object. For example, to view the properties and methods that are available when you assign `Get-Date` to `$CurrentDate`, you can use the `$CurrentDate | Get-Member`

command. When you use the `$CurrentDate | Get-Member -MemberType Property` command, the following properties are returned in a list:

```
Name         MemberType Definition
----         ---------- ----------
Date         Property   System.DateTime Date {get;}
Day          Property   System.Int32 Day {get;}
DayOfWeek    Property   System.DayOfWeek DayOfWeek {get;}
DayOfYear    Property   System.Int32 DayOfYear {get;}
Hour         Property   System.Int32 Hour {get;}
Kind         Property   System.DateTimeKind Kind {get;}
Millisecond  Property   System.Int32 Millisecond {get;}
Minute       Property   System.Int32 Minute {get;}
Month        Property   System.Int32 Month {get;}
Second       Property   System.Int32 Second {get;}
Ticks        Property   System.Int64 Ticks {get;}
TimeOfDay    Property   System.TimeSpan TimeOfDay {get;}
Year         Property   System.Int32 Year {get;}
```

You can then call any of these properties by typing the variable, a period (`.`), and then the property that you want to view. For example, to view the year that is stored on a variable, use the following command:

```
$CurrentDate.Year
```

By accessing the properties of a variable, you can easily manipulate and use each piece of information that is stored in the variable without the use of text parsing.

Storing the output of the Dir command in a variable

You can also store the output of the `Dir` command in a variable. Because the `Dir` command returns multiple rows when it runs, each row that is returned is stored in a variable as a new array element. You can then access each file object that is stored in the newly created array. For more information about arrays, see Arrays.

The following command assigns the output of the `Dir` command to the `$DirOutput` variable:

```
$DirOutput = Dir
```

You can then select a specific file object by specifying the array index that you want to view as follows:

```
$DirOutput[1].Name
```

Or you can create a simple loop that cycles through the whole array and displays the name and file size of each file that is stored in the array as follows:

```
0..$DirOutput.Length | ForEach { $DirOutput[$_].Name + " is
" + $DirOutput[$_].Length + " bytes long." }
```

The following list examines this example:

- The `0..$DirOutput.Length` command instructs the Shell to output an integer from `0` to the maximum length of the array that is stored in the `$DirOutput` variable.
- The output of the `0..$DirOutput.Length` command is piped to the `ForEach` command that loops through each element of the array until it reaches the end of the array. The `ForEach` command runs the commands that are enclosed in the braces " `{ }` ".
- The `$_` variable stores the current object that is in the pipeline. In this case, the object in the pipeline is an integer that is produced by the `0..$DirOutput.Length` command as it counts from `0` to the maximum length of the array. This variable is used in the `$DirOutput[$_].Name` command and `$DirOutput[$_].Length` command to select the array element to access.
- For more information about the `$_` variable, see Shell variables.
- The plus " `+` " signs concatenate the output of the `$DirOutput[$_].Name` command and `$DirOutput[$_].Length` command, together with the strings supplied, to create output similar to the following:

```
abv_dg.dll is 416144 bytes long.
addxa.dll is 285056 bytes long.
ASDat.MSI is 5626880 bytes long.
ASEntDat.MSI is 5626880 bytes long.
ASEntIRS.MSI is 910336 bytes long.
ASEntSig.MSI is 45056 bytes long.
BPA.Common.dll is 211848 bytes long.
BPA.ConfigCollector.dll is 101272 bytes long.
BPA.NetworkCollector.dll is 52128 bytes long.
```

These examples show that you can use the `Length` property more than one time to display different information about the same variable. You can do this because more than one type of data is stored in the `$DirOutput` variable. The first type of data is the directory object itself, and the second type of data is the file objects. When you run the `$DirObject.Length` command, without specifying an array index, you're accessing the directory parent object types that are stored in the array. When you specify an array index, such as `$DirObject[5].Length`, you're accessing the file child objects that are stored in the directory object.

This behavior exists on many objects. You can typically access many levels of object data that are contained in a single variable. The ability to access this data makes the Shell quite flexible.

WhatIf, Confirm, and ValidateOnly switches

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-04*

Both experienced administrators and script writers, and administrators who are new to Exchange and scripting, can benefit from using the *WhatIf*, *Confirm*, and *ValidateOnly* switches. These switches let you control how your commands run and indicate exactly what a command will do before it affects data. This functionality is quite valuable as you transition from your test environment into your production environment and as you roll out new scripts or commands.

The *WhatIf*, *Confirm*, and *ValidateOnly* switches are especially useful when you use them with commands that modify objects that are returned by using a filter or by using a **Get** command in a pipeline. This topic describes each switch and also provides an example command for each switch.

Important:

If you want to use the *WhatIf*, *Confirm*, and *ValidateOnly* switches with commands in a script, you must add the appropriate switch to each command in the script, and not on the command line that calls the script.

Note:

WhatIf, *Confirm*, and *ValidateOnly* are called switch parameters. For more information about switch parameters, see Parameters.

WhatIf switch

The *WhatIf* switch instructs the command to which it is applied to run but only to display the objects that would be affected by running the command and what changes would be made to those objects. The switch does not actually change any of those objects. When you use the *WhatIf* switch, you can see whether the changes that would be made to those objects match your expectations, without the worry of modifying those objects.

When you run a command together with the *WhatIf* switch, you put the *WhatIf* switch at the end of the command, as in the following example:

```
New-AcceptedDomain -Name "Contoso Domain" -DomainName "contoso.com" -WhatIf
```

When you run this example command, the following text is returned by the Shell:

```
What if: Creating Accepted Domain "Contoso Domain" with
domain name "contoso.com".
```

Confirm switch

The *Confirm* switch instructs the command to which it is applied to stop processing before any changes are made. The command then prompts you to acknowledge each action before it continues. When you use the *Confirm* switch, you can step through changes to objects to make sure that changes are made only to the specific objects that you want to change. This functionality is useful when you apply changes to many objects and want precise control over the operation of the Shell. A confirmation prompt is displayed for each object before the Shell modifies the object.

By default, the Shell automatically applies the *Confirm* switch to cmdlets that have the following verbs:

- **Clear**
- **Disable**
- **Dismount**
- **Move**
- **Remove**
- **Stop**
- **Suspend**
- **Uninstall**

When a cmdlet runs that has any of these verbs, the Shell automatically stops the command and waits for your acknowledgement before it continues to process.

If you want to manually apply the *Confirm* switch to a command, include the *Confirm* switch at the end of the command, as in the following example:

```
Get-JournalRule | Enable-JournalRule -Confirm
```

When you run this example command, the following confirmation prompt is returned by the Shell:

```
Confirm
Are you sure you want to perform this action?
Enabling journal rule "Litigation Journal Rule".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend
 [?] Help
(default is "Y"):
```

The confirmation prompt gives you the following choices:

- **[Y] Yes** Type **Y** to instruct the command to continue the operation. The next operation will present another confirmation prompt. `[Y] Yes` is the default choice.
- **[A] Yes to All** Type **A** to instruct the command to continue the operation and all subsequent

operations. You will not receive additional confirmation prompts for the duration of this command.

- **[N] No** Type **N** to instruct the command to skip this operation and continue with the next operation. The next operation will present another confirmation prompt.
- **[L] No to All** Type **L** to instruct the command to skip this operation and all subsequent operations.
- **[S] Suspend** Type **S** to pause the current pipeline and return to the command line. Type **Exit** to resume the pipeline.
- **[?] Help** Type **?** to display confirmation prompt Help on the command line.

If you want to override the default behavior of the Shell and suppress the confirmation prompt for cmdlets on which it is automatically applied, you can include the *Confirm* switch with a value of `$False`, as in the following example:

```
Get-JournalRule | Disable-JournalRule -Confirm:$False
```

In this case, no confirmation prompt is displayed.

Caution:

The default value of the *Confirm* switch is `$True`. The default behavior of the Shell is to automatically display a confirmation prompt. If you suppress this default behavior, you instruct the command to suppress all confirmation prompts for the duration of that command. The command will process all objects that meet the criteria for the command without confirmation.

ValidateOnly switch

The *ValidateOnly* switch instructs the command to which it is applied to evaluate all the conditions and requirements that are needed to perform the operation before you apply any changes. The *ValidateOnly* switch is available on cmdlets that may take a long time to run, have several dependencies on multiple systems, or affect critical data, such as mailboxes.

When you apply the *ValidateOnly* switch to a command, the command runs through the whole process. The command performs each action as it would without the *ValidateOnly* switch. But the command doesn't change any objects. When the command completes its process, it displays a summary with the results of the validation. If the validation indicates that the command was successful, you can run the command again without the *ValidateOnly* switch.

When you run a command together with the *ValidateOnly* switch, you put the *ValidateOnly* switch at the end of the command.

Working with command output

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-09*

The Exchange Management Shell offers several methods that you can use to format command output. This topic discusses the following subjects:

- How to format data Control how the data that you see is formatted by using the **Format-List**, **Format-Table**, and **Format-Wide** cmdlets.
- How to output data Determine whether data is output to the Shell console window or to a file by using the **Out-Host** and **Out-File** cmdlets. Included in this topic is a sample script to output data to Microsoft Internet Explorer.
- How to filter data Filter data by using either of the following filtering methods:
  - Server-side filtering, available on certain cmdlets.
  - Client-side filtering, available on all cmdlets by piping the results of a command to the **Where-Object** cmdlet.

To use the functionality that is described in this topic, you must be familiar with the following concepts:

- Pipelining
- Shell variables
- Comparison operators

How to format data

If you call formatting cmdlets at the end of the pipeline, you can override the default formatting to control what data is displayed and how that data appears. The formatting cmdlets are **Format-List**, **Format-Table**, and **Format-Wide**. Each has its own distinct output style that differs from the other formatting cmdlets.

Format-List

The **Format-List** cmdlet takes input from the pipeline and outputs a vertical columned list of all the specified properties of each object. You can specify which properties you want to display by using the *Property* parameter. If the **Format-List** cmdlet is called without any parameters specified, all properties are output. The **Format-List** cmdlet wraps lines instead of truncating them. One of the best uses for the **Format-List** cmdlet is to override the default output of a cmdlet so that you can retrieve additional or more focused information.

For example, when you call the **Get-Mailbox** cmdlet, you only see a limited amount of information in table format. If you pipe the output of the **Get-Mailbox** cmdlet to the **Format-List** cmdlet and add parameters for the additional or more focused information that you want to view, you can retrieve the output that you want.

You can also specify a wildcard character "*" with a partial property name. If you include a wildcard

character, you can match multiple properties without having to type each property name individually. For example, `Get-Mailbox | Format-List -Property Email*` returns all properties that begin with `Email`.

The following examples show the different ways that you can view the same data returned by the **Get-Mailbox** cmdlet.

```
Get-Mailbox TestUser1
Name                      Alias                ServerName
     ProhibitSendQuo


    ta
----                      -----                ----------
     ---------------
TestUser1                 TestUser1            mbx
     unlimited
```

In the first example, the **Get-Mailbox** cmdlet is called without specific formatting so the default output is in table format and contains a predetermined set of properties.

```
Get-Mailbox TestUser1 | Format-List -Property
Name,Alias,EmailAddresses
Name           : TestUser1
Alias          : TestUser1
EmailAddresses : {SMTP:TestUser1@contoso.com}
```

In the second example, the output of the **Get-Mailbox** cmdlet is piped to the **Format-List** cmdlet, together with specific properties. As you can see, the format and content of the output is significantly different.

```
Get-Mailbox TestUser1 | Format-List -Property Name, Alias,
Email*
Name                      : Test User
Alias                     : TestUser1
EmailAddresses            : {SMTP:TestUser1@contoso.com}
EmailAddressPolicyEnabled : True
```

In the last example, the output of the **Get-Mailbox** cmdlet is piped to the **Format-List** cmdlet as in the second example. However, in the last example, a wildcard character is used to match all properties that start with `Email`.

If more than one object is passed to the **Format-List** cmdlet, all specified properties for an object are displayed and grouped by object. The display order depends on the default parameter for the

cmdlet. The default parameter is most frequently the *Name* parameter or the *Identity* parameter. For example, when the **Get-Childitem** cmdlet is called, the default display order is file names in alphabetical order. To change this behavior, you must call the **Format-List** cmdlet, together with the *GroupBy* parameter, and the name of a property value by which you want to group the output. For example, the following command lists all files in a directory and then groups these files by extension.

```
Get-Childitem | Format-List Name,Length -GroupBy Extension
    Extension: .xml
Name   : Config_01.xml
Length : 5627
Name   : Config_02.xml
Length : 3901
    Extension: .bmp
Name   : Image_01.bmp
Length : 746550
Name   : Image_02.bmp
Length : 746550
    Extension: .txt
Name   : Text_01.txt
Length : 16822
Name   : Text_02.txt
Length : 9835
```

In this example, the **Format-List** cmdlet has grouped the items by the *Extension* property that is specified by the *GroupBy* parameter. You can use the *GroupBy* parameter with any valid property for the objects in the pipeline stream.

Format-Table

The **Format-Table** cmdlet lets you display items in a table format with label headers and columns of property data. By default, many cmdlets, such as the **Get-Process** and **Get-Service** cmdlets, use the table format for output. Parameters for the **Format-Table** cmdlet include the *Properties* and *GroupBy* parameters. These parameters work exactly as they do with the **Format-List** cmdlet.

The **Format-Table** cmdlet also uses the *Wrap* parameter. This parameter enables long lines of property information to display completely instead of truncating at the end of a line. To see how the *Wrap* parameter is used to display returned information, compare the output of the **Get-Command** command in the following two examples.

In the first example, when the **Get-Command** cmdlet is used to display command information about the **Get-Process** cmdlet, the information for the *Definition* property is truncated.

```
Get-Command Get-Process | Format-Table Name,Definition
Name                                               Definition
----                                               ----------
get-process                                        get-process [[-
ProcessName] String[]...
```

In the second example, the *Wrap* parameter is added to the command to force the complete contents of the *Definition* property to display.

```
Get-Command Get-Process | Format-Table Name,Definition -
Wrap
Get-Process                                        Get-Process [[-
Name] <String[]>] [-Comp
                                                   uterName <String[]
>] [-Module] [-FileVe
                                                   rsionInfo] [-
Verbose] [-Debug] [-ErrorA
                                                   ction
<ActionPreference>] [-WarningActi
                                                   on
<ActionPreference>] [-ErrorVariable
                                                   <String>] [-
WarningVariable <String>] [
                                                   -OutVariable
<String>] [-OutBuffer <Int
                                                   32>]
                                                   Get-Process -Id
<Int32[]> [-ComputerNam
                                                   e <String[]>] [-
Module] [-FileVersionIn
                                                   fo] [-Verbose] [-
Debug] [-ErrorAction <
                                                   ActionPreference>]
[-WarningAction <Act
                                                   ionPreference>] [-
ErrorVariable <String
                                                   >] [-
WarningVariable <String>] [-OutVar
                                                   iable <String>] [-
```

```
OutBuffer <Int32>]
                                                    Get-Process [-
ComputerName <String[]>]
                                                    [-Module] [-
FileVersionInfo] -InputObje
                                                    ct <Process[]> [-
Verbose] [-Debug] [-Er
                                                    rorAction
<ActionPreference>] [-Warning
                                                    Action
<ActionPreference>] [-ErrorVaria
                                                    ble <String>] [-
WarningVariable <String
                                                    >] [-OutVariable
<String>] [-OutBuffer
                                                    <Int32>]
```

As with the **Format-List** cmdlet, you can also specify a wildcard character "*" with a partial property name. By including a wildcard character, you can match multiple properties without typing each property name individually.

Format-Wide

The **Format-Wide** cmdlet provides a much simpler output control than the other format cmdlets. By default, the **Format-Wide** cmdlet tries to display as many columns of property values as possible on a line of output. By adding parameters, you can control the number of columns and how the output space is used.

In the most basic usage, calling the **Format-Wide** cmdlet without any parameters arranges the output in as many columns as will fit the page. For example, if you run the **Get-Childitem** cmdlet and pipe its output to the **Format-Wide** cmdlet, you will see the following display of information:

```
Get-ChildItem | Format-Wide
    Directory: FileSystem::C:\WorkingFolder
Config_01.xml                               Config_02.xml
Config_03.xml                               Config_04.xml
Config_05.xml                               Config_06.xml
Config_07.xml                               Config_08.xml
Config_09.xml                               Image_01.bmp
Image_02.bmp                                Image_03.bmp
Image_04.bmp                                Image_05.bmp
```

```
Image_06.bmp                    Text_01.txt
Text_02.txt                     Text_03.txt
Text_04.txt                     Text_05.txt
Text_06.txt                     Text_07.txt
Text_08.txt                     Text_09.txt
Text_10.txt                     Text_11.txt
Text_12.txt
```

Generally, calling the **Get-Childitem** cmdlet without any parameters displays the names of all files in the directory in a table of properties. In this example, by piping the output of the **Get-Childitem** cmdlet to the **Format-Wide** cmdlet, the output was displayed in two columns of names. Notice that only one property type can be displayed at a time, specified by a property name that follows the **Format-Wide** cmdlet. If you add the *Autosize* parameter, the output is changed from two columns to as many columns as can fit the screen width.

```
Get-ChildItem | Format-Wide -AutoSize
    Directory: FileSystem::C:\WorkingFolder
Config_01.xml   Config_02.xml   Config_03.xml
Config_04.xml   Config_05.xml
Config_06.xml   Config_07.xml   Config_08.xml
Config_09.xml   Image_01.bmp
Image_02.bmp    Image_03.bmp    Image_04.bmp
Image_05.bmp    Image_06.bmp
Text_01.txt     Text_02.txt     Text_03.txt     Text_04.txt
    Text_05.txt
Text_06.txt     Text_07.txt     Text_08.txt     Text_09.txt
    Text_10.txt
Text_11.txt     Text_12.txt
```

In this example, the table is arranged in five columns, instead of two columns. The *Column* parameter offers more control by letting you specify the maximum number of columns to display information as follows:

```
Get-ChildItem | Format-Wide -Column 4
    Directory: FileSystem::C:\WorkingFolder
Config_01.xml       Config_02.xml       Config_03.xml
 Config_04.xml
Config_05.xml       Config_06.xml       Config_07.xml
 Config_08.xml
Config_09.xml       Image_01.bmp        Image_02.bmp
 Image_03.bmp
```

```
Image_04.bmp          Image_05.bmp          Image_06.bmp
 Text_01.txt
Text_02.txt           Text_03.txt           Text_04.txt
 Text_05.txt
Text_06.txt           Text_07.txt           Text_08.txt
 Text_09.txt
Text_10.txt           Text_11.txt           Text_12.txt
```

In this example, the number of columns is forced to four by using the *Column* parameter.

How to output data

Out-Host and Out-File cmdlets

The **Out-Host** cmdlet is an unseen default cmdlet at the end of the pipeline. After all formatting is applied, the **Out-Host** cmdlet sends the final output to the console window for display. You don't have to explicitly call the **Out-Host** cmdlet, because it's the default output. You can override sending the output to the console window by calling the **Out-File** cmdlet as the last cmdlet in the command. The **Out-File** cmdlet then writes the output to the file that you specify in the command as in the following example:

```
Get-ChildItem | Format-Wide -Column 4 | Out-File c:
\OutputFile.txt
```

In this example, the **Out-File** cmdlet writes the information that is displayed in the **Get-ChildItem | Format-Wide -Column 4** command to a file that is named `OutputFile.txt`. You can also redirect pipeline output to a file by using the redirection operator, which is the right-angle bracket (>). To append pipeline output of a command to an existing file without replacing the original file, use the double right-angle brackets (>>), as in the following example:

```
Get-ChildItem | Format-Wide -Column 4 >> C:\OutputFile.txt
```

In this example, the output from the **Get-Childitem** cmdlet is piped to the **Format-Wide** cmdlet for formatting and then is written to the end of the `OutputFile.txt` file. Notice that if the `OutputFile.txt` file didn't exist, use of the double right-angle brackets (>>) would create the file.

For more information about pipelines, see Pipelining.

For more information about the syntax used in the previous examples, see Syntax.

Viewing data in Internet Explorer

Because of the flexibility and ease of scripting in the Exchange Management Shell, you can take the

data that is returned by commands and format and output them in almost limitless ways.

The following example shows how you can use a simple script to output the data that is returned by a command and display it in Internet Explorer. This script takes the objects that are passed through the pipeline, opens an Internet Explorer window, and then displays the data in Internet Explorer:

```
$Ie = New-Object -Com InternetExplorer.Application
$Ie.Navigate("about:blank")
while ($Ie.Busy) { Sleep 1 }
$Ie.Visible = $True
$Ie.Document.Write("$Input")
# If the previous line doesn't work on your system,
uncomment the line below.
# $Ie.Document.IHtmlDocument2_Write("$Input")
$Ie
```

To use this script, save it to the `C:\Program Files\Microsoft\Exchange Server\V15\Scripts` directory on the computer where the script will be run. Name the file `Out-Ie.ps1`. After you save the file, you can then use the script as a regular cmdlet.

Note:

To run scripts in Exchange 2013, scripts must be added to an unscoped management role and you must be assigned the management role either directly or through a management role group. For more information, see Understanding management roles.

The `Out-Ie` script assumes that the data it receives is valid HTML. To convert the data that you want to view into HTML, you must pipe the results of your command to the **ConvertTo-Html** cmdlet. You can then pipe the results of that command to the `Out-Ie` script. The following example shows how to view a directory listing in an Internet Explorer window:

```
Get-ChildItem | Select Name,Length | ConvertTo-Html | Out-
Ie
```

How to filter data

The Shell gives you access to a large quantity of information about your servers, mailboxes, Active Directory, and other objects in your organization. Although access to this information helps you better understand your environment, this much information can be overwhelming. The Shell lets you control this information and return only the data that you want to see by using filtering. The following types of filtering are available:

- **Server-side filtering** Server-side filtering takes the filter that you specify on the command line and submits it to the Exchange server that you query. That server processes the query and returns only the data that matches the filter that you specified.

Server-side filtering is performed only on objects where tens or hundreds of thousands of results could be returned. Therefore, only the recipient management cmdlets, such as the **Get-Mailbox** cmdlet, and queue management cmdlets, such as the **Get-Queue** cmdlet, support server-side filtering. These cmdlets support the *Filter* parameter. This parameter takes the filter expression that you specify and submits it to the server for processing.

- **Client-side filtering** Client-side filtering is performed on the objects in the local console window in which you are currently working. When you use client-side filtering, the cmdlet retrieves all the objects that match the task that you are performing to the local console window. The Shell then takes all the returned results, applies the client-side filter to those results, and returns to you only the results that match your filter. All cmdlets support client-side filtering. This is invoked by piping the results of a command to the **Where-Object** cmdlet.

Server-side filtering

The implementation of server-side filtering is specific to the cmdlet on which it is supported. Server-side filtering is enabled only on specific properties on the objects that are returned. For more information, see the Help for the following cmdlets:

Get-ActiveSyncDevice	Get-ActiveSyncDeviceClass	Get-CASMailbox	Get-Contact	Get-DistributionGroup
Get-DynamicDistributionGroup	Get-Group	Get-Mailbox	Get-MailboxStatistics	Get-MailContact
Get-MailPublicFolder	Get-MailUser	Get-Message	Get-MobileDevice	Get-Queue
Get-QueueDigest	Get-Recipient	Get-RemoteMailbox	Get-RoleGroup	Get-SecurityPrincipal
Get-StoreUsageStatistics	Get-UMMailbox	Get-User	Get-UserPhoto	Remove-Message
Resume-Message	Resume-Queue	Retry-Queue	Suspend-Message	Suspend-Queue

Client-side filtering

Client-side filtering can be used with any cmdlet. This capability includes those cmdlets that also support server-side filtering. As described earlier in this topic, client-side filtering accepts all the data that is returned by a previous command in the pipeline, and in turn, returns only the results that match the filter that you specify. The **Where-Object** cmdlet performs this filtering. It can be shortened to **Where**.

As data passes through the pipeline, the **Where** cmdlet receives the data from the previous object and then filters the data before passing it on to the next object. The filtering is based on a script block that is defined in the **Where** command. The script block filters data based on the object's properties and values.

The **Clear-Host** cmdlet is used to clear the console window. In this example, you can find all the defined aliases for the **Clear-Host** cmdlet if you run the following command:

```
Get-Alias | Where {$_.Definition -eq "Clear-Host"}
CommandType     Name                                Definition
-----------     ----                                ----------
Alias           clear                               clear-host
Alias           cls                                 clear-host
```

The **Get-Alias** cmdlet and the **Where** command work together to return the list of aliases that are defined for the **Clear-Host** cmdlet and no other cmdlets. The following table outlines each element of the **Where** command that is used in the example.

Elements of the Where command

Element	Description
`{ }`	Braces enclose the script block that defines the filter.
`$_`	This special variable automatically initiates and binds to the objects in the pipeline.
`Definition`	The `Definition` property is the property of the current pipeline objects that stores the name of the alias definition. When `Definition` is used with the `$_` variable, a period comes before the property name.
`-eq`	This comparison operator for "equal to" is used to specify that the results must exactly match the property value that is supplied in the

	expression.
"**Clear-Host**"	In this example, "**Clear-Host**" is the value for which the command is parsing.

In the example, the objects that are returned by the **Get-Alias** cmdlet represent all the defined aliases on the system. Even though you don't see them from the command line, the aliases are collected and passed to the **Where** cmdlet through the pipeline. The **Where** cmdlet uses the information in the script block to apply a filter to the alias objects.

The special variable `$_`represents the objects that are being passed. The `$_`variable is automatically initiated by the Shell and is bound to the current pipeline object. For more information about this special variable, see Shell variables.

Using standard "dot" notation (object.property), the `Definition` property is added to define the exact property of the object to evaluate. The `-eq` comparison operator then compares the value of this property to `"Clear-Host"`. Only the objects that have the `Definition` property that match this criterion are passed to the console window for output. For more information about comparison operators, see Comparison operators.

After the **Where** command has filtered the objects returned by the **Get-Alias** cmdlet, you can pipe the filtered objects to another command. The next command processes only the filtered objects returned by the Where command.

Cmdlet extension agents

Exchange Server 2013 > Exchange Management Shell >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-12-05*

Cmdlet extension agents are components in Microsoft Exchange Server 2013 invoked by Exchange 2013 cmdlets when the cmdlets run. As the name implies, cmdlet extension agents extend the capabilities of the cmdlets that invoke them by assisting in processing data or performing additional actions based on the requirements of the cmdlet. Cmdlet extension agents are available on any server role.

Agents can modify, replace, or extend functionality of Exchange Management Shell cmdlets. An agent can provide a value for a required parameter that isn't provided on a command, override a value provided by a user, perform other actions outside of the cmdlet workflow while a cmdlet runs, and more.

For example, the **New-Mailbox** cmdlet accepts the *Database* parameter that specifies the mailbox

database in which to create a new mailbox. In Microsoft Exchange Server 2007, if you don't specify the *Database* parameter when you run the **New-Mailbox** cmdlet, the command fails. However, in Exchange 2013, the **New-Mailbox** cmdlet invokes the `Mailbox Resources Management` agent when the cmdlet runs. If the *Database* parameter isn't specified, the `Mailbox Resources Management` agent automatically determines a suitable mailbox database on which to create the new mailbox and inserts that value into the *Database* parameter.

Cmdlet extension agents can only be invoked by Exchange 2013 and Microsoft Exchange Server 2010 cmdlets. Exchange 2007 cmdlets and cmdlets provided by other Microsoft and third-party products can't invoke cmdlet extension agents. Scripts also can't invoke cmdlet extension agents directly. However, if scripts contain Exchange 2013 cmdlets, those cmdlets continue to call the cmdlet extension agents.

Looking for management tasks related to cmdlet extension agents? See Manage cmdlet extension agents.

Agent priority

The priority of an agent determines the order in which the agent is invoked while a cmdlet runs. An agent that has a higher priority, closer to zero, is invoked first. The priority of an agent becomes important when two or more agents attempt to set the value of the same property. The highest priority agent that attempts to set a property value succeeds, and all subsequent attempts to set the same property by lower priority agents are ignored. For example, if the **Name** property on an object is modified by an agent with a priority of 3 and another agent with a priority of 6 modifies the same object, the modification made by the agent with a priority of 6 is ignored.

If you want to use the `Scripting agent` to set the value of properties that might be set by other, higher priority agents, you have the following options:

- Disable the agent that currently sets the property.
- Set the `Scripting agent` to a priority higher than the existing agent you want to replace.
- Keep the priorities of the agents the same and make sure that the script that runs under the `Scripting agent` respects the value provided by the other agents.

Caution:

Changing the priority or replacing the functionality of a built-in agent is an advanced operation. Be sure that you completely understand the changes you're making.

For more information about changing the priority of an agent, see Manage cmdlet extension agents.

Built-in agents

Exchange 2013 includes several agents that can be invoked when a cmdlet runs. The following table lists the agents, their order, and whether the agents are enabled by default. You can't add or remove agents to or from a server running Exchange 2013. However, you can use the `Scripting`

agent to run Windows PowerShell scripts to extend the functionality of the cmdlets that use it. For more information about the Scripting agent, see the "Scripting agent" section later in this topic.

You can enable or disable most agents or change the priority of the agents if you want to replace the functionality of a specific agent with functionality you provide in a custom script that you call using the Scripting agent. However, some agents can't be disabled. Agents that can't be disabled are called *system agents* and have their *IsSystem* property set to $True. The following table provides information about Exchange 2013 cmdlet extension agents, including system agents.

The configuration for agents is stored at the organization level. When you enable or disable an agent, or set its priority, you set that agent configuration across every server in the organization. The exception is adding scripts to the Scripting agent. You must update the scripts on each server individually. For more information about configuring scripts for use with the Scripting agent, see the "Scripting agent" section later in this topic.

Caution:

Changing the priority of agents, or enabling or disabling agents, can cause unintended effects if you don't completely understand what each agent does and how they interact with Exchange cmdlets. Before you change the configuration of any agent, be sure you fully understand the changes and results you want and that you verify that your custom script will work as intended.

Exchange 2013 cmdlet extension agents

Agent name	Priority	Enabled by default	System agent
Admin Audit Log agent	255	True	Yes
Scripting agent	6	False	No
Mailbox Resources Management agent	5	True	No
OAB Resources Management agent	4	True	No
Query Base DN agent	3	True	No
Provisioning Policy agent	2	True	No
Rus agent	1	True	No
Mailbox Creation Time agent	0	True	No

Scripting agent

You can use the Scripting agent cmdlet extension agent in Exchange 2013 to insert your own scripting logic into the execution of Exchange cmdlets. Using the Scripting agent, you can add

conditions, override values, and set up reporting.

Caution:

When you enable the `Scripting agent` cmdlet extension agent, the agent is invoked every time a cmdlet is run on a server running Exchange 2013. This includes not only cmdlets run directly by you in the Exchange Management Shell, but also cmdlets run by Exchange services, and the Exchange Administration Center (EAC). We strongly recommend that you test your scripts and any changes you make to the configuration file before you copy your updated configuration file to your Exchange 2013 servers and enable the `Scripting agent` cmdlet extension agent.

Every time an Exchange cmdlet is run, the cmdlet invokes the `Scripting agent` cmdlet extension agent. When this agent is invoked, the cmdlet checks whether any scripts are configured to be invoked by the cmdlet. If a script should be run for a cmdlet, the cmdlet tries to invoke any APIs defined in the script. The following APIs are available and are invoked in the following order:

1. **ProvisionDefaultProperties** This API can be used to set values of properties on objects when they're created. When you set a value, that value is returned to the cmdlet and the cmdlet sets the value on the property. You can fill in values on properties if the user didn't specify a value, or you can override the value specified by the user. This API respects the values set by higher priority agents. The `Scripting agent` cmdlet extension agent won't overwrite the values set by higher priority agents.
2. **UpdateAffectedIConfigurable** This API can be used to set values of properties on objects after all other processing has been completed, but the `Validate` API hasn't yet been invoked. This API respects the values set by higher priority agents. The `Scripting agent` cmdlet extension agent won't overwrite the values set by higher priority agents.
3. **Validate** This API can be used to validate the values on an object's properties that are about to be set by the cmdlet. This API is called just before a cmdlet writes any data. You can configure validation checks that allow a cmdlet to either succeed or fail. If a cmdlet passes the validation checks in this API, the cmdlet is allowed to write the data. If the cmdlet fails the validation checks, it returns any errors defined in this API.
4. **OnComplete** This API is used after all cmdlet processing is complete. It can be used to perform post-processing tasks, such as writing data to an external database.

Note:

The `Scripting agent` cmdlet extension agent isn't invoked when cmdlets with the `Get` verb are run.

Scripting agent configuration file

The `Scripting agent` configuration file contains all the scripts that you want the `Scripting agent` to run. Scripts in the configuration file are contained within XML tags that define the beginning and end of the script and various input parameters required to pass data to the script. Scripts are written using Windows PowerShell syntax. The configuration file is an XML file that uses the elements or attributes in the following table.

Attributes of Scripting agent configuration file

Element	Attribute	Description
`Configuration`	Not applicable	This element contains all the scripts that the `Scripting agent` cmdlet extension agent can run. The `Feature` tag is a child of this tag. There is only one `Configuration` tag in the configuration file.
`Feature`	Not applicable	This element contains a set of scripts that relate to a feature. Each script, defined in the `ApiCall` child tag, extends a specific part of the cmdlet execution pipeline. This tag contains the `Name` and `Cmdlets` attributes. There can be multiple `Feature` tags under the `Configuration` tag.
	`Name`	This attribute contains the name of the feature. Use this attribute to help identify which feature is extended by the scripts contained within the tag.
	`Cmdlets`	This attribute contains a list of the Exchange cmdlets used by the set of scripts in this feature extension. You can specify multiple cmdlets by separating each cmdlet with a comma.

`ApiCall`	Not applicable	This element contains scripts that can extend a part of the cmdlet execution pipeline. Each script is defined by the API call name in the cmdlet execution pipeline it's extending. The following are the API names that can be extended: • `ProvisionDefaultProperties` • `UpdateAffectedIConfigurable` • `Validate` • `OnComplete`
	`Name`	This attribute includes the name of the API call that's extending the cmdlet execution pipeline.
`Common`	Not applicable	This element contains functions that can be used by any script in the configuration file.

Every Exchange 2013 server includes the file ScriptingAgentConfig.xml.sample in the *<installation path>*\V15\Bin\CmdletExtensionAgents folder. This file must be renamed to ScriptingAgentConfig.xml on every Exchange 2013 server if you enable the Scripting Agent cmdlet extension agent. The sample configuration file contains sample scripts that you can use to help you understand how to add scripts to the configuration file.

After you add a script to the configuration file, or if you make a change to the configuration file, you must update the file on every Exchange 2013 server in your organization. This must be done to make sure that each server contains an up-to-date version of the scripts that the `Scripting Agent` cmdlet extension agent runs.

Some characters typically used in scripts also have a special meaning in XML. To use these characters in your script, use escape sequences. For example, the following characters use an escape sequence:

- Instead of a greater than sign (>), use `>`
- Instead of a less than sign (<), use `$lt;`
- Instead of an ampersand (&), use `&`

Enable the Scripting agent

The `Scripting agent` cmdlet extension agent is disabled by default. When you enable the `Scripting agent`, the agent is enabled for the entire Exchange 2013 organization. Before you enable the

`scripting agent`, verify that the `scripting agent` configuration file has been correctly renamed and updated with your scripts on every Exchange 2013 server. You will receive an error message each time a cmdlet runs if you don't rename the configuration file correctly or copy a configuration file to this computer from another Exchange 2013 server.

To enable the `scripting agent`, you must do the following:

1. Rename the ScriptingAgentConfig.xml.sample file in **<installation path>\V15\Bin\CmdletExtensionAgents** to ScriptingAgentConfig.xml on every Exchange 2013 server in your organization.

Note:

You can copy the configuration file from one Exchange 2013 server to other Exchange 2013 servers. Be sure you update the configuration file you want to copy before you copy it.

2. Add your script to the renamed configuration file on every Exchange 2013 server in your organization.
3. Enable the `scripting agent` cmdlet extension agent. For more information about enabling cmdlet extension agents, see Manage cmdlet extension agents.

Scripting Agent priority

By default, the `scripting agent` cmdlet extension agent runs after every other agent, with the exception of the `scripting agent` agent. If you want a script you created to replace an existing agent, you must either disable the other agent or change the priority of either agent so that the `scripting agent` cmdlet extension agent runs first. For more information about how to disable or change the priority of agents, see Manage cmdlet extension agents.

Manage cmdlet extension agents

Exchange Server 2013 > Exchange Management Shell > Cmdlet extension agents >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-11-19*

This topic shows you how to enable, disable, view, and change the priority of cmdlet extension agents in Microsoft Exchange Server 2013. For more information about cmdlet extension agents in Exchange 2013, see Cmdlet extension agents.

What do you need to know before you begin?

- Estimated time to complete each procedure: less than 5 minutes
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Cmdlet extension agents" entry in the Exchange and

Shell infrastructure permissions topic.

- Before you enable the `Scripting Agent`, you must verify that it's configured correctly. For more information about the `Scripting Agent`, see Cmdlet extension agents.
- You must use the Shell to perform these procedures.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

What do you want to do?

Enable a cmdlet extension agent

When you enable a cmdlet extension agent in Exchange 2013, the agent is run on every server running Exchange 2013 in the organization. When an agent is enabled, it's made available to cmdlets, which can then use the agent to perform additional operations.

Caution:

Before you enable an agent, be sure that you're aware of how the agent works and what impact the agent will have on your organization.

This example enables a cmdlet extension agent by using the **Enable-CmdletExtensionAgent** cmdlet. You must specify the name of the agent you want to enable when you run the cmdlet. Before you enable the `Scripting Agent`, you need to make sure that you've deployed the `ScriptingAgentConfig.xml` configuration file to all the servers in your organization. If you don't deploy the configuration file first and you enable the `Scripting Agent`, all non-**Get** cmdlets fail when they're run. This example enables the `Scripting Agent`.

```
Enable-CmdletExtensionAgent "Scripting Agent"
```

For detailed syntax and parameter information, see Enable-CmdletExtensionAgent.

Disable a cmdlet extension agent

When you disable a cmdlet extension agent in Exchange 2013, the agent is disabled on every server running Exchange 2013 in the organization. When an agent is disabled, it's not made available to cmdlets. Cmdlets can no longer use the agent to perform additional operations.

Caution:

Before you disable an agent, be sure that you're aware of how the agent works and what impact disabling the agent will have on your organization.

To disable a cmdlet extension agent, use the **Disable-CmdletExtensionAgent** cmdlet. Specify the

name of the agent you want to disable when you run the cmdlet. This example disables the `Scripting Agent`.

```
Disable-CmdletExtensionAgent "Scripting Agent"
```

For detailed syntax and parameter information, see Disable-CmdletExtensionAgent.

View existing cmdlet extension agents

Viewing cmdlet extension agents enables you to see which agents are run first and which agents are enabled in an Exchange 2013 organization. For more information about pipelining and the **Format-Table** cmdlet, see the following topics:

- Pipelining
- Working with command output

This example gets the details of a specific cmdlet extension agent by using the **Get-CmdletExtensionAgent** cmdlet. In this example, the details of the `Mailbox Permissions Agent` are returned.

```
Get-CmdletExtensionAgent "Mailbox Permissions Agent"
```

This example gets multiple cmdlet extension agents by using the **Get-CmdletExtensionAgent** cmdlet, and then pipes the output to the **Format-Table** cmdlet. This example displays a list of all of the cmdlet extension agents in the organization, and by using the **Format-Table** cmdlet, the **Name**, **Enabled**, and **Priority** properties of each agent are displayed in a table.

```
Get-CmdletExtensionAgent | Format-Table Name, Enabled,
Priority
```

For detailed syntax and parameter information, see Get-CmdletExtensionAgent.

Change the priority of a cmdlet extension agent

The ability to change the priority of a cmdlet extension agent in Exchange 2013 is useful when you want a certain agent to be called by a cmdlet before another agent. This is especially useful if you create a custom script that's run in the `Scripting Agent`, and you want that script to take precedence over a built-in agent. For more information about the `Scripting Agent`, see Cmdlet extension agents.

Caution:
Changing the priority or replacing the functionality of a built-in agent is an advanced operation. Be sure that you completely understand the changes you're making.

Agents are ordered from zero to the maximum number of agents. The closer to zero the agent is, the higher the priority of the agent. Agents with a higher priority are called first. For more

information about agent priorities, see Cmdlet extension agents.

This example changes the priority of a cmdlet extension agent by using the **Set-CmdletExtensionAgent** cmdlet. In this example, the priority of the `Scripting Agent` is changed to 3.

```
Set-CmdletExtensionAgent "Scripting Agent" -Priority 3
```

For detailed syntax and parameter information, see Set-CmdletExtensionAgent.

Exchange Management Shell quick reference for Exchange 2013

Exchange Server 2013 > Exchange Management Shell >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2012-10-03*

This topic describes the most frequently used cmdlets available in the release to manufacturing (RTM) and later versions of Microsoft Exchange Server 2013 and provides examples of their use.

Note:
More content will be added about other areas of Exchange 2013 soon.

For more information about the Exchange Management Shell in Exchange 2013 and all the available cmdlets, see the following topics:

- Exchange Management Shell
- Exchange 2013 cmdlets

What would you like to learn about?

Common cmdlet actions

The following verbs are supported by most cmdlets and are associated with a specific action.

New	The New verb creates an instance of something, such as a new configuration setting, a new database, or a new SMTP connector.
Remove	The Remove verb removes an instance of something, such as a mailbox or transport rule.

	All Remove cmdlets support the *WhatIf* and *Confirm* parameters. For more information about these parameters, see Important Parameters.	
Enable	The Enable verb enables a setting or mail-enables a recipient.	
Disable	The Disable verb disables an enabled setting or mail-disables a recipient. All Disable tasks also support the *WhatIf* and *Confirm* parameters. For more information about these parameters, see Important Parameters.	
Set	The Set verb modifies specific settings of an object, such as the alias of a contact or the deleted item retention of a mailbox database.	
Get	The Get verb queries a specific object or a subset of a type of object, such as a specific mailbox, all mailbox users, or mailbox users in a domain.	

Important parameters

The following parameters help you control how your commands run and indicate exactly what a command will do before it affects data.

Identity	The *Identity* parameter identifies the unique object for the task. It's typically used with Enable, Disable, Remove, Set, and Get cmdlets. *Identity* is also a positional parameter, which means that you don't have to specify *Identity* when you specify the parameter's value on the command line. For example, *Get-Mailbox -Identity user1* queries for the mailbox of *user1*. *Get-Mailbox*

	user1 is equivalent to *Get-Mailbox -Identity user1*.
WhatIf	The *WhatIf* parameter instructs the cmdlet to simulate the actions that it would take on the object. By using the *WhatIf* parameter, you can view what changes would occur without actually applying any of the changes. The default value is *$true*.
Confirm	The *Confirm* parameter causes the cmdlet to pause processing and requires the administrator to acknowledge what the cmdlet will do before processing continues. The default value is *$true*.
Validate	The *Validate* parameter causes the cmdlet to check that all prerequisites for running the operation are satisfied and that the operation will complete successfully.

Tips and tricks

The following commands are associated with various tasks that you can use when administering Exchange 2013.

Get-Command	This cmdlet retrieves all tasks that can be executed in Exchange 2013.
Get-Command **keyword**	This cmdlet retrieves tasks that have *keyword* in the cmdlet.
Get-*task* | Get-Member	This cmdlet retrieves all properties and methods of *task*.
Get-*task* | Format-List	This cmdlet displays the output of the query in a formatted list. You can pipe the output of any

	Get cmdlet to Format-List to view the whole set of properties that exist on the object returned by that command, or you can specify individual properties that you want to view, separated by commas, as in the following example: *Get-Mailbox *john* | Format-List alias,*quota*
Help *task*	This cmdlet retrieves Exchange Management Shell help information for any task in Exchange 2013, as in the following example: *Help Get-Mailbox*
Get-*task* | Format-List > *file.txt*	This cmdlet exports the output of *task* to a text file: *file.txt*

Permissions

Get-RoleGroupMember "*Organization Management*"	This command retrieves the members of the *Organization Management* management role group.
Get-ManagementRoleAssignment -Role "*Mail Recipient Creation*" -GetEffectiveUsers	This command retrieves a list of all the users who are granted permissions provided by the *Mail Recipient Creation* management role. This includes users who are members of role groups or universal security groups (USGs) that are assigned the Mail Recipient Creation role. This doesn't include users who are members of linked role groups in another forest.
Get-ManagementRoleAssignment -RoleAssignee *Administrator* | Get-ManagementRole | Get-ManagementRoleEntry	This command retrieves a list of cmdlets that the user *Administrator* can run.
ForEach ($RoleEntry in Get-ManagementRoleEntry *	This command retrieves a list of all the users who can run the *Remove-Mailbox* cmdlet.

Remove-Mailbox -parameters Identity) {Get-ManagementRoleAssignment -Role $RoleEntry.Role -GetEffectiveUsers -Delegating $False \| Where-Object {$_.EffectiveUserName -Ne "All Group Members"} | FL Role, EffectiveUserName, AssignmentChain}	
Get-ManagementRoleAssignment -WritableRecipient *kima* -GetEffectiveUsers | FT RoleAssigneeName, EffectiveUserName, Role, AssignmentChain	This command retrieves a list of all users who can modify the mailbox of *kima*.
New-ManagementScope "*Seattle Users*" -RecipientRestrictionFilter { *City* -Eq "*Seattle*" } New-RoleGroup "*Seattle Admins*" -Roles "*Mail Recipients*", "*Mail Recipient Creation*", "*Mailbox Import Export*", -CustomRecipientWriteScope "*Seattle Users*"	This command creates a new management scope and management role group to enable members of the role group to manage recipients in Seattle. First, the *Seattle Users* management scope is created, which matches only recipients who have *Seattle* in the *City* attribute on their user object. Then, a new role group called *Seattle Admins* is created and the *Mail Recipients*, *Mail Recipient Creation*, and *Mailbox Import Export* roles are assigned. The role group is scoped so that its members can manage only users who match the *Seattle Users* recipient filter scope.
New-ManagementScope "*Vancouver Servers*" -ServerRestrictionFilter	This command creates a new management scope and copies an existing role group to enable members of the new role group to manage only servers in the Vancouver Active

{ *ServerSite* -Eq "*Vancouver*" } $RoleGroup = Get-RoleGroup "*Server Management*" New-RoleGroup "*Vancouver Server Management*" -Roles $RoleGroup.Roles -CustomConfigWriteScope "*Vancouver Servers*"	Directory site. First, the *Vancouver Servers* management scope is created, which matches only servers that are located in the *Vancouver* Active Directory site. The Active Directory site is stored in the *ServerSite* attribute on the server objects. Then, a new role group called *Vancouver Server Management* is created that's a copy of the *Server Management* role group. This new role group, however, is scoped to allow its members to manage only servers that match the *Vancouver Servers* configuration filter scope.	
Add-RoleGroupMember "*Organization Management*" -Member *davids*	This command adds the user *davids* to the *Organization Management* role group.	
Get-ManagementRoleAssignment -Role "*Mail Recipient Creation*" -RoleAssignee "*Seattle Admins*" | Remove-ManagementRoleAssignment	This command removes the *Mail Recipient Creation* role from the *Seattle Admins* role group. This command is useful because you don't need to know the name of the management role assignment that assigns the role to the role group.	

Remote Shell

$Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri http://*ExServer.contoso.com*/PowerShell/ -Authentication Kerberos Import-PSSession $Session	These commands open a new remote Shell session between a local domain-joined computer and a remote Exchange 2013 server with the FQDN *ExServer.contoso.com*. Use this command if you want to administer a remote Exchange 2013 server and only have the Windows Management Framework, which

	includes the Windows PowerShell command-line interface, installed on your local computer. This command uses your current logon credentials to authenticate against the remote Exchange 2013 server.
$UserCredential = Get-Credential $Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri http://*ExServer.contoso.com*/PowerShell/ -Authentication Kerberos -Credential $UserCredential Import-PSSession $Session	These commands open a new remote Shell session between a local domain-joined computer and a remote Exchange 2013 server with the FQDN *ExServer.contoso.com*. Use this command if you want to administer a remote Exchange 2013 server and only have the Windows Management Framework, which includes Windows PowerShell, installed on your local computer. This command uses credentials you specify explicitly to authenticate against the remote Exchange 2013 server.
Remove-PSSession $Session	This command closes the remote Shell session between a local computer and the remote Exchange 2013 server.
Import-RecipientDataProperty -Identity "Tony Smith" -SpokenName -FileData *([Byte[]]$(Get-Content -Path "M:\AudioFiles\TonySmith.wma" -Encoding Byte -ReadCount 0))*	This command shows an example of the syntax, shown in italics, required to import a file into a remote Exchange 2013 server using the FileData parameter on a cmdlet. The syntax encapsulates the data contained in the *M:\AudioFiles\TonySmith.wma* file and streams the data to the FileData property on the Import-RecipientDataProperty cmdlet. The FileData parameter accepts data from a file on your local computer using this syntax on most cmdlets.
Export-RecipientDataProperty -Identity	This command shows an example of the syntax,

tony@contoso.com -SpokenName | *ForEach { $_.FileData | Add-Content C:\tonysmith.wma -Encoding Byte}*	shown in italics, required to export a file from a remote Exchange 2013 server. The syntax encapsulates the data stored in the FileData property on the object returned by the cmdlet and then streams the data to your local computer. The data is then stored in the *C:\tonysmith.wma* file. Most cmdlets that output objects with a FileData property use this syntax to export data to a file on your local computer.

Exchange 2013 cmdlets

Exchange Server 2013 > Exchange Management Shell >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-08*

Active Directory cmdlets

Anti-spam and anti-malware cmdlets

Client Access cmdlets

Cmdlet extension agent cmdlets

Email address and address book cmdlets

Federation and hybrid cmdlets

High availability cmdlets

Mail flow cmdlets

Mailbox cmdlets

Mailbox database cmdlets

Mailbox server cmdlets

Move and migration cmdlets

Organization cmdlets

Permissions cmdlets

Policy and compliance cmdlets

Security cmdlets

Server health, monitoring, and performance cmdlets

Sharing and collaboration cmdlets

Unified Messaging cmdlets

Users and Groups Cmdlets

Active Directory cmdlets

Exchange Server 2013 > Exchange Management Shell > Exchange 2013 cmdlets >

Topic Last Modified: *2014-04-28*

Get-AdServerSettings

Set-AdServerSettings

Get-AdSite

Set-AdSite

Get-AdSiteLink

Set-AdSiteLink

Get-DomainController

Get-OrganizationalUnit

Dump-ProvisioningCache

Reset-ProvisioningCache

Get-Trust

Get-UserPrincipalNamesSuffix

Get-AdServerSettings

Exchange Management Shell > Exchange 2013 cmdlets > Active Directory cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-AdServerSettings** cmdlet to view the Active Directory Domain Services (AD DS)

environment settings in the current Windows PowerShell session. The **Get-AdServerSettings** cmdlet replaces the `AdminsessionADSettings` session variable that was used in Microsoft Exchange Server 2007.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-AdServerSettings
```

Examples

EXAMPLE 1

This example displays the session settings for the current session.

```
Get-AdServerSettings | Format-List
```

For more information about pipelining and the **Format-List** cmdlet, see Pipelining and Working with command output.

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Active Directory Domain Services server settings" entry in the Exchange and Shell infrastructure permissions topic.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AdServerSettings

Exchange Management Shell > Exchange 2013 cmdlets > Active Directory cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-08*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-AdServerSettings** cmdlet to manage the Active Directory Domain Services (AD DS) environment in the current Exchange Management Shell session. The **Set-AdServerSettings** cmdlet replaces the *AdminSessionADSettings* session variable that was used in Microsoft Exchange Server 2007.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-AdServerSettings [-ConfigurationDomainController <Fqdn>] [-
PreferredGlobalCatalog <Fqdn>] [-RecipientViewRoot <String>] [-
SetPreferredDomainControllers <MultiValuedProperty>] [-ViewEntireForest
<$true | $false>] <COMMON PARAMETERS>


Set-AdServerSettings [-PreferredServer <Fqdn>] [-RecipientViewRoot
<String>] [-ViewEntireForest <$true | $false>] <COMMON PARAMETERS>


Set-AdServerSettings -RunspaceServerSettings
<RunspaceServerSettingsPresentationObject> <COMMON PARAMETERS>


Set-AdServerSettings -DisableGls <$true | $false> <COMMON PARAMETERS>


Set-AdServerSettings -ForceADInTemplateScope <$true | $false> <COMMON
PARAMETERS>


Set-AdServerSettings -ResetSettings <SwitchParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]] [-WriteOriginatingChangeTimestamp <$true | $false>]
[-WriteShadowProperties <$true | $false>]
```

Examples

EXAMPLE 1

This example sets the recipient scope to the Marketing Users OU in the contoso.com domain for the current session.

```
Set-AdServerSettings -RecipientViewRoot "contoso.com/
Marketing Users"
```

EXAMPLE 2

This example sets the scope of the current session to the entire forest and designates gc1.contoso.com as the preferred global catalog server.

```
Set-AdServerSettings -ViewEntireForest $true -
PreferredGlobalCatalog gc1.contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Active Directory Domain Services server settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DisableGls	Required	System.Boolean	This parameter is reserved for internal Microsoft use.
ForceADInTemplateScope	Required	System.Boolean	This parameter is reserved for internal Microsoft use.
ResetSettings	Required	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
RunspaceServerSettings	Required	Microsoft.Exchange.Data.Directory.Management.RunspaceServerSettingsPresentationObject	The *RunspaceServerSettings* parameter specifies whether to pass an entire configuration object to the command to be processed. This parameter is useful in scripts where an entire object must be passed to the command.
ConfigurationDomainC	Optional	Microsoft.Exchange.Dat	The

ontroller		a.Fqdn	*ConfigurationDomainController* parameter specifies the fully qualified domain name (FQDN) of the configuration domain controller to be used for reading Exchange configuration information in this session.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
PreferredGlobalCatalog	Optional	Microsoft.Exchange.Data.Fqdn	The *PreferredGlobalCatalog* parameter specifies the FQDN of the global catalog server to be used for reading recipient information in this session.
PreferredServer	Optional	Microsoft.Exchange.Dat	The *PreferredServer*

		a.Fqdn	parameter specifies the FQDN of the domain controller to be used for this session.
RecipientViewRoot	Optional	System.String	The *RecipientViewRoot* parameter specifies the organizational unit (OU) to include in the recipient scope for this session. When you specify a recipient scope with this parameter, only the recipients included in the scope are returned. To specify an OU, use the syntax *<FQDN of domain>/<OU tree>*.
SetPreferredDomainControllers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SetPreferredDomainControllers* parameter specifies the list of domain controllers used to read information from Active Directory in this session. You must specify the FQDN of the domain controllers. Separate multiple domain controllers using commas.

ViewEntireForest	Optional	System.Boolean	The *ViewEntireForest* parameter specifies whether all the objects in the forest are viewed and managed in this session. Valid values are `$true` and `$false`. When you specify a value of `$true`, the value stored in the *RecipientViewRoot* parameter is removed and all of the recipients in the forest can be viewed and managed.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WriteOriginatingChangeTimestamp	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.

WriteShadowProperties	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AdSite

Exchange Management Shell > Exchange 2013 cmdlets > Active Directory cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-AdSite** cmdlet to display configuration information about one or more Active Directory sites.

```
Get-AdSite [-Identity <AdSiteIdParameter>] [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example displays the configuration details for the Active Directory site named Default-First-Site-Name.

```
Get-AdSite Default-First-Site-Name
```

Detailed Description

Exchange uses Active Directory sites and the costs assigned to the Active Directory site links to

make message routing decisions.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport configuration" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.AdSiteIdParameter	The *Identity* parameter specifies the identity of the Active Directory site for which you want to view configuration details. The identity can be expressed as a GUID or the Active Directory

			site name. If the Active Directory site name includes spaces, enclose the name in quotation marks (").

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AdSite

Exchange Management Shell > Exchange 2013 cmdlets > Active Directory cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-AdSite** cmdlet to configure the Exchange settings of Active Directory sites.

```
Set-AdSite -Identity <AdSiteIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-HubSiteEnabled <$true | $false>] [-
InboundMailEnabled <$true | $false>] [-MinorPartnerId <Int32>] [-PartnerId
<Int32>] [-ResponsibleForSites <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures the Active Directory site named `Default-First-Site-Name` as a hub site.

```
Set-AdSite Default-First-Site-Name -HubSiteEnabled $true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport configuration" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AdSiteIdParameter	The *Identity* parameter specifies the identity of the Active Directory site you want to modify. You can use any value that uniquely identifies the site. For example, you can use the name, GUID or distinguished name (DN) of the Active Directory site.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Dat	The *DomainController*

		a.Fqdn	parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
HubSiteEnabled	Optional	System.Boolean	The *HubSiteEnabled* parameter specifies whether this site acts as a hub site. The default value is `$false`.
InboundMailEnabled	Optional	System.Boolean	The *InboundMailEnabled* parameter enables or disables receiving incoming messages for all the Exchange located in the Active Directory site. Typically, this parameter is used after Active Directory site failover or maintenance. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. If you set the value to `$false`, none of the Exchange servers in the Active Directory site will be

			able to receive incoming messages.
MinorPartnerId	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
PartnerId	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
ResponsibleForSites	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AdSiteLink

Exchange Management Shell > Exchange 2013 cmdlets > Active Directory cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-AdSiteLink** cmdlet to view configuration information about an Active Directory IP site link.

```
Get-AdSiteLink [-Identity <AdSiteLinkIdParameter>] [-DomainController
<Fqdn>]
```

Examples

EXAMPLE 1

This example returns a list of all IP site links in your organization.

```
Get-ADSiteLink
```

EXAMPLE 2

This example returns a list of all IP site links in your organization that have a specific Exchange cost assigned.

```
Get-AdSiteLink | Where {$_.ExchangeCost -ne $null}
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport configuration" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.AdSiteLinkIdParameter	The *Identity* parameter specifies the name or GUID of the IP site link for which you want to view configuration information.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AdSiteLink

Exchange Management Shell > Exchange 2013 cmdlets > Active Directory cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-AdSiteLink** cmdlet to assign an Exchange-specific cost to an Active Directory IP site link. You can also use this cmdlet to configure the maximum message size that can pass across an Active Directory IP site link.

```
Set-AdSiteLink -Identity <AdSiteLinkIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ExchangeCost <Int32>] [-
MaxMessageSize <Unlimited>] [-Name <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example assigns an Exchange-specific cost of 25 to the IP site link DEFAULT_IP_SITE_LINK and configures a maximum message size limit of 10 MB on the IP site link.

```
Set-AdSiteLink DEFAULT_IP_SITE_LINK -ExchangeCost 25 -
MaxMessageSize 10MB
```

Detailed Description

By default, Microsoft Exchange determines the least cost routing path by using the cost assigned to the Active Directory IP site link. You can use the **Set-AdSiteLink** cmdlet to assign an Exchange-specific cost to the Active Directory IP site link. The Exchange-specific cost is a separate attribute used instead of the Active Directory-assigned cost to determine the least cost routing path.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport configuration" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.AdSiteLinkIdParameter	specifies the name or GUID of the IP site link you want to modify.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

ExchangeCost	Optional	System.Int32	The *ExchangeCost* parameter specifies an Exchange-specific cost for the IP site link. This cost is used instead of the Active Directory-assigned cost. To clear the value of the *ExchangeCost* parameter and revert to using the cost of the IP site link specified in Active Directory, set the value of the *ExchangeCost* parameter to `$null`.
MaxMessageSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxMessageSize* parameter specifies the maximum size of a message that can pass across the Active Directory IP site link. The default value is `unlimited`. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from `64`

			KB through Int64. To remove the message size limit on an Active Directory IP site link, enter a value of unlimited.
Name	Optional	System.String	The *Name* parameter specifies a unique name for the IP site link. The name that you assign overwrites the current identity of the IP site link.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DomainController

Exchange Management Shell > Exchange 2013 cmdlets > Active Directory cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-DomainController** cmdlet to view a list of domain controllers that exist in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-DomainController [-Forest <Fqdn>] [-GlobalCatalog <SwitchParameter>]
<COMMON PARAMETERS>

Get-DomainController [-DomainName <Fqdn>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-Credential <NetworkCredential>]
```

Examples

EXAMPLE 1

This example retrieves a list of global catalog servers in the corp.contoso.com domain.

Because a different set of credentials are required to access this domain, the **Get-Credential** cmdlet is used to obtain the user name and password from the user. The **Get-Credential** cmdlet displays a prompt to the user that accepts the user name and password. The credentials are then stored in the `$UserCredentials` variable.

```
$UserCredentials = Get-Credential
```

The `$UserCredentials` variable is then passed to the *Credential* parameter in the **Get-DomainController** command. To make the list more readable, the output is piped to the **Format-Table** cmdlet and only the `Name` and `ADSite` properties are displayed.

```
Get-DomainController -DomainName corp.contoso.com -
Credential $UserCredentials | Format-Table -AutoSize Name,
ADSite
```

For more information about pipelining and the **Format-Table** cmdlet, see Pipelining and Working with command output.

Detailed Description

The **Get-DomainController** cmdlet is used by the Exchange Administration Center in Microsoft Exchange Server 2013 to populate fields that display domain controller information.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Active Directory Domain Services server settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Credential	Optional	System.Net.NetworkCredential	The *Credential* parameter specifies the credentials to use to access the domain specified if the *DomainName* parameter is used. If the *Forest* parameter is used, the credentials are used to access the forest. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.

DomainName	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainName* parameter specifies the fully qualified domain name (FQDN) of the domain for which you want to return a list of domain controllers.
Forest	Optional	Microsoft.Exchange.Data.Fqdn	The *Forest* parameter specifies the FQDN of the root domain of the forest for which you want to return a list of domain controllers.
GlobalCatalog	Optional	System.Management.Automation.SwitchParameter	The *GlobalCatalog* switch specifies whether the command should return a list of global catalog servers.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-OrganizationalUnit

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-OrganizationalUnit** cmdlet to view a list of organizational units (OUs) that exist in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-OrganizationalUnit [-Identity <ExtendedOrganizationalUnitIdParameter>]
[-SingleNodeOnly <SwitchParameter>] <COMMON PARAMETERS>


Get-OrganizationalUnit [-SearchText <String>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-IncludeContainers
<SwitchParameter>] [-Organization <OrganizationIdParameter>] [-ResultSize
<Unlimited>]
```

Examples

EXAMPLE 1

This example retrieves a list of the first level child OUs beneath the North America OU and formats the output so that only the **Name** and **DistinguishedName** properties are displayed.

```
Get-OrganizationalUnit "North America" -SingleNodeOnly |
Format-Table Name, DistinguishedName
```

For more information about pipelining and the **Format-Table** cmdlet, see Pipelining and Working with command output.

EXAMPLE 2

This example retrieves a list of OUs that match the text string "Executives" and formats the output so that only the **Name** and **DistinguishedName** properties are displayed.

```
Get-OrganizationalUnit -SearchText "Executives" | Format-
Table Name, DistinguishedName
```

For more information about pipelining and the **Format-Table** cmdlet, see Pipelining and Working with command output.

Detailed Description

The **Get-OrganizationalUnit** cmdlet is used by the Exchange Administration Center in Microsoft Exchange Server 2013 to populate fields that display OU information.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Active Directory Domain Services server settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ExtendedOrganizationalUnitIdParameter	The *Identity* parameter specifies the OU to retrieve. If the OU name contains spaces, enclose the value in quotation marks ("). This parameter can't be used with the *SearchText* parameter.
IncludeContainers	Optional	System.Management.Automation.SwitchParameter	The *IncludeContainers* switch instructs the command to return containers while

			returning a list of OUs.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
SearchText	Optional	System.String	The *SearchText* parameter enables you to search the names of all OUs in your organization for the specified string. Only the OUs that match the string you specify are returned. If the string you specify contains spaces, enclose it in quotation marks ("). This parameter can't be used with the *Identity* parameter.
SingleNodeOnly	Optional	System.Management.A	The *SingleNodeOnly*

		utomation.SwitchParameter	switch instructs the command to return only the first level child OUs beneath the OU specified in the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Dump-ProvisioningCache

Exchange Management Shell > Exchange 2013 cmdlets > Active Directory cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Dump-ProvisioningCache** cmdlet to return a list of the cached keys and values for the specified server and Windows PowerShell application.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Dump-ProvisioningCache -Application <String> -GlobalCache
<SwitchParameter> -Server <Fqdn> <COMMON PARAMETERS>


Dump-ProvisioningCache -Application <String> -Server <Fqdn> [-
CurrentOrganization <SwitchParameter>] [-Organizations
<MultiValuedProperty>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-CacheKeys <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example displays all cache keys for the specified server and Windows PowerShell application.

```
Dump-ProvisioningCache -Server EXSRV1.contoso.com -
Application Powershell-Proxy -GlobalCache
```

Detailed Description

The **Dump-ProvisioningCache** cmdlet is for diagnostic purposes only and is rarely used. Exchange administrators or Microsoft support personnel may need to run this cmdlet to troubleshoot problems resulting from incorrect links or properties stamped on newly provisioned recipients, which can be caused by stale data in the provisioning cache.

The **Dump-ProvisioningCache** cmdlet displays a list of the Windows PowerShell provisioning cache keys. Use the value of these cache keys with the **Reset-ProvisioningCache** cmdlet to reset provisioning cache data.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange server configuration settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Application	Required	System.String	The *Application* parameter specifies the specific administrative application to reset the provisioning cache for. You can use the following values: • `Powershell` • `Powershell-LiveId` • `Powershell-Proxy` • `PowershellLiveId-Proxy` • `Ecp` • `Psws`
GlobalCache	Required	System.Management.Automation.SwitchPar	The *GlobalCache* switch specifies that all cache

		ameter	keys are cleared.
Server	Required	Microsoft.Exchange.Data.Fqdn	The *Server* parameter specifies the fully qualified domain name (FQDN) of the server that the application you want to reset is running on.
CacheKeys	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *CacheKeys* parameter specifies the value for the cache key that you want to clear. The format for the values should contain 32 digits separated by four dashes: `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` Use the **Dump-ProvisioningCache** cmdlet to return a list of cache keys.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
CurrentOrganization	Optional	System.Management.	The *CurrentOrganization*

		Automation.SwitchParameter	switch specifies that the provision cache is reset for this organization.
Organizations	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Organizations* parameter specifies the organizations that the provisioning cache will be reset. This parameter is used in multi-tenant deployments.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Reset-ProvisioningCache

Exchange Management Shell > Exchange 2013 cmdlets > Active Directory cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

The **Reset-ProvisioningCache** cmdlet clears the Windows PowerShell provisioning cache of frequently used Active Directory objects. This cmdlet is only used for diagnostic purposes.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Reset-ProvisioningCache -Application <String> -GlobalCache
<SwitchParameter> -Server <Fqdn> <COMMON PARAMETERS>


Reset-ProvisioningCache -Application <String> -Server <Fqdn> [-
CurrentOrganization <SwitchParameter>] [-Organizations
<MultiValuedProperty>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-CacheKeys <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resets the provisioning cache for Windows PowerShell running on the server EXSRV1.contoso.com in an on-premises Exchange organization and specifies that all cache keys are cleared.

```
Reset-ProvisioningCache -Server EXSRV1.contoso.com -
Application Powershell -GlobalCache
```

EXAMPLE 2

This example runs in a multi-tenant deployment by a data center administrator to reset the provisioning cache for Windows PowerShell for the adatum.com tenant and clear all cache keys.

```
Reset-ProvisioningCache -Application Powershell-Proxy -
Server datacenter1.adatum.com -GlobalCache
```

Detailed Description

The **Reset-ProvisioningCache** cmdlet is for diagnostic purposes only and is rarely used. Exchange

administrators need to run this cmdlet only if incorrect links or properties are stamped on newly provisioned recipients, which can be caused by stale data in the provisioning cache. This is a rare occurrence because the provisioning cache has invalidation notification logic.

The **Reset-ProvisioningCache** cmdlet clears the Windows PowerShell provisioning cache of frequently used Active Directory objects. To reduce Active Directory requests, a provisioning cache is initialized in each Windows PowerShell runspace and is used to cache common objects that are frequently used by cmdlets and provisioning handlers. During Exchange cmdlet execution, the provisioning cache loads configuration objects from Active Directory to help run a cmdlet. For example, when you create a mailbox, the **New-Mailbox** cmdlet obtains properties from Active Directory. When running cmdlets, configuration objects such as database containers, administrative role groups, and LegacyDNs are retrieved from Active Directory. Because these types of objects are stable and don't change for months or years after they're created, they're stored in the provisioning cache used by Windows PowerShell. This increases provisioning efficiency and significantly improves cmdlet performance.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange server configuration settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Application	Required	System.String	The *Application* parameter specifies the specific administrative application to reset the provisioning cache for. You can use the following values: • `Powershell` • `Powershell-LiveId` • `Powershell-Proxy` • `PowershellLiveId-Proxy` • `Ecp` • `Psws`
GlobalCache	Required	System.Management.Automation.SwitchParameter	The *GlobalCache* switch specifies that all cache keys are cleared.

Server	Required	Microsoft.Exchange.Data.Fqdn	The *Server* parameter specifies the fully qualified domain name (FQDN) of the server that the application you want to reset is running on.
CacheKeys	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *CacheKeys* parameter specifies the value for the cache key that you want to clear. The format for the values should contain 32 digits separated by four dashes: `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` Use the **Dump-ProvisioningCache** cmdlet to return a list of cache keys.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
CurrentOrganization	Optional	System.Management.Automation.SwitchParameter	The *CurrentOrganization* switch specifies that the provision cache is reset

			for this organization.
Organizations	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Organizations* parameter specifies the organizations that the provisioning cache will be reset. This parameter is used in multi-tenant deployments.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-Trust

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-Trust** cmdlet to return external and forest trusts.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-Trust [-DomainName <Fqdn>]
```

Examples

EXAMPLE 1

This example enumerates all trusts for the domain Contoso.com.

```
Get-Trust -DomainName Contoso.com
```

Detailed Description

The **Get-Trust** cmdlet is used by the Exchange Administration Center in Microsoft Exchange Server 2013 to populate fields that display recipient information.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Active Directory Domain Services server settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainName	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainName* parameter specifies that trusts returned are restricted to the domain name specified.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UserPrincipalNamesSuffix

Exchange Management Shell > Exchange 2013 cmdlets > Active Directory cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-UserPrincipalNamesSuffix** cmdlet to view the user principal name (UPN) suffixes in the Active Directory forest. The UPN suffixes are created in Active Directory Domains and Trusts.

```
Get-UserPrincipalNamesSuffix <COMMON PARAMETERS>

Get-UserPrincipalNamesSuffix [-Identity
<ExtendedOrganizationalUnitIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<ExtendedOrganizationalUnitIdParameter>]
```

Examples

EXAMPLE 1

This example returns all UPN suffixes for the Active Directory forest.

```
Get-UserPrincipalNamesSuffix
```

Detailed Description

UPN suffixes assigned to an organizational unit are stored in the **upnSuffixes** attribute in the **Organizational Unit** object.

The default UPN is contained in the **Canonical Name** attribute on the **Partitions container** object in the configuration naming context. The default UPN suffix identifies the domain in which the user

account is contained. When you create a user account in Active Directory, the default UPN suffix is the DNS name of the first domain in your domain tree.

If you create user accounts by using Active Directory Users and Computers, every user must have a UPN.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient provisioning permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ExtendedOrganizationalUnitIdParameter	The *Identity* parameter specifies the OU to retrieve. If the OU name contains spaces, enclose the value in quotation marks (").
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.ExtendedOrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies the container where you search for all available

			UPN suffixes.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Anti-spam and anti-malware cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-07*

Anti-Malware cmdlets

Get-MalwareFilteringServer

Set-MalwareFilteringServer

Get-MalwareFilterPolicy

New-MalwareFilterPolicy

Remove-MalwareFilterPolicy

Set-MalwareFilterPolicy

Disable-MalwareFilterRule

Enable-MalwareFilterRule

Get-MalwareFilterRule

New-MalwareFilterRule

Remove-MalwareFilterRule

Set-MalwareFilterRule

Anti-Spam cmdlets

Anti-Spam agent logging

Get-AgentLog

Attachment Filtering cmdlets on Edge Transport servers

Add-AttachmentFilterEntry

Get-AttachmentFilterEntry

Remove-AttachmentFilterEntry

Get-AttachmentFilterListConfig

Set-AttachmentFilterListConfig

Connection Filtering cmdlets on Edge Transport servers

IP Allow List cmdlets

Get-IPAllowListConfig

Set-IPAllowListConfig

Add-IPAllowListEntry

Get-IPAllowListEntry

Remove-IPAllowListEntry

IP Allow List Provider cmdlets

Add-IPAllowListProvider

Get-IPAllowListProvider

Remove-IPAllowListProvider

Set-IPAllowListProvider

Test-IPAllowListProvider

Get-IPAllowListProvidersConfig

Set-IPAllowListProvidersConfig

IP Block List cmdlets

Get-IPBlockListConfig

Set-IPBlockListConfig

Add-IPBlockListEntry

Get-IPBlockListEntry

Remove-IPBlockListEntry

IP Block List Provider cmdlets

Add-IPBlockListProvider

Get-IPBlockListProvider

Remove-IPBlockListProvider

Set-IPBlockListProvider

Test-IPBlockListProvider

Get-IPBlockListProvidersConfig

Set-IPBlockListProvidersConfig

Content Filtering cmdlets

Get-ContentFilterConfig

Set-ContentFilterConfig

Add-ContentFilterPhrase

Get-ContentFilterPhrase

Remove-ContentFilterPhrase

Update-SafeList

Mailbox Junk Email configuration cmdlets

Get-MailboxJunkEmailConfiguration

Set-MailboxJunkEmailConfiguration

Recipient Filtering cmdlets on Edge Transport servers

Get-RecipientFilterConfig

Set-RecipientFilterConfig

Sender Filtering cmdlets

Get-SenderFilterConfig

Set-SenderFilterConfig

Sender ID cmdlets

Test-SenderId

Get-SenderIdConfig

Set-SenderIdConfig

Sender Reputation cmdlets

Get-SenderReputationConfig

Set-SenderReputationConfig

Get-AgentLog

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-AgentLog** cmdlet to parse log files that you specify as parameters and collect raw statistics from the filtering that anti-spam agents apply during a time period that you specify.

```
Get-AgentLog [-EndDate <DateTime>] [-Location <LocalLongFullPath>] [-
StartDate <DateTime>] [-TransportService <Hub | Edge | FrontEnd |
MailboxSubmission | MailboxDelivery>]
```

Examples

EXAMPLE 1

This example returns a report that has statistics collected between 09:00 (9 A.M.), January 4, 2010 and 18:00 (6 P.M.), January 8, 2010 in the Front End Transport service on a Client Access server.

```
Get-AgentLog -StartDate "01/04/2010 9:00:00 AM" -EndDate
"01/08/2010 6:00:00 PM" -TransportService FrontEnd
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
EndDate	Optional	System.DateTime	The *EndDate* parameter specifies the date and time that you want to stop collecting statistics. The default time is the current time. When you enter a specific date, use the short date format defined in the **Regional Options** settings configured on the local computer. For example, if your computer is configured to use the short date format *mm/dd/yyyy*, enter 03/01/2010 to specify March 1, 2010. You can enter the date only, for example, `10/05/2010`. Or you can enter the date and time of day. If you enter a time of day and date, you must enclose the argument in quotation marks ("), for example, `"10/05/2010 5:00:00 PM"`.

Location	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *Location* parameter specifies the directory that contains the log files that you can use to build usage reports. The default path is `%ExchangeInstallPath%TransportRoles\Logs\AgentLog`. You must enclose the file path in quotation marks (").
StartDate	Optional	System.DateTime	The *StartDate* parameter specifies the date and time that the collection of statistics starts. The default time is the current time. When you enter a specific date, use the short date format defined in the **Regional Options** settings configured on the local computer. For example, if your computer is configured to use the short date format *mm/dd/yyyy*, enter 03/01/2010 to specify March 1, 2010. You can enter the date only, for example, `10/05/2010`, or you can enter the date and time of day. If you enter a time of day and date, you must

			enclose the argument in quotation marks ("), for example, `"10/05/2010 5:00:00 PM"`.
TransportService	Optional	Microsoft.Exchange.Data.TransportService	The *TransportService* parameter specifies the transport service that you want to view or modify. Valid values for this parameter are: • `Hub` for the Transport service on Mailbox servers. • `MailboxSubmission` for the Mailbox Transport Submission service on Mailbox servers. • `MailboxDelivery` for the Mailbox Transport Delivery service on Mailbox servers. • `FrontEnd` for the Front End Transport service on Client Access servers. • `Edge` on Edge Transport servers.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-AttachmentFilterEntry

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Add-AttachmentFilterEntry** cmdlet to add an entry to the attachment filter list that's used by the Attachment Filtering agent on Edge Transport servers.

```
Add-Attachmentfilterentry -Name <String> -Type <ContentType | FileName> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds an attachment filter entry based on a file name. After running this command, the Attachment Filtering agent filters all attachments that have a .txt extension.

```
Add-AttachmentFilterEntry -Name *.txt -Type FileName
```

EXAMPLE 2

This example adds an attachment filter entry based on the MIME content type `image/jpeg`, which is a JPEG image binary file. After running this command, the Attachment Filtering agent filters all attachments of the MIME content type `image/jpeg`.

```
Add-AttachmentFilterEntry -Name image/jpeg -Type
ContentType
```

Detailed Description

On Edge Transport servers, the Attachment Filtering agent blocks attachments in messages based on the content type and the file name of the attachment. The configuration of the Attachment Filtering agent determines how messages that contain the specified attachments are processed. For more information about how to configure the Attachment Filtering agent, see Set-AttachmentFilterListConfig.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the MIME content type or file name of the attachment. If the *Type* parameter is set to `FileName`, the *Name* parameter can take any exact file name, such as `BadFile.exe`, or file name extension, such as `*.exe`. If the *Type* parameter is set to `ContentType`, the *Name* parameter can take any valid MIME content type.
Type	Required	Microsoft.Exchange.Data.Directory.SystemConfiguration.AttachmentType	The *Type* parameter specifies what type of attachment the attachment filter entry blocks. Valid values are `ContentType` and `FileName`: • `ContentType` This value matches the attachment filter entry against the MIME content type specified in the *Name* parameter.

			• `FileName` This value matches the attachment filter entry against the simple file name specified in the *Name* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AttachmentFilterEntry

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-AttachmentFilterEntry** cmdlet to view the list of attachment filter entries that are used by the Attachment Filtering agent on Edge Transport servers.

```
Get-Attachmentfilterentry [-DomainController <Fqdn>] [-Identity <String>]
```

Examples

EXAMPLE 1

This example returns all attachment filter entries.

```
Get-AttachmentFilterEntry
```

EXAMPLE 2

This example returns only the attachment filter entries that filter file names with a .txt extension.

```
Get-AttachmentFilterEntry FileName:*.txt
```

EXAMPLE 3

This example returns only the attachment filter entries that filter attachments that have the MIME content type `image/jpeg`.

```
Get-AttachmentFilterEntry ContentType:image/jpeg
```

Detailed Description

On Edge Transport servers, the Attachment Filtering agent blocks attachments in messages based on the content type and the file name of the attachment. The configuration of the Attachment Filtering agent determines how messages that contain the specified attachments are processed. For more information about how to configure the Attachment Filtering agent, see Set-AttachmentFilterListConfig.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	System.String	The *Identity* parameter specifies which attachment filter entry the command retrieves. The *Identity* parameter accepts values in the format `Type:Name`, where `Type` is one of the following values: • `ContentType` This value matches the attachment filter entry against the MIME content type. • `FileName` This value matches the attachment filter entry against the simple file name.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-AttachmentFilterEntry

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Remove-AttachmentFilterEntry** cmdlet to remove an entry from the attachment filter list that's used by the Attachment Filtering agent on Edge Transport servers.

```
Remove-Attachmentfilterentry -Identity <String> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the attachment filter entry that filters file names with a .txt extension.

```
Remove-AttachmentFilterEntry -Identity FileName:*.txt
```

EXAMPLE 2

This example removes the attachment filter entry that filters attachments that have the MIME content type image/jpeg.

```
Remove-AttachmentFilterEntry -Identity ContentType:image/
jpeg
```

Detailed Description

On Edge Transport servers, the Attachment Filtering agent blocks attachments in messages based on the content type and the file name of the attachment. The configuration of the Attachment

Filtering agent determines how messages that contain the specified attachments are processed. For more information about how to configure the Attachment Filtering agent, see Set-AttachmentFilterListConfig.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	System.String	The *Identity* parameter specifies the type of attachment that this filter entry removes. The *Identity* parameter accepts values in the format `Type:Name`, where `Type` is one of the following two values: • `ContentType` This value matches the attachment filter entry against the MIME content type. • `FileName` This value matches the attachment filter entry against the simple file name. In `Type:Name`, `Name` can be either the file name of the attachment filter entry to be removed, or the content type of the attachment filter entry to be removed.
Confirm	Optional	System.Management.	The *Confirm* switch can be

		Automation.SwitchParameter	used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AttachmentFilterListConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-AttachmentFilterListConfig** cmdlet to view the configuration of the Attachment Filtering agent on Edge Transport servers.

```
Get-Attachmentfilterlistconfig [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the Attachment Filtering agent configuration for

the computer on which the command is run.

```
Get-AttachmentFilterListConfig | Format-List
```

Detailed Description

On Edge Transport servers, the Attachment Filtering agent blocks attachments in messages based on the content type or the file name of the attachment. The configuration of the Attachment Filtering agent determines how messages that contain the specified attachments are processed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AttachmentFilterListConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Set-AttachmentFilterListConfig** cmdlet to modify the configuration of the Attachment Filtering agent on Edge Transport servers.

```
Set-Attachmentfilterlistconfig [-Action <Reject | Strip | SilentDelete>]
[-AdminMessage <String>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-ExceptionConnectors <MultiValuedProperty>] [-RejectResponse
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the action that the Attachment Filtering agent takes on an attachment that matches an entry on the attachment filter list so that both the email message and attachment aren't delivered to the recipient, and an NDR is sent to the sender.

```
Set-AttachmentFilterListConfig -Action Reject
```

Detailed Description

On Edge Transport servers, the Attachment Filtering agent blocks attachments in messages based

on the content type or the file name of the attachment. The configuration of the Attachment Filtering agent determines how messages that contain the specified attachments are processed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Action	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.FilterActions	The *Action* parameter specifies how the Attachment Filtering agent handles an attachment that matches an entry on the attachment filter list. The default value is `Reject`. You can use one of the following values: • `Reject` This value prevents both the email message and attachment from being delivered to the recipient and issues a non-delivery report (NDR) to the sender. • `Strip` This value removes the offending attachment from the email message and allows the message and other attachments that don't match an entry on the attachment filter list through. A notification that the attachment was removed is added to the

			email message. • `SilentDelete` This value prevents both the email message and the attachment from being delivered to the recipient. No notification that the email message and attachment were blocked is sent to the sender.
AdminMessage	Optional	System.String	The *AdminMessage* parameter specifies the content of a text file that replaces attachments removed by the Attachment Filtering agent. The *AdminMessage* parameter only appears when the Attachment Filtering agent is configured to remove an attachment that's been identified as bad.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
ExceptionConnectors	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptionConnectors* parameter specifies a list of connectors that should be excluded from attachment filtering. Attachment filters aren't applied to email messages received through these connectors. You must use the connector GUID to specify the *ExceptionConnectors* parameter value.
RejectResponse	Optional	System.String	The *RejectResponse*

			parameter specifies the message body that you want delivered in the NDR to senders whose messages contain an attachment that's blocked. The *RejectResponse* parameter is required only if you set the *Action* parameter to `Reject`. Don't exceed 240 characters in the parameter argument. When you pass an argument, you must enclose the *RejectResponse* parameter value in quotation marks (") if the phrase contains spaces, for example: "`Message rejected`". The default setting is `Message rejected due to unacceptable attachments`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply

			any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ContentFilterConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ContentFilterConfig** cmdlet to view the content filter configuration for the computer on which the cmdlet is run.

```
Get-ContentFilterConfig [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the content filter configuration for the computer on which the command is run.

```
Get-ContentFilterConfig | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ContentFilterConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ContentFilterConfig** cmdlet to modify the content filter configuration on a Mailbox server or an Edge Transport server.

```
Set-ContentFilterConfig [-BypassedRecipients <MultiValuedProperty>] [-
BypassedSenderDomains <MultiValuedProperty>] [-BypassedSenders
<MultiValuedProperty>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Enabled <$true | $false>] [-ExternalMailEnabled <$true |
$false>] [-InternalMailEnabled <$true | $false>] [-
OutlookEmailPostmarkValidationEnabled <$true | $false>] [-
QuarantineMailbox <SmtpAddress>] [-RejectionResponse <AsciiString>] [-
SCLDeleteEnabled <$true | $false>] [-SCLDeleteThreshold <Int32>] [-
SCLQuarantineEnabled <$true | $false>] [-SCLQuarantineThreshold <Int32>]
[-SCLRejectEnabled <$true | $false>] [-SCLRejectThreshold <Int32>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example specifies the sender domain woodgrovebank.com as a bypassed domain. Messages received from that domain bypass the Content Filter agent.

```
Set-ContentFilterConfig -BypassedSenderDomains
woodgrovebank.com
```

EXAMPLE 2

This example makes the following modifications to the Content Filter agent configuration:

- It enables and configures the SCL threshold functionalities that quarantine, reject, and delete messages to 5, 6, and 8 respectively.
- It specifies SpamQuarantineMailbox@contoso.com as the spam quarantine mailbox.
- It defines two users for whom the Content Filter won't process messages.

```
Set-ContentFilterConfig -SCLQuarantineEnabled $true -
```

```
SCLRejectEnabled $true -SCLDeleteEnabled $true -
SCLQuarantineThreshold 5 -SCLRejectThreshold 6 -
SCLDeleteThreshold 8 -QuarantineMailbox
SpamQuarantineMailbox@contoso.com -RejectionResponse
"Message rejected due to content restrictions" -
BypassedRecipients user1@contoso.com,user2@contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
BypassedRecipients	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassedRecipients* parameter specifies the SMTP address values of recipients in your organization. The Content Filter agent doesn't process any content filtering for messages bound to the addresses listed on this parameter. To enter multiple SMTP addresses, separate the addresses by using a comma, for example: `recipient1@contoso.com,recipient2@contoso.com`. The maximum number of recipients you can input is 100.

BypassedSenderDomains	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassedSenderDomains* parameter specifies domain name values of sending domains. The Content Filter agent doesn't process any content filtering for messages received from the domains listed on this parameter. To enter multiple domains, separate the addresses by using a comma, for example: `contoso.com,example.com`. A wildcard character (*) can be used to specify all subdomains, for example: `*.contoso.com`. The maximum number of domains you can input is 100.
BypassedSenders	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassedSenders* parameter specifies the SMTP address values of senders. The Content Filter agent doesn't process any content filtering for messages

			received from the addresses listed on this parameter. To enter multiple SMTP addresses, separate the addresses by using a comma, for example: `sender1@contoso.com, sender2@example.com`. The maximum number of SMTP addresses you can input is 100.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController*

			parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter enables or disables the Content Filter agent on the computer on which you're running the command. Valid input for the *Enabled* parameter is `$true` or `$false`. The default setting is `$true`.
ExternalMailEnabled	Optional	System.Boolean	The *ExternalMailEnabled* parameter specifies whether all messages from unauthenticated connections from sources external to your Exchange organization are passed through the Content Filter agent for processing. Valid input for the

			ExternalMailEnabled parameter is `$true` or `$false`. The default setting is `$true`. When the *ExternalMailEnabled* parameter is set to `$true`, all messages from unauthenticated connections are passed through the Content Filter agent for processing.
InternalMailEnabled	Optional	System.Boolean	The *InternalMailEnabled* parameter specifies whether all messages from authenticated connections and from authoritative domains in your enterprise are passed through the Content Filter agent for processing. Valid input for the *InternalMailEnabled* parameter is `$true` or `$false`. The default setting is `$false`. When the *InternalMailEnabled* parameter is set to

			`$true`, all messages from authenticated connections and from authoritative domains in your enterprise are passed through the Content Filter agent for processing.
OutlookEmailPostmarkValidationEnabled	Optional	System.Boolean	The *OutlookEmailPostmarkValidationEnabled* parameter specifies whether the Content Filter agent sends a computational puzzle to the sender's e-mail system for processing. Valid input for the *OutlookEmailPostmarkValidationEnabled* parameter is `$true` or `$false`. When the *OutlookEmailPostmarkValidationEnabled* parameter is set to `$true`, the Content Filter agent sends a computational puzzle to the sender's e-mail system for processing. The results of the puzzle validation are

			factored into the overall spam confidence level (SCL). This functionality is exposed to the Microsoft Outlook user as Office Outlook 2007 E-mail Postmark validation. The default setting is `$false`.
QuarantineMailbox	Optional	Microsoft.Exchange.Data.SmtpAddress	The *QuarantineMailbox* parameter specifies an SMTP address to be used as a spam quarantine mailbox. A spam quarantine mailbox is required when you set the *SCLQuarantineEnabled* parameter to `$true`. All messages that exceed the value set in the *SCLQuarantineThreshold* parameter are sent to the SMTP address that you set in this parameter.
RejectionResponse	Optional	Microsoft.Exchange.Data.AsciiString	The *RejectionResponse* parameter specifies the message body that you want delivered in the non-delivery report

			(NDR) to senders whose messages meet or exceed the *SCLRejectThreshold* value. The *RejectionResponse* parameter is required if you set the *SCLRejectEnabled* parameter to `$true`. The *RejectionResponse* parameter takes a string. Don't exceed 240 characters in the argument. When you pass an argument, you must enclose the *RejectionResponse* parameter in quotation marks (") if the phrase contains spaces, for example: `"Message rejected"`. The default setting is `Message rejected due to content restrictions`.
SCLDeleteEnabled	Optional	System.Boolean	The *SCLDeleteEnabled* parameter specifies whether all messages that exceed the value set in the *SCLDeleteThreshold* parameter are deleted.

			Valid input for the *SCLDeleteEnabled* parameter is `$true` or `$false`. The default setting is `$false`. When the *SCLDeleteEnabled* parameter is set to `$true`, all messages that exceed the value set in the *SCLDeleteThreshold* parameter are deleted.
SCLDeleteThreshold	Optional	System.Int32	The *SCLDeleteThreshold* parameter specifies an integer value from 1 through 9. This value represents the SCL rating that a particular message must exceed for the Content Filter agent to delete the message and not send an NDR. To enable this functionality, you must set the *SCLDeleteEnabled* parameter to `$true`. The default setting is 9.
SCLQuarantineEnabled	Optional	System.Boolean	The *SCLQuarantineEnabled* parameter specifies

			whether all messages that exceed the value set in the *SCLQuarantineThreshold* parameter are sent to the spam quarantine mailbox specified in the *QuarantineMailbox* parameter. Valid input for the *SCLQuarantineEnabled* parameter is `$true` or `$false`. The default setting is `$false`. When the *SCLQuarantineEnabled* parameter is set to `$true`, all messages that exceed the value set in the *SCLQuarantineThreshold* parameter are sent to the spam quarantine mailbox specified in the *QuarantineMailbox* parameter.
SCLQuarantineThreshold	Optional	System.Int32	The *SCLQuarantineThreshold* parameter specifies an integer value from 1 through 9. This value represents the SCL

			rating that a particular message must exceed for the Content Filter agent to quarantine the message. To enable quarantine functionality, you must set the *SCLQuarantineEnabled* parameter to `$true`, and provide a valid SMTP address in the *QuarantineMailbox* parameter. The default setting is 9.
SCLRejectEnabled	Optional	System.Boolean	The *SCLRejectEnabled* parameter specifies whether all messages that exceed the value set in the *SCLRejectThreshold* parameter are rejected, and an NDR is sent to the sender. Valid input for the *SCLRejectEnabled* parameter is `$true` or `$false`. The default setting is `$false`. When *SCLRejectEnabled* parameter is set to

			`$true`, all messages that exceed the value set in the *SCLRejectThreshold* parameter are rejected, and an NDR is sent to the sender.
SCLRejectThreshold	Optional	System.Int32	The *SCLRejectThreshold* parameter specifies an integer value from 1 through 9. This value represents the SCL rating that a particular message must exceed for the Content Filter agent to reject the message and send an NDR to the sender. To enable the delete functionality, you must set the *SCLDeleteEnabled* parameter to `$true`. Also, you can revise the default NDR message by editing the *RejectionResponse* parameter. The default setting is 9.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-ContentFilterPhrase

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Add-ContentFilterPhrase** cmdlet to define custom words for the Content Filter agent. A *custom word* is a word or phrase that the administrator sets for the Content Filter agent to evaluate the content of an e-mail message and apply appropriate filter processing.

```
Add-ContentFilterPhrase -Influence <GoodWord | BadWord> -Phrase <String>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds the phrase `Free credit report` to the Block phrase list. Any messages that contain this phrase will be marked as spam by the Content Filtering agent.

```
Add-ContentFilterPhrase -Phrase "Free credit report" -
Influence BadWord
```

Detailed Description

The **Add-ContentFilterPhrase** cmdlet adds phrases to the Allow or Block phrases list.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Influence	Required	Microsoft.Exchange.Data.Directory.SystemConfiguration.Influence	The *Influence* parameter specifies whether the phrase being added will cause the messages that contain the phrase to be allowed or blocked. Valid values are `GoodWord` and `BadWord`. A message that contains a custom word or phrase that has an *Influence* value of `GoodWord` is automatically assigned

			a spam confidence level (SCL) rating of `0` and therefore bypasses downstream spam processing. A message that contains a custom word or phrase that has an *Influence* value of `BadWord` is automatically assigned an SCL rating of `9` and therefore is treated as spam.
Phrase	Required	System.String	The *Phrase* parameter specifies a custom word or phrase for the Content Filter agent. When you pass an argument, you must enclose the *Phrase* parameter in quotation marks (") if the phrase contains spaces, for example: `"This is a bad phrase"`. Custom phrases must be less than 257 characters in length.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes

			would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ContentFilterPhrase

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ContentFilterPhrase** cmdlet to view one or all custom words that the Content Filter agent processes.

```
Get-ContentFilterPhrase [-Phrase <ContentFilterPhraseIdParameter>] <COMMON
PARAMETERS>

Get-ContentFilterPhrase [-Identity <ContentFilterPhraseIdParameter>]
<COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns all custom words stored on the computer where the command is being run.

```
Get-ContentFilterPhrase
```

EXAMPLE 2

This example returns a specific custom word specified by the *Phrase* parameter. In this example, the custom word is the phrase `Free credit report`.

```
Get-ContentFilterPhrase -Phrase "Free credit report"
```

EXAMPLE 3

This example returns all custom words and phrases that contain the words `free offer`.

```
Get-ContentFilterPhrase | Where {$_.Phrase -like '*free offer*'}
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An

			Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ContentFilterPhraseIdParameter	The *Identity* parameter specifies a custom word or phrase to display. You must enclose the value of the *Identity* parameter in quotation marks ("). **Note:** The *Identity* and *Phrase* parameters are interchangeable.
Phrase	Optional	Microsoft.Exchange.Configuration.Tasks.ContentFilterPhraseIdParameter	The *Phrase* parameter specifies a custom word or phrase to display. You must enclose the value of the *Phrase* parameter in quotation marks ("). **Note:** The *Phrase* and *Identity* parameters are interchangeable.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ContentFilterPhrase

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-ContentFilterPhrase** cmdlet to remove one or all custom words that the Content Filter agent processes.

```
Remove-ContentFilterPhrase [-Phrase <ContentFilterPhraseIdParameter>]
<COMMON PARAMETERS>

Remove-ContentFilterPhrase -Identity <ContentFilterPhraseIdParameter>
<COMMON PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the custom phrase Free credit report from the list of phrases that are filtered.

```
Remove-ContentFilterPhrase -Identity "Free credit report"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ContentFilterPhraseIdParameter	The *Identity* parameter specifies a custom word or phrase to remove. You must enclose the value of the *Identity* parameter in quotation marks ("). **Note:** For this command, the *Identity* parameter performs the same actions as the *Phrase* parameter. The *Identity* and *Phrase* parameters are interchangeable. You can't use the *Phrase* parameter if the *Identity* parameter is used.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Phrase	Optional	Microsoft.Exchange.Configuration.Tasks.ContentFilterPhraseIdParameter	The *Phrase* parameter specifies a custom word or phrase to remove. You must enclose the value of the *Phrase* parameter in quotation marks ("). **Note:** For this command, the *Phrase* parameter performs the same actions as the *Identity* parameter. The *Phrase* and *Identity* parameters are interchangeable. You can't use the *Identity* parameter if the *Phrase* parameter is used.
WhatIf	Optional	System.Management.A	The *WhatIf* switch

		utomation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IPAllowListConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-IPAllowListConfig** cmdlet to view the configuration information for the IP Allow list that's used by the Connection Filtering agent on Edge Transport servers.

```
Get-IPAllowListConfig [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the IP Allow list configuration on the local Edge Transport server.

```
Get-IPAllowListConfig | Format-List
```

Detailed Description

On Edge Transport servers, the Connection Filtering agent acts on the IP address of the incoming SMTP connection to determine what action, if any, to take on an incoming message.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory

			Services (AD LDS) to read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-IPAllowListConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Set-IPAllowListConfig** cmdlet to modify the IP Allow list configuration that's used by the Connection Filtering agent on Edge Transport servers.

```
Set-IPAllowListConfig [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Enabled <$true | $false>] [-ExternalMailEnabled <$true |
$false>] [-InternalMailEnabled <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures connection filtering to use the IP Allow list on messages that come from internal connections.

```
Set-IPAllowListConfig -InternalMailEnabled $true
```

Detailed Description

On Edge Transport servers, the Connection Filtering agent acts on the IP address of the incoming SMTP connection to determine what action, if any, to take on an incoming message.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge

			Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the IP Allow list is used for content filtering. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, the IP Allow list is used for content filtering.
ExternalMailEnabled	Optional	System.Boolean	The *ExternalMailEnabled* parameter specifies whether messages from connections outside of the Exchange organization are evaluated by the IP Allow list. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, messages from external

			connections are evaluated by the IP Allow list.
InternalMailEnabled	Optional	System.Boolean	The *InternalMailEnabled* parameter specifies whether messages from connections inside the Exchange organization are evaluated by the IP Allow list. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. By default, messages from internal connections are not evaluated by the IP Allow list. Authenticated partner messages aren't considered internal mail.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-IPAllowListEntry

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-12*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Add-IPAllowListEntry** cmdlet to add IP Allow list entries to the IP Allow list that's used by the Connection Filtering agent on Edge Transport servers.

```
Add-IPAllowListEntry -IPRange <IPRange> <COMMON PARAMETERS>

Add-IPAllowListEntry -IPAddress <IPAddress> <COMMON PARAMETERS>

COMMON PARAMETERS: [-Comment <String>] [-Confirm [<SwitchParameter>]] [-
ExpirationTime <DateTime>] [-Server <ServerIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds the IP address 192.168.0.100 to the list of allowed IP addresses.

```
Add-IPAllowListEntry -IPAddress 192.168.0.100
```

EXAMPLE 2

This example adds the IP address range 192.168.0.1/24 to the list of allowed IP addresses and configures the IP Allow list entry to expire at 23:59 on January 3, 2013.

```
Add-IPAllowListEntry -IPRange 192.168.0.1/24 -
ExpirationTime "1/3/2013 23:59"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
IPAddress	Required	System.Net.IPAddress	The *IPAddress* parameter specifies a single IP address to add to the IP Allow list, for example, `192.168.0.1`.
IPRange	Required	Microsoft.Exchange.Data.IPRange	The *IPRange* parameter specifies a range of IP addresses to add to the IP Allow list. You can use the following formats: • **CIDR IP** `192.168.0.1/24` • **IP address range** `192.168.0.1-192.168.0.254`.
Comment	Optional	System.String	The *Comment* parameter

			specifies an optional comment. If you specify a value that contains spaces, enclose the value in quotation marks ("), for example: "`This is an admin note`".
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ExpirationTime	Optional	System.DateTime	The *ExpirationTime* parameter specifies a day and time when the IP Allow list entry that you're creating will expire. If you specify a time only, and you don't specify a date, the current day is assumed. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer

			is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can't use this parameter to configure other Edge Transport servers remotely.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IPAllowListEntry

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-IPAllowListEntry** cmdlet to view the IP address entries in the IP Allow list that's used by the Connection Filtering agent on Edge Transport servers.

```
Get-IPAllowListEntry -IPAddress <IPAddress> <COMMON PARAMETERS>
```

```
Get-IPAllowListEntry [-Identity <IPListEntryIdentity>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-ResultSize <Unlimited>] [-Server <ServerIdParameter>]
```

Examples

EXAMPLE 1

This example returns all entries in the IP Allow list on the local Edge Transport server.

```
Get-IPAllowListEntry
```

EXAMPLE 2

This example returns an IP Allow list entry in which the specified IP address is included.

```
Get-IPAllowListEntry -IPAddress 192.168.0.1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
IPAddress	Required	System.Net.IPAddress	The *IPAddress* parameter specifies an IP address to view in the IP Allow list entry or entries. For example, if you have an IP Allow list entry that specifies a range of IP addresses from 192.168.0.1 through 192.168.0.20, enter any IP address in the IP Allow list

			IP address range to return the IP Allow list entry.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.IPListEntryIdentity	The *Identity* parameter specifies the identity integer value of the IP Allow list entry that you want to view. When you add an entry to the IP Allow list, the *Identity* value is automatically assigned.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN

			If you don't use the *Server* parameter, the command is run on the local server. You can't use this parameter to configure other Edge Transport servers remotely.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-IPAllowListEntry

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Remove-IPAllowListEntry** cmdlet to remove IP address entries from the IP Allow list that's used by the Connection Filtering agent on Edge Transport servers.

```
Remove-IPAllowListEntry -Identity <IPListEntryIdentity> [-Confirm
[<SwitchParameter>]] [-Server <ServerIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the IP address 192.168.0.100 from the IP Allow list.

```
Get-IPAllowListEntry | Where {$_.IPRange -eq
'192.168.0.100'} | Remove-IPAllowListEntry
```

EXAMPLE 2

This example removes the IP address range 192.168.0.0/24 from the IP Allow list.

```
Get-IPAllowListEntry | Where {$_.IPRange -eq
'192.168.0.0/24'} | Remove-IPAllowListEntry
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.IPListEntryIdentity	The *Identity* parameter specifies the integer value of the IP Allow list entry that you want to remove. When you add an entry to the IP Allow list, the *Identity* value is automatically assigned. To find the *Identity* value of an IP Allow list entry, use the **Get-IPAllowListEntry** cmdlet.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that

			appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can't use this parameter to configure other Edge Transport servers remotely.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-IPAllowListProvider

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Add-IPAllowListProvider** cmdlet to create IP Allow list providers that are used by the Connection Filtering agent on Edge Transport servers.

```
Add-IPAllowListProvider -LookupDomain <SmtpDomain> -Name <String> [-
AnyMatch <$true | $false>] [-BitmaskMatch <IPAddress>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-IPAddressesMatch <MultiValuedProperty>] [-Priority <Int32>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds a new IP Allow list provider and configures connection filtering to treat any IP address status code returned from the IP Allow list provider as a match and allow the connection. You get the value for the *LookupDomain* parameter from the allow list provider.

```
Add-IPAllowListProvider -Name "Contoso.com Allow List" -
LookupDomain allowlist.contoso.com -AnyMatch $true
```

EXAMPLE 2

This example adds an IP Allow list provider and configures a bitmask return value from the provider. You get the values for the *LookupDomain* and *BitmaskMatch* parameters from the allow list provider.

```
Add-IPAllowListProvider -Name "Fabrikam.com Allow List" -
LookupDomain allowlist.fabrikam.com -BitmaskMatch 127.1.0.1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
LookupDomain	Required	Microsoft.Exchange.Data.SmtpDomain	The *LookupDomain* parameter specifies the host name that's required to use the allow list provider. Connection filtering sends the IP address of the connecting SMTP server to the host name value that you specify. An example value is

			`allowlist.spamservice.com`. The actual value you need to use is provided by the allow list provider.
Name	Required	System.String	The *Name* parameter specifies a descriptive name for the IP Allow list provider.
AnyMatch	Optional	System.Boolean	The *AnyMatch* parameter specifies whether any response by the allow list provider is treated as a match. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When this parameter is set to `$true`, and connection filtering sends the IP address of the connecting SMTP server to the allow list provider, any response code returned by the allow list provider causes connection filtering to allow messages from that source.
BitmaskMatch	Optional	System.Net.IPAddress	The *BitmaskMatch*

			parameter specifies the bit mask status code that's returned by the allow list provider. Use this parameter if the allow list provider returns bitmask responses. Valid input for this parameter is a single IP address in the format `127.0.0.1`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't

			supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the connection filtering uses this IP Allow List provider. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, connection filtering uses new IP Allow List providers that you create.
IPAddressesMatch	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *IPAddressesMatch* parameter specifies the IP address status codes that are returned by the allow list provider. Use this parameter if the allow list provider returns IP address or A record responses. Valid input for this parameter one or more IP addresses in the

			format `127.0.0.1`. You can enter multiple IP addresses separated by commas.
Priority	Optional	System.Int32	The *Priority* parameter specifies the order that the Connection Filtering agent queries the IP Allow list providers that you have configured. A lower priority integer value indicates a higher priority. By default, every time that you add a new IP Allow list provider, the entry is assigned a priority of *N* +1, where *N* is the number of IP Allow list providers that you have configured. If you set the *Priority* parameter to a value that's the same as another IP Allow list provider, the priority of the IP Allow list provider that you add first is incremented by 1.
WhatIf	Optional	System.Management.A	The *WhatIf* switch

		utomation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IPAllowListProvider

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-IPAllowListProvider** cmdlet to view IP Allow list providers that are used by the Connection Filtering agent on Edge Transport servers.

```
Get-IPAllowListProvider [-Identity <IPAllowListProviderIdParameter>] [-
```

```
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns a summary list of all IP Allow list providers.

```
Get-IPAllowListProvider
```

EXAMPLE 2

This example returns detailed information for the IP Allow list provider named Contoso.com.

```
Get-IPAllowListProvider Contoso.com | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active

			Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.IPAllowListProviderIdParameter	The *Identity* parameter specifies the IP Allow list provider that you want to view. You can use any value that uniquely identifies the IP Allow list provider. For example: • Name • Distinguished name (DN) • GUID

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-IPAllowListProvider

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Remove-IPAllowListProvider** cmdlet to remove IP Allow list providers that are used by the Connection Filtering agent on Edge Transport servers.

```
Remove-IPAllowListProvider -Identity <IPAllowListProviderIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the IP Allow list provider named Contoso.com.

```
Remove-IPAllowListProvider Contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.IPAllowListProviderIdParameter	The *Identity* parameter specifies the IP Allow list provider that you want to remove. You can use any value that uniquely identifies the IP Allow list provider. For example: • Name • Distinguished name (DN) • GUID
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that

			appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-IPAllowListProvider

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Set-IPAllowListProvider** cmdlet to modify IP Allow list providers that are used by the Connection Filtering agent on Edge Transport servers.

```
Set-IPAllowListProvider -Identity <IPAllowListProviderIdParameter> [-
AnyMatch <$true | $false>] [-BitmaskMatch <IPAddress>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-IPAddressesMatch <MultiValuedProperty>] [-LookupDomain
<SmtpDomain>] [-Name <String>] [-Priority <Int32>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures the IP Allow list provider named Contoso.com to allow connections from

an IP address if any IP address status codes are returned.

```
Set-IPAllowListProvider Contoso.com -AnyMatch $true
```

EXAMPLE 2

This example sets the priority to 1 for the existing IP Allow list provider named Contoso.com.

```
Set-IPAllowListProvider Contoso.com -Priority 1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.IPAllowListProviderIdParameter	The *Identity* parameter specifies the IP Allow list provider that you want to modify. You can use any value that uniquely identifies the IP Allow list provider. For example: • Name • Distinguished name (DN) • GUID
AnyMatch	Optional	System.Boolean	The *AnyMatch* parameter specifies whether any response by the allow list provider is treated as a match. Valid input for this parameter is `$true` or

			`$false`. The default value is `$false`. When this parameter is set to `$true`, and connection filtering sends the IP address of the connecting SMTP server to the allow list provider, any response code returned by the allow list provider causes connection filtering to allow messages from that source.
BitmaskMatch	Optional	System.Net.IPAddress	The *BitmaskMatch* parameter specifies the bit mask status code that's returned by the allow list provider. Use this parameter if the allow list provider returns bitmask responses. Valid input for this parameter is a single IP address in the format `127.0.0.1`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a

			value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the Connection Filtering agent queries the IP Allow list provider according to the priority set for this IP Allow list provider configuration. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, the Connection

			Filtering agent queries the IP Allow list provider according to the priority set for this IP Allow list provider configuration.
IPAddressesMatch	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *IPAddressesMatch* parameter specifies the IP address status codes that are returned by the allow list provider. Use this parameter if the allow list provider returns IP address or A record responses. Valid input for this parameter one or more IP addresses in the format `127.0.0.1`. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing

			entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
LookupDomain	Optional	Microsoft.Exchange.Data.SmtpDomain	The *LookupDomain* parameter specifies the host name that's required to use the allow list provider. Connection filtering sends the IP address of the connecting SMTP server to the host name value that you specify. An example value is `allowlist.spamservice.com`. The actual value you need to use is provided by the allow list provider.
Name	Optional	System.String	The *Name* parameter specifies a descriptive name for the IP Allow list provider.
Priority	Optional	System.Int32	The *Priority* parameter specifies the order that the Connection Filtering agent queries the IP Allow list providers that you've configured. By default, every time that you add a new IP Allow list provider,

			the entry is assigned a priority of *N*+1, where *N* is the number of IP Allow list providers you've configured. If you set the *Priority* parameter to a value that's the same as another IP Allow list provider, the priority of the IP Allow list provider that you added first is incremented by 1.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-IPAllowListProvider

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Test-IPAllowListProvider** cmdlet to test IP Allow list providers on Edge Transport servers.

```
Test-IPAllowListProvider -Identity <IPAllowListProviderIdParameter> -
IPAddress <IPAddress> [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Server <ServerIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the existing IP Allow list provider named Contoso,com by sending a lookup request to that provider for the IP address 192.168.0.1.

```
Test-IPAllowListProvider Contoso.com -IPAddress 192.168.0.1
```

Detailed Description

On Edge Transport servers, the **Test-IPAllowListProvider** cmdlet checks connectivity to the specified allow list provider and then issues a lookup request to the allow list provider.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.IPAllowListProviderIdPara	The *Identity* parameter specifies the IP Allow list provider that you want to

		meter	test. You can use any value that uniquely identifies the IP Allow list provider. For example: • Name • Distinguished name (DN) • GUID
IPAddress	Required	System.Net.IPAddress	The *IPAddress* parameter specifies an IP address to be used in testing the IP Allow list provider. You need to use a known allowed IP address.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

			The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can't use this parameter to configure other Edge Transport servers remotely.
WhatIf	Optional	System.Management.Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IPAllowListProvidersConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-IPAllowListProvidersConfig** cmdlet to view the settings that that affect all IP Allow list providers that are configured on an Edge Transport server.

```
Get-IPAllowListProvidersConfig [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the IP Allow list providers configuration on the local Edge Transport server.

```
Get-IPAllowListProvidersConfig | Format-List
```

Detailed Description

On Edge Transport servers, IP Allow list providers are used by the Connection Filtering agent. The Connection Filtering agent acts on the IP address of the incoming SMTP connection to determine what action, if any, to take on an incoming message.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to

			read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-IPAllowListProvidersConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Set-IPAllowListProvidersConfig** cmdlet to modify the settings that that affect all IP Allow list providers that are configured on an Edge Transport server.

```
Set-IPAllowListProvidersConfig [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Enabled <$true | $false>] [-ExternalMailEnabled
<$true | $false>] [-InternalMailEnabled <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures connection filtering to use IP Allow list providers on messages that come from internal connections.

```
Set-IPAllowListProvidersConfig -InternalMailEnabled $true
```

Detailed Description

On Edge Transport servers, IP Allow list providers are used by the Connection Filtering agent. The Connection Filtering agent acts on the IP address of the incoming SMTP connection to determine what action, if any, to take on an incoming message.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An

			Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether IP Allow list providers are used for content filtering. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. The default value is `$true`. By default, IP Allow list providers are used for content filtering.
ExternalMailEnabled	Optional	System.Boolean	The *ExternalMailEnabled* parameter specifies whether messages from connections outside of the Exchange organization are evaluated by IP Allow list providers. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, messages from

			external connections are evaluated by IP Allow list providers.
InternalMailEnabled	Optional	System.Boolean	The *InternalMailEnabled* parameter specifies whether messages from connections inside the Exchange organization are evaluated by IP Allow list providers. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. By default, messages from internal connections are not evaluated by IP Allow list providers. Authenticated partner messages aren't considered internal mail.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IPBlockListConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-IPBlockListConfig** cmdlet to view the IP Block list configuration information that's used by the Connection Filtering agent on Edge Transport servers.

```
Get-IPBlockListConfig [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the IP Block list configuration on the local Edge Transport server.

```
Get-IPBlockListConfig | Format-List
```

Detailed Description

On Edge Transport servers, the Connection Filtering agent acts on the incoming SMTP connection to determine what action, if any, to take on an incoming message.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-IPBlockListConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Set-IPBlockListConfig** cmdlet to modify the IP Block list configuration that's used by the Connection Filtering agent on Edge Transport servers.

```
Set-IPBlockListConfig [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Enabled <$true | $false>] [-ExternalMailEnabled <$true |
$false>] [-InternalMailEnabled <$true | $false>] [-
MachineEntryRejectionResponse <AsciiString>] [-
StaticEntryRejectionResponse <AsciiString>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures connection filtering to use the IP Block list on messages that come from internal connections.

```
Set-IPBlockListConfig -InternalMailEnabled $true
```

Detailed Description

On Edge Transport servers, the Connection Filtering agent acts on the IP address of the incoming SMTP connection to determine what action, if any, to take on an incoming message.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the IP Block list is used for content filtering. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, the IP Block list is used for content filtering.
ExternalMailEnabled	Optional	System.Boolean	The *ExternalMailEnabled* parameter specifies whether messages from connections outside of the Exchange organization are evaluated by the IP Block list. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, messages from external connections are evaluated by the IP Block list.
InternalMailEnabled	Optional	System.Boolean	The *InternalMailEnabled* parameter specifies whether messages

			from connections inside the Exchange organization are evaluated by the IP Block list. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. By default, messages from internal connections are not evaluated by the IP Block list. Authenticated partner messages aren't considered internal mail.
MachineEntryRejectionResponse	Optional	Microsoft.Exchange.Data.AsciiString	The *MachineEntryRejectionResponse* parameter specifies customized text in the non-delivery report (NDR) for messages that are blocked by connection filtering due to IP addresses in the IP Block list that were added by sender reputation. The value can't exceed 240 characters. If the value contains spaces,

			enclose the value in double quotation marks (").
StaticEntryRejectionResponse	Optional	Microsoft.Exchange.Data.AsciiString	The *StaticEntryRejectionResponse* parameter specifies a customized text in the NDR for messages that are blocked by connection filtering due to IP addresses in the IP Block list. The value can't exceed 240 characters. If the value contains spaces, enclose the value in double quotation marks (").
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-IPBlockListEntry

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-12*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Add-IPBlockListEntry** cmdlet to add IP Block list entries to the IP Block list that's used by the Connection Filtering agent on Edge Transport servers.

```
Add-IPBlockListEntry -IPRange <IPRange> <COMMON PARAMETERS>

Add-IPBlockListEntry -IPAddress <IPAddress> <COMMON PARAMETERS>

COMMON PARAMETERS: [-Comment <String>] [-Confirm [<SwitchParameter>]] [-
ExpirationTime <DateTime>] [-Server <ServerIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds the IP address 192.168.0.100 to the list of blocked IP addresses.

```
Add-IPBlockListEntry -IPAddress 192.168.0.100
```

EXAMPLE 2

This example adds the IP address range 192.168.0.1/24 to the list of blocked IP addresses and

configures the IP Block list entry to expire at 23:59 on January 3, 2013.

```
Add-IPBlockListEntry -IPRange 192.168.0.1/24 -
ExpirationTime "1/3/2013 23:59"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
IPAddress	Required	System.Net.IPAddress	The *IPAddress* parameter specifies a single IP address to add to the IP Block list, for example, `192.168.0.1`.
IPRange	Required	Microsoft.Exchange.Data.IPRange	The *IPRange* parameter specifies a range of IP addresses to add to the IP Block list. You can use the following formats: • **CIDR IP** `192.168.0.1/24` • **IP address range** `192.168.0.1-192.168.0.254`.
Comment	Optional	System.String	The *Comment* parameter specifies an optional comment. If you specify a value that contains spaces, enclose the value in quotation marks ("), for

			example: "`This is an admin note`".
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ExpirationTime	Optional	System.DateTime	The *ExpirationTime* parameter specifies a day and time when the IP Block list entry that you're creating will expire. If you specify a time only, and you don't specify a date, the current day is assumed. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date

			only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can't use this parameter to configure other Edge Transport servers remotely.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IPBlockListEntry

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-12*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-IPBlockListEntry** cmdlet to view the IP Block list entries that are used by the Connection Filtering agent on Edge Transport servers.

```
Get-IPBlockListEntry -IPAddress <IPAddress> <COMMON PARAMETERS>

Get-IPBlockListEntry [-Identity <IPListEntryIdentity>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-ResultSize <Unlimited>] [-Server <ServerIdParameter>]
```

Examples

EXAMPLE 1

This example returns a list of all IP address entries in the IP Block list.

```
Get-IPBlockListEntry
```

EXAMPLE 2

This example returns machine-generated entries in the IP Block list that are inserted by sender reputation.

```
Get-IPBlockListEntry | where {$_.IsMachineGenerated}
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
IPAddress	Required	System.Net.IPAddress	The *IPAddress* parameter specifies an IP address to view in the IP Block list entry or entries. For example, if you have an IP Block list entry that specifies a range of IP addresses from 192.168.0.1 through 192.168.0.20, enter any IP address in the IP Block list IP address range to return the IP Block list entry.

Identity	Optional	Microsoft.Exchange.Configuration.Tasks.IPListEntryIdentity	The *Identity* parameter specifies the identity integer value of the IP Block list entry that you want to view. When you add an entry to the IP Block list, the *Identity* value is automatically assigned.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command

			is run on the local server. You can't use this parameter to configure other Edge Transport servers remotely.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-IPBlockListEntry

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-01-12*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Remove-IPBlockListEntry** cmdlet to remove IP block list entries that are used by the Connection Filtering agent on Edge Transport servers.

```
Remove-IPBlocklistEntry -Identity <IPListEntryIdentity> [-Confirm
[<SwitchParameter>]] [-Server <ServerIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the IP address 192.168.0.100 from the IP Block list.

```
Get-IPBlockListEntry | Where {$_.IPRange -eq
'192.168.0.100'} | Remove-IPBlockListEntry
```

EXAMPLE 2

This example removes the IP address range 192.168.0.0/24 from the IP Block list.

```
Get-IPBlockListEntry | Where {$_.IPRange -eq
'192.168.0.0/24'} | Remove-IPBlockListEntry
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.IPListEntryIdentity	The *Identity* parameter specifies the integer value of the IP Block list entry that you want to remove. When you add an entry to the IP Block list, the *Identity* value is automatically assigned. To find the *Identity* value of an IP Block list entry, use the **Get-IPBlockListEntry** cmdlet.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To

			suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can't use this command to configure other Edge Transport servers remotely.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-IPBlockListProvider

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Add-IPBlockListProvider** cmdlet to create IP Block list providers that are used by the Connection Filtering agent on Edge Transport servers.

```
Add-IPBlockListProvider -LookupDomain <SmtpDomain> -Name <String> [-
AnyMatch <$true | $false>] [-BitmaskMatch <IPAddress>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-IPAddressesMatch <MultiValuedProperty>] [-Priority <Int32>] [-
RejectionResponse <AsciiString>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds an IP Block list provider and sets a rejection response. You get the value for the *LookupDomain* parameter from the block list provider.

```
Add-IPBlockListProvider -Name "Contoso.com Block List" -
LookupDomain blocklist.contoso.com -RejectionResponse
"Source IP address is listed at the Contoso.com block list
provider"
```

EXAMPLE 2

This example adds an IP Block list provider and configures a bitmask return value from the provider. You get the values for the *LookupDomain* and *BitmaskMatch* parameters from the block list provider.

```
Add-IPBlockListProvider -Name "Fabrikam.com Block List" -
LookupDomain blocklist.fabrikam.com -BitmaskMatch 127.1.0.1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
LookupDomain	Required	Microsoft.Exchange.Data.SmtpDomain	The *LookupDomain* parameter specifies the host name that's required to use the block list provider. Connection filtering sends the IP address of the connecting SMTP server to the host name value that you specify. An example value is `blocklist.spamservice.com`. The actual value you need to use is

			provided by the block list provider.
Name	Required	System.String	The *Name* parameter specifies a descriptive name for the IP Block list provider.
AnyMatch	Optional	System.Boolean	The *AnyMatch* parameter specifies whether any response by the block list provider is treated as a match. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When this parameter is set to `$true`, and connection filtering sends the IP address of the connecting SMTP server to the block list provider, any response code returned by the block list provider causes connection filtering to block messages from that source.
BitmaskMatch	Optional	System.Net.IPAddress	The *BitmaskMatch* parameter specifies the bit mask status code

			that's returned by the block list provider. Use this parameter if the block list provider returns bitmask responses. Valid input for this parameter is a single IP address in the format `127.0.0.1`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An

			Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the connection filtering uses this IP Block list provider. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, connection filtering uses new IP Block List providers that you create.
IPAddressesMatch	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *IPAddressesMatch* parameter specifies the IP address status codes that are returned by the block list provider. Use this parameter if the block list provider returns IP address or A record responses. Valid input for this parameter one or more IP addresses in the format `127.0.0.1`. You can enter multiple IP

			addresses separated by commas.
Priority	Optional	System.Int32	The *Priority* parameter specifies the order that the Connection Filtering agent queries the IP Block list providers. A lower priority integer value indicates a higher priority. By default, every time that you add a new IP Block list provider, the entry is assigned a priority of *N* +1, where *N* is the number of IP Block list provider services that you have configured. If you set the *Priority* parameter to a value that's the same as another IP Block list provider service, the priority of the IP Block list provider that you add first is incremented by 1.
RejectionResponse	Optional	Microsoft.Exchange.Data.AsciiString	The *RejectionResponse* parameter specifies the text that you want to include in the SMTP

			rejection response when messages are blocked by connection filtering. The argument can't exceed 240 characters. If the value contains spaces, enclose the value in quotation marks ("). You should always specify the block list provider in the response so that legitimate senders can contact the block list provider for removal instructions. For example, `"Source IP address is listed at the Contoso.com block list provider"`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IPBlockListProvider

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-IPBlockListProvider** cmdlet to view IP Block list providers that are used by the Connection Filtering agent on Edge Transport servers.

```
Get-IPBlockListProvider [-Identity <IPBlockListProviderIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns a summary list of all IP Block list providers configured on the local Edge Transport server.

```
Get-IPBlockListProvider
```

EXAMPLE 2

This example returns detailed information for the existing IP Block list provider named Contoso.com.

```
Get-IPBlockListProvider -Identity Contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.IPBlockListProviderIdParameter	The *Identity* parameter specifies the IP Block list provider that you want to view. You can use any value that uniquely

			identifies the IP Block list provider. For example: • Name • Distinguished name (DN) • GUID

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-IPBlockListProvider

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Remove-IPBlockListProvider** cmdlet to remove IP Block list providers that are used by the Connection Filtering agent on Edge Transport server.

```
Remove-IPBlockListProvider -Identity <IPBlockListProviderIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the IP Block list provider named Contoso.com.

```
Remove-IPBlockListProvider Contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.IPBlockListProviderIdParameter	The *Identity* parameter specifies the IP Block list provider that you want to remove. You can use any value that uniquely identifies the IP Block list provider. For example: • Name • Distinguished name (DN) • GUID
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Da	The *DomainController* parameter specifies the

		ta.Fqdn	fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-IPBlockListProvider

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Set-IPBlockListProvider** cmdlet to modify IP Block list providers that are used by the Connection Filtering agent on Edge Transport servers.

```
Set-IPBlockListProvider -Identity <IPBlockListProviderIdParameter> [-
AnyMatch <$true | $false>] [-BitmaskMatch <IPAddress>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-IPAddressesMatch <MultiValuedProperty>] [-LookupDomain
<SmtpDomain>] [-Name <String>] [-Priority <Int32>] [-RejectionResponse
<AsciiString>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures connection filtering to block an IP address if any IP address status codes are returned by the IP Block list provider named Contoso.com.

```
Set-IPBlockListProvider Contoso.com -AnyMatch $true
```

EXAMPLE 2

This example sets the priority value to 1 for the IP Block list provider named Contoso.com.

```
Set-IPBlockListProvider Contoso.com -Priority 1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.IPBlockListProviderIdParameter	The *Identity* parameter specifies the IP Block list provider that you want to modify. You can use any value that uniquely identifies the IP Block list provider. For example: • Name • Distinguished name (DN) • GUID
AnyMatch	Optional	System.Boolean	The *AnyMatch* parameter specifies whether any response by the block list provider is treated as a match. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When this parameter is set to `$true`, and connection filtering sends the IP address of the connecting SMTP server to the block list provider, any response code returned by the block list provider causes

			connection filtering to block messages from that source.
BitmaskMatch	Optional	System.Net.IPAddress	The *BitmaskMatch* parameter specifies the bit mask status code that's returned by the block list provider. Use this parameter if the block list provider returns bitmask responses. Valid input for this parameter is a single IP address in the format `127.0.0.1`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

			The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the connection filtering uses this IP Block list provider. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, connection filtering uses new IP Block list providers that you create.
IPAddressesMatch	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *IPAddressesMatch* parameter specifies the IP address status codes that are returned by the block list provider. Use this parameter if the block list provider returns IP address or A record responses. Valid input for this parameter one or more IP addresses in the

			format `127.0.0.1`. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
LookupDomain	Optional	Microsoft.Exchange.Data.SmtpDomain	The *LookupDomain* parameter specifies the host name that's required to use the block list provider. Connection filtering sends the IP address of the connecting SMTP server to the host name value that you specify. An example value is `blocklist.spamservice.`

			com. The actual value you need to use is provided by the block list provider.
Name	Optional	System.String	The *Name* parameter specifies a descriptive name for the IP Block list provider.
Priority	Optional	System.Int32	The *Priority* parameter specifies the order that the Connection Filtering agent queries the IP Block list providers. A lower priority integer value indicates a higher priority. By default, every time that you add a new IP Block list provider, the entry is assigned a priority of *N* +1, where *N* is the number of IP Block list provider services that you have configured. If you set the *Priority* parameter to a value that's the same as another IP Block list provider service, the priority of the IP Block list provider that you add first is incremented by 1.
RejectionResponse	Optional	Microsoft.Exchange.Da	The *RejectionResponse*

		ta.AsciiString	parameter specifies the text that you want to include in the SMTP rejection response when messages are blocked by connection filtering. The argument can't exceed 240 characters. If the value contains spaces, enclose the value in quotation marks ("). You should always specify the block list provider in the response so that legitimate senders can contact the block list provider for removal instructions. For example, `"Source IP address is listed at the Contoso.com block list provider"`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-IPBlockListProvider

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Test-IPBlockListProvider** cmdlet to test IP Block list providers on Edge Transport servers.

```
Test-IPBlockListProvider -Identity <IPBlockListProviderIdParameter> -
IPAddress <IPAddress> [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Server <ServerIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the existing IP Block list provider named Contoso,com by sending a lookup request to that provider for the IP address 192.168.0.1.

```
Test-IPBlockListProvider Contoso.com -IPAddress 192.168.0.1
```

Detailed Description

On Edge Transport servers, the **Test-IPBlockListProvider** cmdlet checks connectivity to the specified block list provider and then issues a lookup request to the block list provider.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.IPBlockListProviderIdParameter	The *Identity* parameter specifies the IP Block list provider that you want to test. You can use any value that uniquely identifies the IP Block list provider. For example: • Name • Distinguished name (DN) • GUID
IPAddress	Required	System.Net.IPAddress	The *IPAddress* parameter specifies an IP address to be used in testing the IP Block list provider. You need to use a known blocked IP address.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command

			is run on the local server. You can't use this parameter to configure other Edge Transport servers remotely.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IPBlockListProvidersConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-IPBlockListProvidersConfig** cmdlet to view the settings that that affect all IP Block list providers that are configured on an Edge Transport server.

```
Get-IPBlockListProvidersConfig [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the IP Block list providers on the local Edge Transport server.

```
Get-IPBlockListProvidersConfig | Format-List
```

Detailed Description

On Edge Transport servers, IP Block list providers are used by the Connection Filtering agent. The Connection Filtering agent acts on the IP address of the incoming SMTP connection to determine what action, if any, to take on an incoming message.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController*

			parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-IPBlockListProvidersConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Set-IPBlockListProvidersConfig** cmdlet to modify the settings that affect all IP Block list providers that are configured on an Edge Transport server.

```
Set-IPBlockListProvidersConfig [-BypassedRecipients <MultiValuedProperty>]
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true
| $false>] [-ExternalMailEnabled <$true | $false>] [-InternalMailEnabled
<$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures connection filtering to use IP Block list providers on messages that come from internal connections, but bypasses filtering for email messages sent to kweku@contoso.com.

```
Set-IPBlockListProvidersConfig -InternalMailEnabled $true -
BypassedRecipients kweku@contoso.com
```

EXAMPLE 2

This example makes the following changes to the list of bypassed recipients:

- Adds the values chris@contoso.com and michelle@contoso.com
- Removes the values laura@contoso.com and julia@contoso.com

```
Set-IPBlockListProvidersConfig -BypassedRecipients
@{Add="chris@contoso.com","michelle@contoso.com";
Remove="laura@contoso.com","julia@contoso.com"}
```

Detailed Description

On Edge Transport servers, IP Block list providers are used by the Connection Filtering agent. The Connection Filtering agent acts on the IP address of the incoming SMTP connection to determine what action, if any, to take on an incoming message.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features - Edge Transport" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
BypassedRecipients	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassedRecipients* parameter specifies the email addresses of internal recipients that are exempted from filtering by IP Block list

			providers. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...` If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...`. To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether IP Block list providers are used for content filtering. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. The default value is `$true`. By default, IP Block list providers are used for content filtering.
ExternalMailEnabled	Optional	System.Boolean	The *ExternalMailEnabled*

			parameter specifies whether messages from connections outside of the Exchange organization are evaluated by IP Block list providers. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, messages from external connections are evaluated by IP Bock list providers.
InternalMailEnabled	Optional	System.Boolean	The *InternalMailEnabled* parameter specifies whether messages from connections inside the Exchange organization are evaluated by IP Block list providers. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. By default, messages from internal connections are not evaluated by IP Block list providers.

			Authenticated partner messages aren't considered internal mail.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxJunkEmailConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MaillboxJunkEmailConfiguration** cmdlet to view the junk email rule configuration for specific mailboxes.

```
Get-MailboxJunkEmailConfiguration -Identity <MailboxIdParameter> [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-
ReadFromDomainController <SwitchParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example displays the junk email rule configuration for the user named David Pelton.

```
Get-MailboxJunkEmailConfiguration "David Pelton"
```

EXAMPLE 2

This example displays the junk email rule configuration for all mailboxes in your organization that have the junk email rule enabled.

```
Get-MailboxJunkEmailConfiguration -Identity * | Where
{$_.Enabled -eq $true}
```

Detailed Description

The junk email rule helps Microsoft Outlook and Outlook Web App users to automatically remove any spam that gets past anti-spam filters and reaches their mailboxes.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox junk email configuration" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox. This parameter accepts

			the following values: • Alias Example: JPhillips • Canonical DN Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • Display Name Example: Jeff Phillips • Distinguished Name (DN) Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • Domain\Account Example: Atlanta\JPhillips • GUID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • Immutable ID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • Legacy Exchange DN Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • SMTP Address Example: Jeff.Phillips@contoso.com • User Principal Name Example: JPhillips@contoso.com You can use the wildcard character (*) to view the junk email rule

			configuration for multiple mailboxes.
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	The *ReadFromDomainController* switch specifies that information should be read from a domain controller in the user's domain. If you run the

			command `Set-AdServerSettings -ViewEntireForest $true` to include all objects in the forest and you don't use the *ReadFromDomainController* switch, it's possible that information will be read from a global catalog that has outdated information. When you use the *ReadFromDomainController* switch, multiple reads might be necessary to get the information. You don't have to specify a value with this switch. **Note:** By default, the recipient scope is set to the domain that hosts your Exchange servers.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxJunkEmailConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailboxJunkEmailConfiguration** cmdlet to configure the junk email rule for specific mailboxes.

```
Set-MailboxJunkEmailConfiguration -Identity <MailboxIdParameter> [-
BlockedSendersAndDomains <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-ContactsTrusted <$true | $false>] [-
DomainController <Fqdn>] [-Enabled <$true | $false>] [-IgnoreDefaultScope
<SwitchParameter>] [-TrustedListsOnly <$true | $false>] [-
TrustedSendersAndDomains <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the junk email rule configuration for the user named David Pelton.

```
Set-MailboxJunkEmailConfiguration "David Pelton" -Enabled
$false
```

EXAMPLE 2

This example makes the following configuration changes to the junk email rule for the user named

Michele Martin:

- Adds `contoso.com` and `fabrikam.com` as trusted domains without affecting any existing trusted domain entries.
- Adds `jane@fourthcoffee.com` as a blocked sender without affecting any existing blocked sender entries.

```
Set-MailboxJunkEmailConfiguration "Michele Martin" -
TrustedSendersAndDomains
@{Add="contoso.com","fabrikam.com"} -
BlockedSendersAndDomains @{Add="jane@fourthcoffee.com"}
```

EXAMPLE 3

This example identifies any mailboxes for which the junk email rule is configured to treat contacts as trusted senders and then changes the junk email configuration to not treat contacts as trusted senders.

```
Get-MailboxJunkEmailConfiguration * | Where
{$_.ContactsTrusted -eq $true} | Set-
MailboxJunkEmailConfiguration -ContactsTrusted $false
```

Detailed Description

The junk email rule helps Microsoft Outlook and Outlook Web App users to automatically remove any spam that gets past anti-spam filters and reaches the users' mailboxes. With this cmdlet, users and administrators can make changes to the junk email rule that's configured for a specific mailbox.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox junk email configuration" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter identifies the mailbox. This parameter accepts the following values: • `Alias` Example: JPhillips

			• Canonical DN Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • Display Name Example: Jeff Phillips • Distinguished Name (DN) Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • Domain\Account Example: Atlanta\JPhillips • GUID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • Immutable ID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • Legacy Exchange DN Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • SMTP Address Example: Jeff.Phillips@contoso.com • User Principal Name Example: JPhillips@contoso.com
BlockedSendersAndDomains	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BlockedSendersAndDomains* parameter specifies a list of the individual senders and domains that are blocked by the junk

			email rule. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ContactsTrusted	Optional	System.Boolean	The *ContactsTrusted*

			parameter specifies whether the contacts in the Contacts folder are treated as trusted senders. Valid input for this parameter is `$true` or `$false`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter enables or disables the junk email rule on this mailbox. Valid input for this parameter is `$true` or `$false`.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire

			forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
TrustedListsOnly	Optional	System.Boolean	The *TrustedListsOnly* parameter specifies that only messages from trusted senders and domains are allowed to be sent to the mailbox. All other messages are considered spam by the junk email rule. Valid input for this parameter is `$true` or `$false`.

TrustedSendersAndDomains	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *TrustedSendersAndDomains* parameter specifies a list of the individual senders and domains that are considered trusted senders. Messages from these senders and domains aren't processed by the junk email rule. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it

			would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MalwareFilteringServer

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MalwareFilteringServer** cmdlet to view the Malware agent settings in the Transport service on a Mailbox server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MalwareFilteringServer [-Identity <MalwareFilteringServerIdParameter>]
[-DomainController <Fqdn>]
```

Examples

Example 1

This example displays a summary of the Exchange Malware agent settings on all Mailbox servers in your organization.

```
Get-MalwareFilteringServer
```

Example 2

This example returns the detailed Exchange Malware agent settings on a Mailbox server named Mailbox01.

```
Get-MalwareFilteringServer Mailbox01 | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight

			Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MalwareFilteringServerIdParameter	The *Identity* parameter specifies the server where you want to view the anti-malware settings. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MalwareFilteringServer

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-MalwareFilteringServer** cmdlet to configure the Malware agent settings in the Transport service on a Mailbox server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MalwareFilteringServer -Identity <MalwareFilteringServerIdParameter>
[-BypassFiltering <$true | $false>] [-Confirm [<SwitchParameter>]] [-
DeferAttempts <Int32>] [-DeferWaitTime <Int32>] [-DomainController <Fqdn>]
[-ForceRescan <$true | $false>] [-MinimumSuccessfulEngineScans <Int32>] [-
PrimaryUpdatePath <String>] [-ScanErrorAction <Block | Allow>] [-
ScanTimeout <Int32>] [-SecondaryUpdatePath <String>] [-UpdateFrequency
<Int32>] [-UpdateTimeout <Int32>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example sets the following Malware agent settings on the Mailbox server named Mailbox01:

- Sets the update frequency interval to 2 hours
- Sets the time to wait between resubmit attempts to 10 minutes

```
Set-MalwareFilteringServer Mailbox01 -UpdateFrequency 120 -
DeferWaitTime 10
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MalwareFilteringServerIdParameter	The *Identity* parameter specifies the server where you want to configure the anti-malware settings. You can use any value that uniquely identifies the server. For example: • Name • FQDN

			• Distinguished name (DN) • Exchange Legacy DN
BypassFiltering	Optional	System.Boolean	The *BypassFiltering* parameter temporarily bypasses malware filtering without disabling the Malware agent on the server. The Malware agent is still active, and the agent is still called for every message, but no malware filtering is actually performed. This allows you to temporarily disable and then enable malware filtering on the server without disrupting mail flow by restarting the Microsoft Exchange Transport service. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a

			value with the *Confirm* switch.
DeferAttempts	Optional	System.Int32	The *DeferAttempts* parameter specifies the maximum number of times to defer a message that can't be scanned by the Malware agent. Valid input for this parameter is an integer between 1 and 5. The default value is 3. After the maximum number of deferrals is reached, the action taken by the Malware agent depends on the error. For scan timeouts and engine errors, the action is to fail the message and return a non-delivery report (NDR) to the sender immediately after the last defer attempt. For all other errors, the message is retried for up to 48 hours, with each retry attempt taking place one hour longer than the last one. For example, starting after the last defer attempt, the first retry attempt will occur in 1 hour, the next

			retry attempt will occur 2 hours after that, the next retry attempt will occur 3 hours after the second retry attempt, and so on for up to 48 hours. After 48 hours have elapsed, the action is to fail the message and return a non-delivery report (NDR) to the sender.
DeferWaitTime	Optional	System.Int32	The *DeferWaitTime* parameter specifies the time period in minutes to increase the interval to resubmit messages for malware filtering in an effort to reduce the workload on the server. For example, the first retry after the original failed scan occurs after the interval specified by the *DeferWaitTime* parameter. The second retry after the first retry occurs after two times the value of the *DeferWaitTime* parameter. The third retry after the second retry occurs after three times

			the value of the *DeferWaitTime* parameter, and so on. The maximum number of retries is controlled by the *DeferAttempts* parameter. Valid input for this parameter is an integer between 0 and 15. The default value is 5. This means the first resubmit occurs 5 minutes after the original failed scan, the second retry occurs 10 minutes after the first retry, the third retry occurs 15 minutes after the second retry, and so on. The value 0 means messages are resubmitted for malware filtering after any failed scanning attempts without any delay.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
ForceRescan	Optional	System.Boolean	The *ForceRescan* parameter specifies that messages should be scanned by the malware agent, even if the message was already scanned by Exchange Online Protection. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
MinimumSuccessfulEngineScans	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
PrimaryUpdatePath	Optional	System.String	The *PrimaryUpdatePath* parameter specifies where to download malware scanning engine updates. The default value is http://forefrontdl.microsoft.com/server/scanengineupdate.

			The location specified by the *PrimaryUpdatePath* parameter is always tried first.
ScanErrorAction	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MalwareScanErrorAction	The *ScanErrorAction* parameter specifies the action to take when a message can't be scanned by the malware filter. Valid values for this parameter are `Block` or `Allow`. The default value is `Block`.
ScanTimeout	Optional	System.Int32	The *ScanTimeout* parameter specifies the timeout interval in seconds for messages that can't be scanned by the malware filter. Valid input for this parameter is an integer between 10 and 900. The default value is 300 (5 minutes).
SecondaryUpdatePath	Optional	System.String	The *SecondaryUpdatePath* parameter specifies an alternate download location for malware scanning engine updates. The default values is blank ($null). This means no alternate download

			location is specified. The alternate download location is used when the location specified by the *PrimaryUpdatePath* parameter is unavailable for the time period specified by the *UpdateTimeout* parameter. On the next malware scanning engine update, the location specified by the *PrimaryUpdate* path parameter is tried first.
UpdateFrequency	Optional	System.Int32	The *UpdateFrequency* parameter specifies the frequency interval in minutes to check for malware scanning engine updates. Valid input for this parameter is an integer between 1 and 38880 (27 days). The default value is 60 (one hour). The locations to check for updates are specified by the *PrimaryUpdatePath* and *SecondaryUpdatePath* parameters.
UpdateTimeout	Optional	System.Int32	The *UpdateTimeout*

			parameter specifies the timeout interval in seconds to use when checking for malware scanning engine updates. Valid input for this parameter is an integer between 60 and 300. The default value is 150 seconds (2.5 minutes). If the location specified by the *PrimaryUpdatePath* parameter is unavailable for the time period specified by the *UpdateTimeout* parameter value, the location specified by the *SecondaryUpdatePath* parameter is used.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MalwareFilterPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the *Get-MalwareFilterPolicy* cmdlet to view the malware filter policies in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MalwareFilterPolicy [-Identity <MalwareFilterPolicyIdParameter>] [-
DomainController <Fqdn>] [-IgnoreDehydratedFlag <SwitchParameter>] [-
IncludeInvalidPolicies <SwitchParameter>] [-Organization
<OrganizationIdParameter>]
```

Examples

Example 1

This example retrieves a summary list of all malware filter policies in your organization.

```
Get-MalwareFilterPolicy
```

Example 2

This example retrieves detailed configuration information for the malware filter policy named Default.

```
Get-MalwareFilterPolicy Default | Format-List
```

Detailed Description

Malware filter policies contain the malware settings and a list of domains to which those settings apply. A domain can't belong to more than one malware filter policy.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MalwareFilterPolicyIdParameter	The *Identity* parameter specifies the malware filter policy that you want to view. You can use any value that uniquely identifies the policy. For example, you can use the name, GUID or distinguished name (DN) of the malware filter policy.

IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IncludeInvalidPolicies	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MalwareFilterPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the *New-MalwareFilterPolicy* cmdlet to create malware filter policies in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MalwareFilterPolicy -Name <String> [-Action <DeleteMessage |
DeleteAttachmentAndUseDefaultAlertText |
DeleteAttachmentAndUseCustomAlertText>] [-AdminDisplayName <String>] [-
```

```
BypassInboundMessages <$true | $false>] [-BypassOutboundMessages <$true |
$false>] [-Confirm [<SwitchParameter>]] [-CustomAlertText <String>] [-
CustomExternalBody <String>] [-CustomExternalSubject <String>] [-
CustomFromAddress <SmtpAddress>] [-CustomFromName <String>] [-
CustomInternalBody <String>] [-CustomInternalSubject <String>] [-
CustomNotifications <$true | $false>] [-DomainController <Fqdn>] [-
EnableExternalSenderAdminNotifications <$true | $false>] [-
EnableExternalSenderNotifications <$true | $false>] [-
EnableInternalSenderAdminNotifications <$true | $false>] [-
EnableInternalSenderNotifications <$true | $false>] [-
ExternalSenderAdminAddress <SmtpAddress>] [-IgnoreDehydratedFlag
<SwitchParameter>] [-InternalSenderAdminAddress <SmtpAddress>] [-
Organization <OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a new malware filter policy named Contoso Malware Filter Policy with the following settings:

- Block messages that contain malware.
- Don't notify the message sender when malware is detected in the message.
- Notify the administrator admin@contoso.com when malware is detected in a message from an internal sender.

```
New-MalwareFilterPolicy -Name "Contoso Malware Filter
Policy" -EnableInternalSenderAdminNotifications $true -
InternalSenderAdminAddress admin@contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies a name for the malware filter policy. If the value contains spaces, enclose the value in quotation marks (").

Action	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MalwareFilteringAction	The *Action* parameter specifies the action to take when malware is detected in a message. Valid values for this parameter are: • `BlockMessage`: blocks the message when malware is detected. • `ReplaceWithDefaultAlert`: delivers the message, but replaces the message contents with the default alert text when malware is detected. • `ReplaceWithCustomAlert`: delivers the message, but replaces the message contents with the custom alert text specified by the *AlertText* parameter when malware is detected. The default value is `BlockMessage`.
AdminDisplayName	Optional	System.String	The *AdminDisplayName* parameter specifies a description for the malware filter policy. If the value contains spaces, enclose the value in quotation marks (").
BypassInboundMessages	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013.

			The *BypassInboundMessages* parameter skips or enforces malware scanning on incoming messages. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. This means malware scanning occurs on incoming messages by default.
BypassOutboundMessages	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *BypassOutboundMessages* parameter skips or enforces malware scanning on outgoing messages. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. This means malware scanning occurs on outgoing messages by default.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CustomAlertText	Optional	System.String	The *CustomAlertText* parameter specifies the custom alert text to insert in the message when malware is detected and the value of the *Action* parameter is set to `ReplaceWithCustomAlert`. This parameter is required when the *CustomNotifications* parameter is set to `$true`.
CustomExternalBody	Optional	System.String	The *CustomExternalBody* parameter specifies the body of the custom notification message that's sent to an external sender when a message contains malware. This parameter is required when the *CustomNotifications* parameter is set to `$true`.
CustomExternalSubject	Optional	System.String	The *CustomExternalSubject* parameter specifies the

			subject of the custom notification message that's sent to an external sender when a message contains malware. This parameter is required when the *CustomNotifications* parameter is set to `$true`.
CustomFromAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *CustomFromAddress* parameter specifies the From address of the custom notification message that's sent to an internal or external sender when a message contains malware. This parameter is required when the *CustomNotifications* parameter is set to `$true`.
CustomFromName	Optional	System.String	The *CustomExternalFromName* parameter specifies the From name of the custom notification message that's sent to internal or external senders when a message contains malware. This parameter is required when the *CustomNotifications* parameter is set to `$true`.

CustomInternalBody	Optional	System.String	The *CustomInternalBody* parameter specifies the body of the custom notification message that's sent to an internal sender when a message contains malware. This parameter is required when the *CustomNotifications* parameter is set to `$true`.
CustomInternalSubject	Optional	System.String	The *CustomInternalSubject* parameter specifies the subject of the custom notification message that's sent to an internal sender when a message contains malware. This parameter is required when the *CustomNotifications* parameter is set to `$true`.
CustomNotifications	Optional	System.Boolean	The *CustomNotifications* parameter enables or disables the custom notification message to the sender when the message contains malware. Valid input for this parameter is `$true` or `$false`. The default value

			is `$false`. If you enable custom notification messages by setting this parameter to `$true`, you specify the details of the custom notification message using the *CustomFromAddress* ,*CustomFromName*, *CustomExternalSubject*, *CustomExternalBody*, *CustomInternalSubject*, and *CustomInternalBody* parameters.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EnableExternalSenderAdminNotifications	Optional	System.Boolean	The *EnableExternalSenderAdminNotifications* parameter enables or disables sending notification messages to

			an administrator when malware is detected in messages from external senders. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. Specify the administrator to receive the notification messages by using the *ExternalSenderAdminAddress* parameter.
EnableExternalSenderNotifications	Optional	System.Boolean	The *EnableExternalSenderNotifications* parameter enables or disables sending notification messages to senders when malware is detected in messages from external senders. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
EnableInternalSenderAdminNotifications	Optional	System.Boolean	The *EnableExternalSenderAdminNotifications* parameter enables or disables sending notification messages to an administrator when malware is detected in

			messages from internal senders. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. Specify the administrator to receive the notification messages by using the *InternalSenderAdminAddress* parameter.
EnableInternalSenderNotifications	Optional	System.Boolean	The *EnableExternalSenderAdminNotifications* parameter enables or disables sending notification messages to senders when malware is detected in messages from internal senders. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
ExternalSenderAdminAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *ExternalSenderAdminAddress* parameter specifies the email address of the administrator who will receive notifications messages when messages from external senders contain malware.

			Notification messages are sent to the specified email address only if the *EnableExternalSenderAdminNotifications* parameter is set to `$true`.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
InternalSenderAdminAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *InternalSenderAdminAddress* parameter specifies the email address of the administrator who will receive notifications messages when messages from external senders contain malware. Notification messages are sent to the specified email address only if the *EnableInternalSenderAdminNotifications* parameter is set to `$true`.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MalwareFilterPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-19*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MalwareFilterPolicy** cmdlet to remove malware filter policies from your organization.

Note:
When a policy is removed and there are rules associated with it, the rules are not removed when the policy is removed. This is by design. If you want to remove the associated rules, you need to do this separately via the Remove-MalwareFilterRule cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MalwareFilterPolicy -Identity <MalwareFilterPolicyIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Force
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example removes the malware filter policy named Contoso Malware Filter Policy

```
Remove-MalwareFilterPolicy "Contoso Malware Filter Policy"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MalwareFilterPolicyIdParameter	The *Identity* parameter specifies the malware filter policy you want to remove. You can use any value that uniquely identifies the policy. For example, you can use the name, GUID, or distinguished name (DN) of the malware filter policy.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run.

			To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for

			administrative input. You don't have to specify a value with this parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MalwareFilterPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MalwareFilterPolicy** cmdlet to modify malware filter policies in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MalwareFilterPolicy -Identity <MalwareFilterPolicyIdParameter> [-
Action <DeleteMessage | DeleteAttachmentAndUseDefaultAlertText |
DeleteAttachmentAndUseCustomAlertText>] [-AdminDisplayName <String>] [-
BypassInboundMessages <$true | $false>] [-BypassOutboundMessages <$true |
$false>] [-Confirm [<SwitchParameter>]] [-CustomAlertText <String>] [-
CustomExternalBody <String>] [-CustomExternalSubject <String>] [-
CustomFromAddress <SmtpAddress>] [-CustomFromName <String>] [-
CustomInternalBody <String>] [-CustomInternalSubject <String>] [-
CustomNotifications <$true | $false>] [-DomainController <Fqdn>] [-
EnableExternalSenderAdminNotifications <$true | $false>] [-
EnableExternalSenderNotifications <$true | $false>] [-
EnableInternalSenderAdminNotifications <$true | $false>] [-
EnableInternalSenderNotifications <$true | $false>] [-
ExternalSenderAdminAddress <SmtpAddress>] [-IgnoreDehydratedFlag
<SwitchParameter>] [-InternalSenderAdminAddress <SmtpAddress>] [-
MakeDefault <SwitchParameter>] [-Name <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example modifies the malware filter policy named Contoso Malware Filter Policy with the following settings:

- Delete messages that contain malware.
- Don't notify the message sender when malware is detected in the message.
- Notify the administrator admin@contoso.com when malware is detected in a message from an internal sender.

```
Set-MalwareFilterPolicy -Identity "Contoso Malware Filter
Policy" -Action DeleteMessage -
EnableInternalSenderAdminNotifications $true -
InternalSenderAdminAddress admin@contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MalwareFilterPolicyIdParameter	The *Identity* parameter specifies the malware filter policy you want to modify. You can use any value that uniquely identifies the policy. For example, you can use the name, GUID or distinguished name (DN) of the malware filter policy.
Action	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MalwareFilteringAction	The *Action* parameter specifies the action to take when malware is detected in a message. Valid values for this parameter are: • `DeleteMessage`: deletes the message when malware is detected. • `ReplaceWithDefaultAlert`: delivers the message, but replaces the message contents with the default alert text when malware is detected. • `ReplaceWithCustomAlert`: delivers the message, but replaces the message contents with the custom alert text specified by the *AlertText* parameter when malware is detected. The default value is

			`DeleteMessage`.
AdminDisplayName	Optional	System.String	The *AdminDisplayName* parameter specifies a description for the malware filter policy. If the value contains spaces, enclose the value in quotation marks (").
BypassInboundMessages	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *BypassInboundMessages* parameter skips or enforces malware scanning on incoming messages. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. This means malware scanning occurs on incoming messages by default.
BypassOutboundMessages	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *BypassOutboundMessages* parameter skips or enforces malware scanning on outgoing

			messages. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. This means malware scanning occurs on outgoing messages by default.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CustomAlertText	Optional	System.String	The *CustomAlertText* parameter specifies the custom alert text to insert in the message when malware is detected and the value of the *Action* parameter is set to `ReplaceWithCustomAlert`. This parameter is required when the *CustomNotifications* parameter is set to `$true`.
CustomExternalBody	Optional	System.String	The *CustomExternalBody* parameter specifies the body of the custom

			notification message that's sent to an external sender when a message contains malware. This parameter is required when any of the following parameters are set to `$true`: • *CustomNotifications* • *EnableExternalSenderAdminNotifications* • *EnableExternalSenderNotifications*
CustomExternalSubject	Optional	System.String	The *CustomExternalSubject* parameter specifies the subject of the custom notification message that's sent to an external sender when a message contains malware. This parameter is required when any of the following parameters are set to `$true`: • *CustomNotifications* • *EnableExternalSenderAdminNotifications* • *EnableExternalSenderNotifications*
CustomFromAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *CustomFromAddress* parameter specifies the

			From address of the custom notification message that's sent to an internal or external sender when a message contains malware. This parameter is required when the *CustomNotifications* parameter is set to `$true`.
CustomFromName	Optional	System.String	The *CustomFromName* parameter specifies the From name of the custom notification message that's sent to internal or external senders when a message contains malware. This parameter is required when the *CustomNotifications* parameter is set to `$true`.
CustomInternalBody	Optional	System.String	The *CustomInternalBody* parameter specifies the body of the custom notification message that's sent to an internal sender when a message contains malware. This parameter is required when any of the following parameters are set to `$true`: • *CustomNotifications*

			• *EnableExternalSenderAdminNotifications* • *EnableExternalSenderNotifications*
CustomInternalSubject	Optional	System.String	The *CustomInternalSubject* parameter specifies the subject of the custom notification message that's sent to an internal sender when a message contains malware. This parameter is required when any of the following parameters are set to `$true`: • *CustomNotifications* • *EnableExternalSenderAdminNotifications* • *EnableExternalSenderNotifications*
CustomNotifications	Optional	System.Boolean	The *CustomNotifications* parameter enables or disables the custom notification message to the sender when the message contains malware. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you enable custom

			notification messages by setting this parameter to `$true`, you specify the details of the custom notification message using the *CustomFromAddress* ,*CustomFromName*, *CustomExternalSubject*, *CustomExternalBody*, *CustomInternalSubject*, and *CustomInternalBody* parameters.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EnableExternalSenderAdminNotifications	Optional	System.Boolean	The *EnableExternalSenderAdminNotifications* parameter enables or disables sending notification messages to an administrator when malware is detected in

			messages from external senders. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. Specify the administrator to receive the notification messages by using the *ExternalSenderAdminAddress* parameter.
EnableExternalSenderNotifications	Optional	System.Boolean	The *EnableExternalSenderNotifications* parameter enables or disables sending notification messages to senders when malware is detected in messages from external senders. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
EnableInternalSenderAdminNotifications	Optional	System.Boolean	The *EnableExternalSenderAdminNotifications* parameter enables or disables sending notification messages to an administrator when malware is detected in messages from internal senders. Valid input for

			this parameter is `$true` or `$false`. The default value is `$false`. Specify the administrator to receive the notification messages by using the *InternalSenderAdminAddress* parameter.
EnableInternalSenderNotifications	Optional	System.Boolean	The *EnableExternalSenderAdminNotifications* parameter enables or disables sending notification messages to senders when malware is detected in messages from internal senders. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
ExternalSenderAdminAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *ExternalSenderAdminAddress* parameter specifies the email address of the administrator who will receive notifications messages when messages from external senders contain malware. Notification messages are sent to the specified email

			address only if the *EnableExternalSenderAdminNotifications* parameter is set to `$true`.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
InternalSenderAdminAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *InternalSenderAdminAddress* parameter specifies the email address of the administrator who will receive notifications messages when messages from external senders contain malware. Notification messages are sent to the specified email address only if the *EnableInternalSenderAdminNotifications* parameter is set to `$true`.
MakeDefault	Optional	System.Management.Automation.SwitchParameter	The *MakeDefault* switch makes this the default malware filter policy. You don't have to specify a value with this switch.
Name	Optional	System.String	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.	The *WhatIf* switch

		Automation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-MalwareFilterRule

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-MalwareFilterRule** cmdlet to disable malware filter rules in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-MalwareFilterRule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example disables the enabled malware filter rule named Contoso Recipients.

```
Disable-MalwareFilterRule "Contoso Recipients"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the malware filter rule that you want to disable. You can use any value that uniquely identifies the rule. For example, you can use the name, GUID, or distinguished name (DN) of the malware filter rule.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do

			before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-MalwareFilterRule

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-MalwareFilterRule** cmdlet to enable malware filter rules in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-MalwareFilterRule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example enables the disabled malware filter rule named Contoso Recipients.

```
Enable-MalwareFilterRule "Contoso Recipients"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the malware filter rule that you want to enable. You can use any value that uniquely identifies the rule. For example, you can use the name, GUID or distinguished name (DN) of the malware filter rule.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MalwareFilterRule

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-MalwareFilterRule** cmdlet to view malware filter rules in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MalwareFilterRule [-Identity <RuleIdParameter>] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>] [-State <Enabled |
Disabled>]
```

Examples

Example 1

This example retrieves a summary list of all malware filter rules in your organization.

```
Get-MalwareFilterRule
```

Example 2

This example retrieves detailed configuration information for the malware filter rule named Contoso Recipients.

```
Get-MalwareFilterRule "Contoso Recipients" | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the malware filter rule that you want to view. You can use any value that uniquely identifies the rule. For example, you can use the name, GUID or distinguished name (DN) of the malware filter rule.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
State	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleState	The *State* parameter filters the results by enabled or disabled malware filter rules. Valid input for this parameter is `Enabled` or `Disabled`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MalwareFilterRule

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MalwareFilterRule** cmdlet to create malware filter rules in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MalwareFilterRule -MalwareFilterPolicy
<MalwareFilterPolicyIdParameter> -Name <String> [-Comments <String>] [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-ExceptIfRecipientDomainIs <Word[]>] [-ExceptIfSentTo
<RecipientIdParameter[]>] [-ExceptIfSentToMemberOf <RecipientIdParameter[]
>] [-Organization <OrganizationIdParameter>] [-Priority <Int32>] [-
RecipientDomainIs <Word[]>] [-SentTo <RecipientIdParameter[]>] [-
SentToMemberOf <RecipientIdParameter[]>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example creates a new malware filter rule named Contoso Recipients with the following settings: If the recipient is in the domain contoso.com, apply the malware filter policy named Contoso Malware Filter Policy.

```
New-MalwareFilterRule -Name "Contoso Recipients" -
MalwareFilterPolicy "Contoso Malware Filter Policy" -
RecipientDomainIs contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
MalwareFilterPolicy	Required	Microsoft.Exchange.Configuration.Tasks.MalwareFilterPolicyIdParameter	The *MalwareFilterPolicy* parameter specifies the malware filter policy to apply to messages that match the conditions defined by this malware filter rule. You can use any value that uniquely identifies the policy. For example, you can specify the name, GUID, or distinguished name (DN) of the content filter policy. **Note:** You can't specify the default malware filter policy. And, you can't specify a malware filter policy that's already associated with another malware filter rule.
Name	Required	System.String	The *Name* parameter specifies a unique name for the malware filter rule.

Comments	Optional	System.String	The *Comments* parameter specifies informative comments for the rule, such as what the rule is used for or how it has changed over time. The length of the comment can't exceed 1024 characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

Enabled	Optional	System.Boolean	The *Enabled* parameter enables or disables the malware filter rule. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
ExceptIfRecipientDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfRecipientDomainIs* parameter specifies the recipient's domain. The rule isn't applied to messages sent to recipients whose email addresses are in the specified domain.
ExceptIfSentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfSentTo* parameter specifies a recipient. The rule isn't applied to messages sent to the specified recipient.
ExceptIfSentToMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfSentToMemberOf* parameter specifies a distribution group. The rule isn't applied to messages where any recipient is a member of the specified group. **Note:** If the distribution group is removed after

			creation of the rule, no exception is made.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Priority	Optional	System.Int32	The *Priority* parameter specifies the priority for this rule. Rules with a lower priority value are processed first. If you modify the priority of the rule, the position of the rule in the rule list changes to match the priority that you specified. The value of this parameter must be greater than or equal to 0, and must be one less than the total number of rules in your organization. For example, if you configured 8 rules, you can set this parameter to any value from 0 through 7.
RecipientDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *RecipientDomainIs* parameter specifies the recipient's domain. The

			rule is applied to messages sent to recipients whose email addresses are in the specified domain. This parameter is used to define a rule condition.
SentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentTo* parameter specifies a recipient. The rule is applied to messages sent to the specified recipient. This parameter is used to define a rule condition.
SentToMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentToMemberOf* parameter specifies a distribution group. The rule is applied to messages where any recipient is a member of the specified group. **Note:** If the distribution group is removed after creation of the rule, no action is taken. This parameter is used to define a rule condition.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MalwareFilterRule

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MalwareFilterRule** cmdlet to remove malware filter rules from your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MalwareFilterRule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example removes the malware filter rule named Contoso Recipients.

```
Remove-MalwareFilterRule "Contoso Recipients"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the malware filter rule that you want to remove. You can use any value that uniquely identifies the rule. For example, you can use the name, GUID or distinguished name (DN) of the malware filter rule.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default

			when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MalwareFilterRule

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MalwareFilterRule** cmdlet to modify malware filter rules in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MalwareFilterRule -Identity <RuleIdParameter> [-Comments <String>] [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
ExceptIfRecipientDomainIs <Word[]>] [-ExceptIfSentTo
<RecipientIdParameter[]>] [-ExceptIfSentToMemberOf <RecipientIdParameter[]
>] [-MalwareFilterPolicy <MalwareFilterPolicyIdParameter>] [-Name
<String>] [-Priority <Int32>] [-RecipientDomainIs <Word[]>] [-SentTo
<RecipientIdParameter[]>] [-SentToMemberOf <RecipientIdParameter[]>] [-
WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example adds an exception to the malware filter rule named Contoso Recipients for members of the distribution group named Contoso Human Resources.

```
Set-MalwareFilterRule "Contoso Recipients" -
ExceptIfSentToMemberOf "Contoso Human Resources"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-malware" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the malware filter rule that you want to view. You can use any value that uniquely identifies the rule. For example, you can use the name, GUID, or distinguished name (DN) of the malware filter rule.
Comments	Optional	System.String	The *Comments* parameter specifies informative comments for the rule, such as what the rule is used for or how it has changed over time. The length of the comment can't exceed 1024 characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExceptIfRecipientDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfRecipientDomainIs* parameter specifies the recipient's domain. The rule isn't applied to messages sent to recipients whose email addresses are in the specified domain.
ExceptIfSentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfSentTo* parameter specifies a recipient. The rule isn't

			applied to messages sent to the specified recipient.
ExceptIfSentToMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfSentToMemberOf* parameter specifies a distribution group. The rule isn't applied to messages where any recipient is a member of the specified group. **Note:** If the distribution group is removed after creation of the rule, no exception is made.
MalwareFilterPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MalwareFilterPolicyIdParameter	The *MalwareFilterPolicy* parameter specifies the malware filter policy to apply to messages that match the conditions defined by this malware filter rule. You can use any value that uniquely identifies the policy. For example, you can specify the name, GUID, or distinguished name (DN) of the content filter policy. **Note:** You can't specify the

			default malware filter policy. And, you can't specify a malware filter policy that's already associated with another malware filter rule.
Name	Optional	System.String	The *Name* parameter specifies a unique name for the malware filter rule.
Priority	Optional	System.Int32	The *Priority* parameter specifies the priority for this rule. Rules with a lower priority value are processed first. If you modify the priority of the rule, the position of the rule in the rule list changes to match the priority that you specified. The value of this parameter must be greater than or equal to 0, and must be one less than the total number of rules in your organization. For example, if you configured 8 rules, you can set this parameter to any value from 0 through 7.

RecipientDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *RecipientDomainIs* parameter specifies the recipient's domain. The rule is applied to messages sent to recipients whose email addresses are in the specified domain. This parameter is used to define a rule condition.
SentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentTo* parameter specifies a recipient. The rule is applied to messages sent to the specified recipient. This parameter is used to define a rule condition.
SentToMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentToMemberOf* parameter specifies a distribution group. The rule is applied to messages where any recipient is a member of the specified group. **Note:** If the distribution group is removed after creation of the rule, no action is taken. This parameter is used

			to define a rule condition.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-RecipientFilterConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-RecipientFilterConfig** cmdlet to view the recipient filter configuration information for the computer on which the command is run.

```
Get-RecipientFilterConfig [-Identity <OrganizationIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the recipient filter configuration for the computer on which the command is run.

```
Get-RecipientFilterConfig | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance

			of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-RecipientFilterConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-RecipientFilterConfig** cmdlet to enable and configure the Recipient Filter agent.

```
Set-RecipientFilterConfig [-BlockedRecipients <MultiValuedProperty>] [-
BlockListEnabled <$true | $false>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Enabled <$true | $false>] [-ExternalMailEnabled
<$true | $false>] [-InternalMailEnabled <$true | $false>] [-
RecipientValidationEnabled <$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the Recipient Filter agent configuration so that recipient validation is enabled.

```
Set-RecipientFilterConfig -RecipientValidationEnabled $true
```

EXAMPLE 2

This example makes the following changes to the Recipient Filter agent configuration:

- Enables the Blocked Recipients list.
- Adds two users to the Blocked Recipients list.

```
Set-RecipientFilterConfig -BlockListEnabled $true -
BlockedRecipients user1@contoso.com,user2@contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
BlockedRecipients	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BlockedRecipients* parameter specifies one or more SMTP addresses. To enter multiple SMTP addresses, separate the addresses by using a comma. The maximum number of individual SMTP addresses that you can input is 800.
BlockListEnabled	Optional	System.Boolean	The *BlockListEnabled* parameter specifies whether the Recipient

			Filter agent blocks messages sent to recipients listed in the *BlockedRecipients* parameter. Valid input for the *BlockListEnabled* parameter is `$true` or `$false`. The default setting is `$false`. When the *BlockListEnabled* parameter is set to `$true`, the Recipient Filter agent blocks messages sent to recipients listed in the *BlockedRecipients* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the Recipient Filter agent is enabled on the computer on which you're running the command. Valid input for the *Enabled* parameter is `$true` or `$false`. The default setting is `$true`. When the *Enabled* parameter is set to `$true`, the Recipient Filter agent is enabled on the computer on which you're running the

			command.
ExternalMailEnabled	Optional	System.Boolean	The *ExternalMailEnabled* parameter specifies whether all messages received from unauthenticated connections by servers external to your organization are passed through the Recipient Filter agent for processing. Valid input for the *ExternalMailEnabled* parameter is `$true` or `$false`. The default setting is `$true`. When the *ExternalMailEnabled* parameter is set to `$true`, all messages received from unauthenticated connections by servers external to your organization are passed through the Recipient Filter agent for processing.
InternalMailEnabled	Optional	System.Boolean	The *InternalMailEnabled*

			parameter specifies whether all messages from authenticated sender domains that belong to authoritative domains in the enterprise are passed through the Recipient Filter agent for processing. Valid input for the *InternalMailEnabled* parameter is `$true` or `$false`. The default setting is `$false`. When the *InternalMailEnabled* parameter is set to `$true`, all messages from authenticated sender domains that belong to authoritative domains in the enterprise are passed through the Recipient Filter agent for processing.
RecipientValidationEnabled	Optional	System.Boolean	The *RecipientValidationEnabled* parameter specifies whether the Recipient Filter agent

			blocks messages addressed to recipients that don't exist in the organization. Valid input for the *RecipientValidationEnabled* parameter is `$true` or `$false`. The default setting is `$false`. When the *RecipientValidationEnabled* parameter is set to `$true`, the Recipient Filter agent blocks messages addressed to recipients that don't exist in the organization.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-SafeList

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Update-SafeList** cmdlet to update the safelist aggregation data in Active Directory. Safelist aggregation data is used in the built-in anti-spam filtering in Microsoft Exchange. EdgeSync replicates safelist aggregation data to Edge Transport servers in the perimeter network.

```
Update-SafeList -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-EnsureJunkEmailRule
<SwitchParameter>] [-IncludeDomains <SwitchParameter>] [-Type <SafeSenders
| SafeRecipients | Both | BlockedSenders | All>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example updates Safe Senders List data for the single user kim@contoso.com.

```
Update-Safelist kim@contoso.com
```

EXAMPLE 2

This example updates safelist data for all user mailboxes in your Exchange organization. By default, the Exchange Management Shell is configured to retrieve or modify objects that reside in the domain in which the Exchange server resides. Therefore, to retrieve all the mailboxes in your Exchange organization, you must first set the scope of the Shell to the entire forest using the **Set-**

AdServerSettings cmdlet. For more information, see Set-AdServerSettings.

```
Set-AdServerSettings -ViewEntireForest $true
Get-Mailbox -ResultSize Unlimited -RecipientTypeDetails
UserMailbox | Update-Safelist
```

Detailed Description

The **Update-SafeList** cmdlet reads the safelist aggregation data stored on a Microsoft Outlook user mailbox and then hashes and writes the data to the corresponding user object in Active Directory. The command compares the binary attribute created to any value stored on the attribute. If the two values are identical, the command doesn't update the user attribute value with the safelist aggregation data. Safelist aggregation data contains the Outlook user's Safe Senders List and Safe Recipients List.

Be mindful of the network and replication traffic that may be generated when you run the **Update-SafeList** cmdlet. Running the command on multiple mailboxes where safelists are heavily used may generate a significant amount of traffic. We recommend that if you run the command on multiple mailboxes, you should run the command during off-peak, non-business hours.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "SafeList aggregation" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox from which you want to collect safelist aggregation data. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name`

			Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EnsureJunkEmailRule	Optional	System.Management.Automation.SwitchParameter	The *EnsureJunkEmailRule* parameter specifies whether to force the junk email rule to be turned on for the mailbox if the rule isn't turned on already. **Note:** The junk email rule can only be created after the user logs on to the mailbox. If the user has never logged on to the mailbox, this parameter can't turn on the junk email rule.
IncludeDomains	Optional	System.Management.Automation.SwitchParameter	The *IncludeDomains* switch specifies whether to include the sender domains specified by users in Outlook in the safelist aggregation data. By default, domains specified by the senders aren't included.

			In most cases, we don't recommend that you include domains because users may include the domains of large Internet service providers (ISPs), which could unintentionally provide addresses that may be used or spoofed by spammers.
Type	Optional	Microsoft.Exchange.Management.RecipientTasks.UpdateType	The *Type* parameter specifies which user-generated list is updated to the user object. Valid values for this parameter are `SafeSenders`, `SafeRecipients`, and `Both`. The default value is `SafeSenders`. **Important:** The safelist aggregation feature doesn't act on Safe Recipients List data. We don't recommend running the *Type* parameter with the `SafeRecipients` or `Both` values.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SenderFilterConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-SenderFilterConfig** cmdlet to view the Sender Filter configuration information for the computer on which the command is run.

```
Get-SenderFilterConfig [-Identity <OrganizationIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the Sender Filter configuration for the computer on which the command is run.

```
Get-SenderFilterConfig | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-SenderFilterConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-SenderFilterConfig** cmdlet to modify the Sender Filter agent configuration.

```
Set-SenderFilterConfig [-Action <StampStatus | Reject>] [-
BlankSenderBlockingEnabled <$true | $false>] [-BlockedDomains
<MultiValuedProperty>] [-BlockedDomainsAndSubdomains
<MultiValuedProperty>] [-BlockedSenders <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-ExternalMailEnabled <$true | $false>] [-InternalMailEnabled
<$true | $false>] [-RecipientBlockedSenderAction <Reject | Delete>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example makes the following modifications to the Sender Filter agent configuration:

- It enables blocking of blank senders.
- It blocks messages from lucernepublishing.com and all subdomains.
- It adds user1@contoso.com and user2@contoso.com to the blocked senders list without affecting any existing entries.

```
Set-SenderFilterConfig -BlankSenderBlockingEnabled $true -
BlockedDomainsAndSubdomains lucernepublishing.com -
BlockedSenders
@{Add="user1@contoso.com","user2@contoso.com"}
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Action	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.BlockedSenderAction	The *Action* parameter specifies the action that the Sender Filter agent takes on messages from blocked senders or domains. Valid input for this parameter is `StampStatus` or `Reject`. The default value is `Reject`.
BlankSenderBlockingEnabled	Optional	System.Boolean	The *BlankSenderBlockingEnabled* parameter blocks or allows messages that don't contain a sender value in the SMTP command **MAIL FROM**. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
BlockedDomains	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BlockedDomains* parameter specifies the

			domain names to block. When the Sender Filter agent encounters a message from a domain on this list, the Sender Filter agent takes the action specified by the *Action* parameter. Valid input for this parameter is one or more domains or subdomains. Wildcard characters aren't permitted. For example, if you specify the values contoso.com and marketing.contoso.com, only messages from those domains are blocked by the Sender Filter agent. Messages from sales.contoso.com aren't blocked by the Sender Filter agent. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise

			require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...`. To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. The maximum number of entries for this parameter is 800.
BlockedDomainsAndSubdomains	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BlockedDomainsAndSubdomains* parameter specifies the domain names to block. When the Sender Filter agent encounters a message from a domain on this list, or from any of the domain's subdomains, the Sender Filter agent takes the action specified by the *Action* parameter. Valid input for this parameter is one or more domains.

			Wildcard characters aren't permitted. For example, if you specify the value contoso.com, messages from contoso.com, sales.contoso.com, and all other subdomains of contoso.com are blocked by the Sender Filter agent. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...`. To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. The maximum number of entries for this

			parameter is 800.
BlockedSenders	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BlockedSenders* parameter specifies one or more SMTP email addresses to block. When the Sender Filter agent encounters a message from a sender on this list, the Sender Filter agent takes the action specified by the *Action* parameter. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...`. To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.

			The maximum number of entries for this parameter is 800.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Enabled	Optional	System.Boolean	The *Enabled* parameter enables or disables sender filtering on your Exchange server. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
ExternalMailEnabled	Optional	System.Boolean	The *ExternalMailEnabled* parameter enables or disables sender filtering on unauthenticated connections from external messaging servers. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
InternalMailEnabled	Optional	System.Boolean	The *InternalMailEnabled* parameter enables or disables sender filtering on authenticated connections from authoritative domains in your organization. Valid input for this parameter is `$true` or `$false`. The default

			value is `$false`.
RecipientBlockedSenderAction	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.RecipientBlockedSenderAction	The *RecipientBlockedSenderAction* parameter specifies the action that the Sender Filter agent takes on messages received from blocked senders that are defined by SafeList aggregation. SafeList aggregation adds blocked senders that are defined by your users in Microsoft Outlook or Outlook Web App to the Blocked Senders list that's used by the Sender Filter agent. Valid input for this parameter is `Delete` or `Reject`. The default value is `Reject`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-SenderId

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-SenderId** cmdlet to test whether a specified IP address is the legitimate sending address for a specified SMTP address.

```
Test-SenderId -IPAddress <IPAddress> -PurportedResponsibleDomain
<SmtpDomain> [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
HelloDomain <String>] [-Server <ServerIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example checks whether the IP address 192.168.0.1 is the legitimate sender address for the domain contoso.com.

```
Test-SenderId -IPAddress 192.168.0.1 -
PurportedResponsibleDomain contoso.com
```

Detailed Description

The **Test-SenderId** cmdlet provides the results of a Sender ID check for the IP address and the corresponding domain name that you specify.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
IPAddress	Required	System.Net.IPAddress	The *IPAddress* parameter specifies the originating IP address of the sending server.
PurportedResponsibleDomain	Required	Microsoft.Exchange.Data.SmtpDomain	The *PurportedResponsibleDomain* parameter specifies the domain name that you want to verify with Sender ID.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
HelloDomain	Optional	System.String	The *HelloDomain* parameter specifies the domain address displayed in the HELO or EHLO SMTP commands from this sender.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example:

			• Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SenderIdConfig

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-SenderIdConfig** cmdlet to view the Sender ID configuration information for the computer on which the command is run.

```
Get-SenderIdConfig [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the Sender ID configuration for the computer on which the command is run.

```
Get-SenderIdConfig | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't

			supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-SenderIdConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-SenderIdConfig** cmdlet to modify the configuration of the Sender ID agent.

```
Set-SenderIdConfig [-BypassedRecipients <MultiValuedProperty>] [-
BypassedSenderDomains <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-ExternalMailEnabled <$true | $false>] [-InternalMailEnabled
<$true | $false>] [-SpoofedDomainAction <StampStatus | Reject | Delete>]
[-TempErrorAction <StampStatus | Reject | Delete>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example makes the following modifications to the Sender ID configuration:

- It sets the Sender ID agent to delete all messages sent from spoofed domains.
- It specifies two recipients for the Sender ID agent to exclude when it processes messages.

```
Set-SenderIdConfig -SpoofedDomainAction Delete -
BypassedRecipients user1@contoso.com,user2@contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
BypassedRecipients	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassedRecipients* parameter specifies one or more SMTP email addresses. Messages bound for the email addresses listed in this parameter are excluded from processing by the Sender ID agent. You can specify multiple values separated by commas. You can enter a maximum of 100 email addresses.
BypassedSenderDomains	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassedSenderDomai*

			ns parameter specifies one or more domain names. Messages that originate from the domains listed in this parameter are excluded from processing by the Sender ID agent. You can specify multiple values separated by commas. You can enter a maximum of 100 domain names.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

			The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the Sender ID agent is enabled on the computer on which you're running the command. Valid input for the *Enabled* parameter is `$true` or `$false`. The default setting is `$true`. When the *Enabled* parameter is set to `$true`, the Sender ID agent is enabled on the computer on which you're running the command.
ExternalMailEnabled	Optional	System.Boolean	The *ExternalMailEnabled* parameter specifies whether all messages

			from unauthenticated connections external to your organization are passed through the Sender ID agent for processing. Valid input for the *ExternalMailEnabled* parameter is `$true` or `$false`. The default setting is `$true`. When the *ExternalMailEnabled* parameter is set to `$true`, all messages from unauthenticated connections external to your organization are passed through the Sender ID agent for processing.
InternalMailEnabled	Optional	System.Boolean	The *InternalMailEnabled* parameter specifies whether all messages from authenticated sender domains that belong to authoritative domains in your enterprise are passed through the Sender ID agent for processing.

			Valid input for the *InternalMailEnabled* parameter is `$true` or `$false`. The default setting is `$false`. When the *InternalMailEnabled* parameter is set to `$true`, all messages from authenticated sender domains that belong to authoritative domains in your enterprise are passed through the Sender ID agent for processing.
SpoofedDomainAction	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.SenderIdAction	The *SpoofedDomainAction* parameter specifies the action that the Sender ID agent takes on the message when the sender domain shows evidence of being spoofed. The *SpoofedDomainAction* parameter takes the following values: `StampStatus`, `Reject`, or `Delete`. The default value is `StampStatus`.
TempErrorAction	Optional	Microsoft.Exchange.Dat	The *TempErrorAction*

		a.Directory.SystemConfiguration.SenderIdAction	parameter specifies the action that the Sender ID agent takes on the message when a Sender ID status of TempError is returned. The *TempErrorAction* parameter takes the following values: `StampStatus`, `Reject`, or `Delete`. The default value is `StampStatus`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SenderReputationConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-SenderReputationConfig** cmdlet to view the configuration information for sender reputation on the computer on which the command is run.

```
Get-SenderReputationConfig [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the sender reputation configuration for the computer on which the command is run.

```
Get-SenderReputationConfig | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-SenderReputationConfig

Exchange Management Shell > Exchange 2013 cmdlets > Anti-spam and anti-malware cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-SenderReputationConfig** cmdlet to modify the sender reputation configuration on a Mailbox server or an Edge Transport server.

```
Set-SenderReputationConfig [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Enabled <$true | $false>] [-ExternalMailEnabled
<$true | $false>] [-InternalMailEnabled <$true | $false>] [-
OpenProxyDetectionEnabled <$true | $false>] [-ProxyServerName <String>] [-
ProxyServerPort <Int32>] [-ProxyServerType <None | Socks4 | Socks5 |
HttpConnect | HttpPost | Telnet | Cisco | Wingate>] [-
SenderBlockingEnabled <$true | $false>] [-SenderBlockingPeriod <Int32>] [-
SrlBlockThreshold <Int32>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example makes the following modifications to the sender reputation configuration:

- It sets the sender reputation action to block all senders whose sender reputation level (SRL) rating exceeds the SRL threshold.
- It sets the SRL blocking threshold to 6.
- It sets the number of hours that senders are put on the blocked senders list to 36 hours.

```
Set-SenderReputationConfig -SenderBlockingEnabled $true -
SrlBlockThreshold 6 -SenderBlockingPeriod 36
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Anti-spam features" entry in the Anti-spam and anti-malware permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter enables or disables sender reputation on your Exchange server. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
ExternalMailEnabled	Optional	System.Boolean	The *ExternalMailEnabled* parameter allows or prevents sender

			reputation from processing all messages from unauthenticated connections that are external to your Exchange organization. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
InternalMailEnabled	Optional	System.Boolean	The *InternalMailEnabled* parameter allows or prevents sender reputation from processing all messages from authenticated sender domains that are authoritative domains in your Exchange organization. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
OpenProxyDetectionEnabled	Optional	System.Boolean	The *OpenProxyDetectionEnabled* parameter allows or prevents sender reputation from

			connecting to the source IP address to determine if the sender is an open proxy. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. The values of the *OpenProxyDetectionEnabled* and *SenderBlockingEnabled* parameters can both be set to `$true`, but they both can't be set to `$false`. If one value is `$true` and the other is `$false`, and you change the `$true` value to `$false`, the parameter that was previously `$false` will automatically change to `$true`.
ProxyServerName	Optional	System.String	The *ProxyServerName* parameter specifies the name of your organization's proxy server. Sender reputation uses this parameter to connect to the Internet.

ProxyServerPort	Optional	System.Int32	The *ProxyServerPort* parameter specifies the port number that's used by your organization's proxy server. Sender reputation uses this parameter to connect to the Internet.
ProxyServerType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ProxyType	The *ProxyServerType* parameter classifies your organization's proxy server. Sender reputation uses this parameter to connect to the Internet. Valid input for this parameter is `None`, `Socks4`, `Socks5`, `HttpConnect`, `HttpPost`, `Telnet`, `Cisco`, or `Wingate`. The default value is `None`.
SenderBlockingEnabled	Optional	System.Boolean	The *SenderBlockingEnabled* parameter allows or prevents sender reputation from blocking senders when the source server fails an open proxy test. Valid input for this

			parameter is `$true` or `$false`. The default value is `$true`. You can temporarily block senders for up to 48 hours when you use the *SenderBlockingPeriod* parameter. The values of the *OpenProxyDetectionEnabled* and *SenderBlockingEnabled* parameters can both be set to `$true`, but they both can't be set to `$false`. If one value is `$true` and the other is `$false`, and you change the `$true` value to `$false`, the parameter that was previously `$false` will automatically change to `$true`.
SenderBlockingPeriod	Optional	System.Int32	The *SenderBlockingPeriod* parameter specifies the number of hours that a sender remains on the blocked senders list when their source IP

			address fails the open proxy test. Valid input for this parameter is an integer from 0 through 48. The default value is 24.
SrlBlockThreshold	Optional	System.Int32	The *SrlBlockThreshold* specifies the sender reputation level (SRL) rating that must be exceeded for sender reputation to block a sender. Valid input for this parameter is an integer value from 0 through 9. The default value is 7.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Client Access cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-28*

Autodiscover cmdlets

Export-AutoDiscoverConfig

Get-AutodiscoverVirtualDirectory

New-AutodiscoverVirtualDirectory

Remove-AutodiscoverVirtualDirectory

Set-AutodiscoverVirtualDirectory

New-OutlookProvider

Remove-OutlookProvider

Get-OutlookProvider

Set-OutlookProvider

Client Access mailbox settings cmdlets

Get-CASMailbox

Set-CASMailbox

Client Access server cmdlets

Get-AuthRedirect

New-AuthRedirect

Remove-AuthRedirect

Set-AuthRedirect

Get-ClientAccessArray

Get-ClientAccessServer

Set-ClientAccessServer

Exchange ActiveSync and mobile device cmdlets

Mobile device cmdlets

Clear-MobileDevice

Get-MobileDevice

Remove-MobileDevice

Get-MobileDeviceMailboxPolicy

New-MobileDeviceMailboxPolicy

Remove-MobileDeviceMailboxPolicy

Set-MobileDeviceMailboxPolicy

Get-MobileDeviceStatistics

Exchange ActiveSync cmdlets

Note:
The **ActiveSyncMailboxPolicy** cmdlets have been replaced by the **MobileDeviceMailboxPolicy** cmdlets.

Test-ActiveSyncConnectivity

Clear-ActiveSyncDevice

Get-ActiveSyncDevice

Remove-ActiveSyncDevice

Get-ActiveSyncDeviceAccessRule

New-ActiveSyncDeviceAccessRule

Remove-ActiveSyncDeviceAccessRule

Set-ActiveSyncDeviceAccessRule

Get-ActiveSyncDeviceAutoblockThreshold

Set-ActiveSyncDeviceAutoblockThreshold

Get-ActiveSyncDeviceClass

Remove-ActiveSyncDeviceClass

Get-ActiveSyncDeviceStatistics

Export-ActiveSyncLog

Get-ActiveSyncMailboxPolicy

New-ActiveSyncMailboxPolicy

Remove-ActiveSyncMailboxPolicy

Set-ActiveSyncMailboxPolicy

Get-ActiveSyncOrganizationSettings

Set-ActiveSyncOrganizationSettings

Get-ActiveSyncVirtualDirectory

New-ActiveSyncVirtualDirectory

Remove-ActiveSyncVirtualDirectory

Set-ActiveSyncVirtualDirectory

OWA for Devices cmdlets

Disable-PushNotificationProxy

Enable-PushNotificationProxy

Text messaging cmdlets

Clear-TextMessagingAccount

Get-TextMessagingAccount

Set-TextMessagingAccount

Compare-TextMessagingVerificationCode

Send-TextMessagingVerificationCode

Exchange admin center (EAC) and Exchange Control Panel (ECP) virtual directory cmdlets

Test-EcpConnectivity

Get-EcpVirtualDirectory

New-EcpVirtualDirectory

Remove-EcpVirtualDirectory

Set-EcpVirtualDirectory

Exchange Web Services virtual directory cmdlets

Test-WebServicesConnectivity

Get-WebServicesVirtualDirectory

New-WebServicesVirtualDirectory

Remove-WebServicesVirtualDirectory

Set-WebServicesVirtualDirectory

Outlook connectivity cmdlets

Get-MapiVirtualDirectory

New-MapiVirtualDirectory

Remove-MapiVirtualDirectory

Set-MapiVirtualDirectory

Get-OutlookAnywhere

Set-OutlookAnywhere

Test-OutlookConnectivity

Get-RpcClientAccess

Set-RpcClientAccess

Outlook Web App cmdlets

Test-CalendarConnectivity

Get-MailboxCalendarConfiguration

Set-MailboxCalendarConfiguration

Get-MailboxMessageConfiguration

Set-MailboxMessageConfiguration

Get-MailboxRegionalConfiguration

Set-MailboxRegionalConfiguration

Get-MailboxSpellingConfiguration

Set-MailboxSpellingConfiguration

Get-OwaMailboxPolicy

New-OwaMailboxPolicy

Remove-OwaMailboxPolicy

Set-OwaMailboxPolicy

Get-OwaVirtualDirectory

New-OwaVirtualDirectory

Remove-OwaVirtualDirectory

Set-OwaVirtualDirectory

Get-SmimeConfig

Set-SmimeConfig

POP3 and IMAP4 cmdlets

Test-ImapConnectivity

Get-IMAPSettings

Set-ImapSettings

Test-PopConnectivity

Get-POPSettings

Set-PopSettings

Windows PowerShell virtual directory cmdlets

Test-PowerShellConnectivity

Get-PowerShellVirtualDirectory

New-PowerShellVirtualDirectory

Remove-PowerShellVirtualDirectory

Set-PowerShellVirtualDirectory

Test-ActiveSyncConnectivity

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-ActiveSyncConnectivity** cmdlet to perform a full synchronization against a specified mailbox to test the configuration of Microsoft Exchange ActiveSync.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-ActiveSyncConnectivity [-AllowUnsecureAccess <SwitchParameter>] [-
ClientAccessServer <ServerIdParameter>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-LightMode <SwitchParameter>] [-
MailboxCredential <PSCredential>] [-MailboxServer <ServerIdParameter>] [-
MonitoringContext <SwitchParameter>] [-MonitoringInstance <String>] [-
ResetTestAccountCredentials <SwitchParameter>] [-Timeout <UInt32>] [-
TrustAnySSLCertificate <SwitchParameter>] [-URL <String>] [-
UseAutodiscoverForClientAccessServer <SwitchParameter>] [-UserType <LEGACY
| EDU | BPOS>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the Exchange ActiveSync connectivity for the mailbox PaulS on the Client Access server computer CAS01.

```
Test-ActiveSyncConnectivity -ClientAccessServer contoso
\CAS01 -URL "http://contoso.com/mail" -MailboxCredential
"PaulS"
```

EXAMPLE 2

This example tests the Exchange ActiveSync connectivity for the mailbox PaulS using the Autodiscover URL.

```
Test-ActiveSyncConnectivity -
UseAutodiscoverForClientAccessServer $true -URL "http://
contoso.com/mail" -MailboxCredential "pauls@contoso.com"
```

EXAMPLE 3

This example tests the Exchange ActiveSync connectivity for the mailbox PaulS.

```
Test-ActiveSyncConnectivity -AllowUnsecureAccess $true -URL
"http://contoso.com/mail" -MailboxCredential "contoso
\pauls"
```

Detailed Description

The **Test-ActiveSyncConnectivity** cmdlet performs a full synchronization between a mobile device and a specified mailbox to test the functionality of Exchange ActiveSync. If the synchronization fails, a message is displayed in the Exchange Management Shell.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync server settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
AllowUnsecureAccess	Optional	System.Management.Automation.SwitchParameter	The *AllowUnsecureAccess* parameter allows the test to continue over an unsecured channel that doesn't require Secure Sockets Layer (SSL).
ClientAccessServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *ClientAccessServer* parameter specifies the Client Access server computer that the device uses to synchronize.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't

			have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
LightMode	Optional	System.Management.Automation.SwitchParameter	The *LightMode* parameter instructs the command to perform only a subset of the connectivity tests. The tests performed are the **Options** and **FolderSync** commands.
MailboxCredential	Optional	System.Management.Automation.PSCredential	The *MailboxCredential* parameter specifies the mailbox that the device synchronizes with.
MailboxServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *MailboxServer* parameter specifies the Mailbox server computer that contains the associated mailbox.
MonitoringContext	Optional	System.Management.Automation.SwitchPara	The *MonitoringContext* switch includes the

		meter	associated monitoring events and performance counters in the results. You don't need to specify a value with this switch. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
MonitoringInstance	Optional	System.String	The *MonitoringInstance* parameter specifies whether the task is to be run against the mailbox of the user currently running the task. It's only used when this task is run from System Center Operations Manager 2007 or Operations Manager.
ResetTestAccountCredentials	Optional	System.Management.Automation.SwitchParameter	The *ResetTestAccountCredentials* switch resets the

			password for the test account that's used to run this command. The password for the test account is typically reset every seven days. Use this switch to force a password reset any time it's required for security reasons.
Timeout	Optional	System.UInt32	The *Timeout* parameter specifies the amount of time (in seconds) to wait for a response from the command.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	The *TrustAnySSLCertificate* parameter specifies whether the test uses any available SSL certificate to run the test.
URL	Optional	System.String	The *URL* parameter specifies the external URL for Exchange ActiveSync.
UseAutodiscoverForClientAccessServer	Optional	System.Management.Automation.SwitchParameter	The *UseAutodiscoverForClientAccessServer* parameter specifies whether the test should

			use the Autodiscover service to locate the Client Access server.
UserType	Optional	Microsoft.Exchange.Monitoring.DatacenterUserType	The *UserType* parameter specifies whether the user is an on-premise or online user.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Clear-ActiveSyncDevice

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Clear-ActiveSyncDevice** cmdlet to delete all data from a mobile phone.

For information about the parameter sets in the Syntax section below, see Syntax.

⚠**Warning:**
The **Clear-ActiveSyncDevice** cmdlet will be removed in a future version of Exchange. Use the **Clear-MobileDevice** cmdlet instead. If you have any scripts that use the **Clear-ActiveSyncDevice** cmdlet, update them to use the **Clear-MobileDevice** cmdlet.

```
Clear-ActiveSyncDevice -Identity <MobileDeviceIdParameter> [-Cancel
<SwitchParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-NotificationEmailAddresses <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example clears all data from the mobile device with the identity WM_JeffHay.

```
Clear-ActiveSyncDevice -Identity WM_JeffHay
```

EXAMPLE 2

This example clears all data from the mobile device for Tony Smith and sends a confirmation email message to tony@contoso.com.

```
Clear-ActiveSyncDevice -Identity WM_TonySmith -
NotificationEmailAddresses "tony@contoso.com"
```

EXAMPLE 3

This example cancels a previously sent **Clear-ActiveSyncDevice** command request for Tony Smith's mobile device.

```
Clear-ActiveSyncDevice -Identity WM_TonySmith -Cancel $true
```

Detailed Description

The **Clear-ActiveSyncDevice** cmdlet deletes all user data from a mobile device the next time the device receives data from the server running Microsoft Exchange Server 2013. This cmdlet sets the *DeviceWipeStatus* parameter to `$true`. The mobile device acknowledges the cmdlet and records the time stamp in the *DeviceWipeAckTime* parameter.

After you run this cmdlet, you receive a warning that states: "This command will force all the data on the device to be permanently deleted. Do you want to continue?" You must respond to the warning for the cmdlet to run on the mobile phone.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync user settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MobileDeviceIdParameter	The *Identity* parameter specifies the identity of the device that you want to reset.
Cancel	Optional	System.Management.Automation.SwitchParameter	The *Cancel* switch specifies whether the command should be canceled. If you use the *Cancel* switch, a cancellation request is issued for the remote device wipe.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
NotificationEmailAddresses	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *NotificationEmailAddresses* parameter specifies the notification email address for the remote device wipe confirmation.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the

			WhatIf switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ActiveSyncDevice

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ActiveSyncDevice** cmdlet to retrieve the list of devices in your organization that have active Microsoft Exchange ActiveSync partnerships.

Warning:
The **Get-ActiveSyncDevice** cmdlet will be removed in a future version of Exchange. Use the **Get-MobileDevice** cmdlet instead. If you have any scripts that use the **Get-ActiveSyncDevice** cmdlet, update them to use the **Get-MobileDevice** cmdlet.

```
Get-ActiveSyncDevice [-Identity <ActiveSyncDeviceIdParameter>] <COMMON
PARAMETERS>

Get-ActiveSyncDevice -Mailbox <MailboxIdParameter> <COMMON PARAMETERS>
```

```
COMMON PARAMETERS: [-DomainController <Fqdn>] [-Filter <String>] [-
Monitoring <SwitchParameter>] [-Organization <OrganizationIdParameter>] [-
OrganizationalUnit <OrganizationalUnitIdParameter>] [-ResultSize
<Unlimited>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example returns all the Exchange ActiveSync mobile devices that Tony Smith has used that are associated with his mailbox.

```
Get-ActiveSyncDevice -Identity "TonySmith"
```

EXAMPLE 2

This example returns all the Exchange ActiveSync mobile devices that Tony Smith has used that are associated with his mailbox.

```
Get-ActiveSyncDevice -Mailbox "Redmond\TonySmith"
```

Detailed Description

The **Get-ActiveSyncDevice** cmdlet returns identification, configuration, and status information for each device.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile devices user settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Mailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the user whose mobile devices you want to retrieve.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter specifies a set of attributes used to filter the list of returned mobile devices.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ActiveSyncDeviceIdParameter	The *Identity* parameter specifies the device to retrieve. One of the following values is used to identify a mobile device in Active Directory: • `GUID` • `DeviceIdentity` • `Multi-TenantID`
Monitoring	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Monitoring* parameter specifies whether mobile devices created by monitoring accounts are included in the **Get-ActiveSyncDevice** cmdlet output. The default value is *$false*.
Organization	Optional	Microsoft.Exchange.Co	The *Organization*

		nfiguration.Tasks.OrganizationIdParameter	parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies the organizational unit (OU) where the task is run.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute to sort by.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ActiveSyncDevice

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-ActiveSyncDevice** cmdlet to remove the mobile device partnership information

that you specify from a user's mobile device list stored in a mailbox on a computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ActiveSyncDevice -Identity <MobileDeviceIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the mobile device partnership for the device WM_JeffHay.

```
Remove-ActiveSyncDevice -Identity WM_JeffHay
```

EXAMPLE 2

This example removes the mobile device partnership for the device iPhone_TonySmith after displaying the confirm prompt.

```
Remove-ActiveSyncDevice -Identity iPhone_TonySmith -Confirm
$true
```

EXAMPLE 3

This example removes the mobile device partnership for the device Tablet_JeffHay after displaying the confirm prompt.

```
Remove-ActiveSyncDevice -Identity Tablet_JeffHay -Confirm
$true
```

Detailed Description

The **Remove-ActiveSyncDevice** cmdlet is useful for removing mobile devices that no longer synchronize successfully with the server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync user settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.MobileDeviceIdParameter	The *Identity* parameter uniquely identifies the specific device partnership to be removed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ActiveSyncDeviceAccessRule

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ActiveSyncDeviceAccessRule** cmdlet to retrieve an access group of Exchange mobile devices along with their access level.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ActiveSyncDeviceAccessRule [-Identity
<ActiveSyncDeviceAccessRuleIdParameter>] [-DomainController <Fqdn>] [-
Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example lists all the rules currently blocking mobile phones.

```
Get-ActiveSyncDeviceAccessRule | where {$_.AccessLevel -eq
'Block'}
```

EXAMPLE 2

This example lists all device access rules set up on the server.

```
Get-ActiveSyncDeviceAccessRule | Format-List
Characteristic, QueryString, AccessLevel
```

Detailed Description

You can create multiple groups of devices: allowed devices, blocked devices, and quarantined devices with the **New-ActiveSyncDeviceAccessRule** cmdlet. The **Get-ActiveSyncDeviceAccessRule** cmdlet retrieves the settings for any existing group.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ActiveSyncDeviceAccessRuleIdParameter	The *Identity* parameter specifies the unique identifier for the device access rule.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ActiveSyncDeviceAccessRule

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-ActiveSyncDeviceAccessRule** cmdlet to define an access group of Exchange mobile devices along with their access level.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-ActiveSyncDeviceAccessRule -AccessLevel <Allow | Block | Quarantine> -
Characteristic <DeviceType | DeviceModel | DeviceOS | UserAgent> -
```

```
QueryString <String> [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a device access rule that applies to devices defined with the *QueryString* parameter set to iPhone for the device model and blocks those devices.

```
New-ActiveSyncDeviceAccessRule -QueryString iPhone -
Characteristic DeviceModel -AccessLevel Block
```

EXAMPLE 2

This example creates a device access rule that applies to devices defined with the *QueryString* parameter set to `NokiaE521/2.00()MailforExchange` for the user agent and allows those devices.

```
New-ActiveSyncDeviceAccessRule -QueryString
NokiaE521/2.00()MailforExchange -Characteristic UserAgent -
AccessLevel Allow
```

Detailed Description

You can create multiple groups of devices: Allowed devices, blocked devices, and quarantined devices.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
AccessLevel	Required	Microsoft.Exchange.Data.Directory.SystemConfiguration.DeviceAccessLevel	The *AccessLevel* parameter specifies the state of all devices.
Characteristic	Required	Microsoft.Exchange.Data.Directory.SystemConf	The *Characteristic* parameter specifies the

		iguration.DeviceAccessCharacteristic	device characteristic that the query string is to be compared to.
QueryString	Required	System.String	The *QueryString* parameter specifies which devices this rule applies to.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Co	This parameter is

		nfiguration.Tasks.OrganizationIdParameter	reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ActiveSyncDeviceAccessRule

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-ActiveSyncDeviceAccessRule** cmdlet to remove any existing device access rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ActiveSyncDeviceAccessRule -Identity
<ActiveSyncDeviceAccessRuleIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes all device access rules.

```
Get-ActiveSyncDeviceAccessRule | Remove-
ActiveSyncDeviceAccessRule
```

EXAMPLE 2

This example removes all device access rules that have a name that starts with Contoso.

```
Get-ActiveSyncDeviceAccessRule | where {$_.Name -like
'Contoso*'} | Remove-ActiveSyncDeviceAccessRule
```

Detailed Description

If you've created device access rules for groups of devices, you can use the **Remove-ActiveSyncDeviceAccessRule** cmdlet to remove any access rule.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ActiveSyncDeviceAccessRuleIdParameter	The *Identity* parameter specifies the identity of the device access rule.
Confirm	Optional	System.Management.A	The *Confirm* switch

		utomation.SwitchParameter	causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ActiveSyncDeviceAccessRule

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-ActiveSyncDeviceAccessRule** cmdlet to set the level of access for the rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ActiveSyncDeviceAccessRule -Identity
<ActiveSyncDeviceAccessRuleIdParameter> [-AccessLevel <Allow | Block |
Quarantine>] [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the access level granted to phones covered by the rule ContosoPhone (DeviceModel) to Quarantine.

```
Set-ActiveSyncDeviceAccessRule 'ContosoPhone(DeviceModel)'
-AccessLevel Quarantine
```

EXAMPLE 2

This example changes the organization's device access rule so that all devices explicitly allowed to access Microsoft Exchange ActiveSync are quarantined.

```
Get-ActiveSyncDeviceAccessRule | Where {$_.AccessLevel -eq 'Allow'} | Set-ActiveSyncDeviceAccessRule -AccessLevel Quarantine
```

Detailed Description

Your rule can define multiple groups of devices: allowed devices, blocked devices, and quarantined devices.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ActiveSyncDeviceAccessRuleIdParameter	The *Identity* parameter specifies the identity of the device access rule.
AccessLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DeviceAccessLevel	The *AccessLevel* parameter specifies whether the devices are allowed, blocked, or quarantined.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do

			before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ActiveSyncDeviceAutoblockThreshold

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ActiveSyncDeviceAutoblockThreshold** cmdlet to obtain the Autoblock settings for Microsoft Exchange ActiveSync mobile devices..

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ActiveSyncDeviceAutoblockThreshold [-Identity
<ActiveSyncDeviceAutoblockThresholdIdParameter>] [-DomainController
<Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves the threshold settings for the Autoblock threshold rule for UserAgentChanges.

```
Get-ActiveSyncDeviceAutoblockThreshold -Identity
"UserAgentChanges"
```

Detailed Description

Microsoft Exchange and Exchange ActiveSync have the capability to block Exchange ActiveSync mobile devices if these devices display any of a defined list of behaviors that have the capability to cause issues with the server. The **Get-ActiveSyncDeviceAutoblockThreshold** cmdlet returns the

settings for the requested threshold rule.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync Autoblock settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ActiveSyncDeviceAutoblockThresholdIdParameter	The *Identity* parameter specifies the name of the Autoblock threshold rule.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ActiveSyncDeviceAutoblockThreshold

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ActiveSyncDeviceAutoblockThreshold** cmdlet to change settings for autoblocking mobile devices.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ActiveSyncDeviceAutoblockThreshold -Identity
<ActiveSyncDeviceAutoblockThresholdIdParameter> [-AdminEmailInsert
<String>] [-BehaviorTypeIncidenceDuration <EnhancedTimeSpan>] [-
BehaviorTypeIncidenceLimit <Int32>] [-Confirm [<SwitchParameter>]] [-
DeviceBlockDuration <EnhancedTimeSpan>] [-DomainController <Fqdn>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the autoblock threshold rule UserAgentChanges with several settings. It limits the number of accepted UserAgent changes to 2, specifies that the incidence duration is 1440 minutes, and blocks the mobile device for 1440 minutes. Lastly, it inserts a message into the administrative email message sent to the user.

```
Set-ActiveSyncDeviceAutoblockThreshold -Identity
"UserAgentChanges" BehaviorTypeIncidenceLimit 2 -
BehaviorTypeIncidenceDuration 1440 -DeviceBlockDuration
1440 -AdminEmailInsert "<B>Your device has been blocked.</
B> "]
```

EXAMPLE 2

This example sets the autoblock threshold rule RecentCommands with several settings. It limits the number of accepted RecentCommands changes to 5, specifies that the incidence duration is 720 minutes, and blocks the mobile device for 720 minutes. Lastly, it inserts a message into the administrative email message sent to the user.

```
Set-ActiveSyncDeviceAutoblockThreshold -Identity
"RecentCommands" BehaviorTypeIncidenceLimit 5 -
BehaviorTypeIncidenceDuration 720 -DeviceBlockDuration 720
-AdminEmailInsert "<B>Your device has been blocked.</B> "]
```

Detailed Description

Microsoft Exchange and Microsoft Exchange ActiveSync can block Exchange ActiveSync mobile devices if these devices display any of a defined list of behaviors that can potentially cause issues with the server. The **Set-ActiveSyncDeviceAutoblockThreshold** cmdlet can modify an existing autoblock threshold rule and change a variety of settings including the duration of blocking.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync Autoblock settings" entry in the <Clients and mobile devices permissions> topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ActiveSyncDeviceAutoblockThresholdIdParameter	The *Identity* parameter specifies the name of the autoblock threshold rule.
AdminEmailInsert	Optional	System.String	The *AdminEmailInsert* parameter specifies the text to include in the email sent to the user when a mobile device triggers an autoblock threshold rule.
BehaviorTypeIncidenceDuration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *BehaviorTypeIncidenceDuration* parameter specifies the interval (in minutes) within which the BehaviorType must occur to trigger the autoblock rule.

BehaviorTypeIncidenceLimit	Optional	System.Int32	The *BehaviorTypeIncidenceLimit* parameter specifies the number of occurrences of the behavior type needed to trigger blocking.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DeviceBlockDuration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *DeviceBlockDuration* parameter specifies the length of time (in minutes) that the mobile device is blocked.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this

			configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ActiveSyncDeviceClass

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ActiveSyncDeviceClass** cmdlet to retrieve a list of mobile devices that have connected to the servers running Microsoft Exchange Server 2013 in an organization. The cmdlet returns the mobile device type and model information.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ActiveSyncDeviceClass [-Identity <ActiveSyncDeviceClassIdParameter>]
[-DomainController <Fqdn>] [-Filter <String>] [-Organization
<OrganizationIdParameter>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example returns a list of all devices with the **DeviceType** of PocketPC.

```
Get-ActiveSyncDeviceClass -Filter {DeviceType -eq
"PocketPC"}
```

EXAMPLE 2

This example lists all device types within the organization along with a count of the number of devices of each type present.

```
Get-ActiveSyncDeviceClass | group-object -property
DeviceType
```

Detailed Description

You can use this cmdlet to view a list of mobile phones or devices by type. For example, you can return a list of all Android mobile digital devices in the organization or all Windows Phone devices in the organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter specifies the attribute by which to filter the data. The following attributes are supported: • **DeviceType** • **DeviceModel**
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ActiveSyncDeviceClassIdParameter	The *Identity* parameter specifies the group of devices on which to scope the task.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. You can sort by only one attribute at a time. You can sort by the following attributes: • **DeviceType** • **DeviceModel** The results are sorted in

			ascending order.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ActiveSyncDeviceClass

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-ActiveSyncDeviceClass** cmdlet to clean up the list of mobile devices synchronizing with Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ActiveSyncDeviceClass -Identity <ActiveSyncDeviceClassIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example retrieves the list of devices connecting to Exchange 2013, and then removes all inactive mobile devices from the list.

```
Get-ActiveSyncDeviceClass | RemoveActiveSyncDeviceClass
```

Detailed Description

The **Remove-ActiveSyncDeviceClass** cmdlet cleans up the list of devices associated with the

Exchange 2013 organization. Mobile phones and devices that are inactive or have been remote wiped are removed from the list, and the Microsoft Exchange ActiveSync process regenerates the list with the current mobile phones and devices that are connecting to Exchange 2013.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ActiveSyncDeviceClassIdParameter	The *Identity* parameter specifies the group of devices on which to scope the task.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ActiveSyncDeviceStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ActiveSyncDeviceStatistics** cmdlet to retrieve the list of mobile devices configured to synchronize with a specified user's mailbox and return a list of statistics about the mobile devices.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ActiveSyncDeviceStatistics -Mailbox <MailboxIdParameter> <COMMON
PARAMETERS>

Get-ActiveSyncDeviceStatistics -Identity <ActiveSyncDeviceIdParameter>
<COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-GetMailboxLog
<SwitchParameter>] [-NotificationEmailAddresses <MultiValuedProperty>] [-
ShowRecoveryPassword <SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves the statistics for the mobile phone configured to synchronize with the mailbox that belongs to the user Tony Smith.

```
Get-ActiveSyncDeviceStatistics -Identity TonySmith
```

EXAMPLE 2

This example uses the **Get-CASMailbox** cmdlet to determine who in the organization has a Microsoft Exchange ActiveSync mobile device. For each mobile device, the Exchange ActiveSync device statistics are retrieved.

```
$UserList = Get-CASMailbox -Filter
{hasactivesyncdevicepartnership -eq $true -and -not
displayname -like "CAS_{*"} | Get-Mailbox
$UserList | foreach { Get-ActiveSyncDeviceStatistics -
Mailbox $_}
```

EXAMPLE 3

This example retrieves the statistics for the mobile phone configured to synchronize with the mailbox that belongs to the user Tony Smith. It also outputs the Exchange ActiveSync log file and sends it to the System Administrator at admin@contoso.com.

```
Get-ActiveSyncDeviceStatistics -Mailbox TonySmith -
GetMailboxLog $true -NotificationEmailAddresses
"admin@contoso.com"
```

Detailed Description

The **Get-ActiveSyncDeviceStatistics** cmdlet returns a list of statistics about each mobile device. Additionally, it allows you to retrieve logs and send those logs to a recipient for troubleshooting purposes.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync user settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ActiveSyncDeviceIdParameter	The *Identity* parameter specifies the user's device ID. If the *Mailbox* parameter is specified, the *Identity* parameter is disabled.
Mailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the user mailbox for which you want to retrieve the mobile phone statistics.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
GetMailboxLog	Optional	System.Management.Automation.SwitchParameter	The *GetMailboxLog* parameter specifies whether to send the

			mailbox logs via email to the administrator running the task. If the parameter is set to `$true`, the command sends the mailbox logs via email to the administrator running the task. The default value of this parameter is `$false`.
NotificationEmailAddresses	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *NotificationEmailAddresses* parameter specifies an optional list of comma-separated aliases or email addresses where the mailbox logs are sent. If the *GetMailboxLog* parameter is set to `$false`, this parameter is ignored.
ShowRecoveryPassword	Optional	System.Management.Automation.SwitchParameter	The *ShowRecoveryPassword* parameter specifies whether to return the recovery password for the mobile phone as one of the displayed statistics. If this

			parameter is set to `$true`, the command returns the recovery password for the mobile phone as one of the displayed statistics.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-ActiveSyncLog

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Export-ActiveSyncLog** cmdlet to parse the Internet Information Services (IIS) logs and return information about Microsoft Exchange ActiveSync usage, either on the screen or in an output file.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Export-ActiveSyncLog -Filename <String> [-Confirm [<SwitchParameter>]] [-
EndDate <DateTime>] [-Force <SwitchParameter>] [-OutputPath <String>] [-
OutputPrefix <String>] [-StartDate <DateTime>] [-UseGMT <SwitchParameter>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example exports the Exchange ActiveSync log for the date range 06/08/12 to 06/09/12. The times on the report are in Coordinated Universal Time (UTC), and the report is saved in c:\exreports\easreports.

```
Export-ActiveSyncLog -Filename:"c:\Windows\System32\LogFiles\W2SVC1\ex060812.log" -StartDate:"06/08/12" -EndDate:"06/09/12" -UseGMT:$true -OutputPath:"c:\exreports\easreports"
```

EXAMPLE 2

This example exports the Exchange ActiveSync log for the date range 06/20/12 to 07/20/12 by reading all log files in the D:\logs directory. All prompts are suppressed while running the report, and a confirmation message is displayed. The times on the report are in UTC, and the report is saved in c:\exreports\easreports.

```
Dir D:\Logs\*.log | Export-ActiveSyncLog -Filename:"c:\Windows\System32\LogFiles\W2SVC1\ex072012.log" -StartDate:"06/20/12" -EndDate:"07/20/12" -UseGMT:$true -Force $true -Confirm -OutputPath:"c:\exreports\easreports"
```

EXAMPLE 3

This example exports the Exchange ActiveSync log for the date range 02/01/12 to 02/09/12. The times on the report are in UTC, and the report is saved in c:\exreports\easreports.

```
Export-ActiveSyncLog -Filename: "c:\Windows\System32\LogFiles\W2SVC1\ex020912.log" -StartDate:"02/01/12" -EndDate:"02/09/12" -UseGMT:$true -OutputPath:"c:\exreports\easreports"
```

Detailed Description

The **Export-ActiveSyncLog** cmdlet parses the IIS log files and returns information about Exchange ActiveSync usage. This cmdlet can export the output to a file or display it in the Exchange Management Shell.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync server settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Filename	Required	System.String	The *Filename* parameter specifies the name of the input file.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
EndDate	Optional	System.DateTime	The *EndDate* parameter specifies the end date of the date range of the report.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't

			provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
OutputPath	Optional	System.String	The *OutputPath* parameter specifies the name and location for the output file.
OutputPrefix	Optional	System.String	The *OutputPrefix* parameter specifies a prefix to append to the name of the output file.
StartDate	Optional	System.DateTime	The *StartDate* parameter specifies the start date of the date range for the report.
UseGMT	Optional	System.Management.Automation.SwitchParameter	The *UseGMT* switch specifies that Coordinated Universal Time (Greenwich Mean Time) is used for the time in the report output. By default, if this parameter isn't specified, local time is used.
WhatIf	Optional	System.Management.A	The *WhatIf* switch

		utomation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ActiveSyncMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ActiveSyncMailboxPolicy** cmdlet to retrieve the Mobile Device mailbox policy settings for a specific Mobile Device mailbox policy.

For information about the parameter sets in the Syntax section below, see Syntax.

Warning:

The **Get-ActiveSyncMailboxPolicy** cmdlet will be removed in a future version of Microsoft Exchange. Use the **Get-MobileDeviceMailboxPolicy** cmdlet instead. If you have any scripts that use the **Get-ActiveSyncMailboxPolicy** cmdlet, update them to use the **Get-MobileDeviceMailboxPolicy** cmdlet.

```
Get-ActiveSyncMailboxPolicy [-Identity <MailboxPolicyIdParameter>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example returns the policy settings for the Mobile Device mailbox policy named SalesPolicy.

```
Get-ActiveSyncMailboxPolicy -Identity "SalesPolicy"
```

EXAMPLE 2

This example returns the policy settings for the Mobile Device mailbox policy named Default.

```
Get-ActiveSyncMailboxPolicy -Identity "Default"
```

EXAMPLE 3

This example returns the policy settings for the Mobile Device mailbox policy named Management.

```
Get-ActiveSyncMailboxPolicy -Identity "Management"
```

Detailed Description

A Mobile Device mailbox policy is a group of settings that specifies how mobile devices enabled for Microsoft Exchange ActiveSync connect to the computer running Exchange. Exchange supports multiple Mobile Device mailbox policies. The **Get-ActiveSyncMailboxPolicy** cmdlet displays all the policy settings for the specified policy. These settings include password settings, file access settings, and attachment settings.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile Device mailbox policy settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the policy name.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ActiveSyncMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-ActiveSyncMailboxPolicy** cmdlet to create a Microsoft Mobile Device mailbox policy object.

Warning:
The **New-ActiveSyncMailboxPolicy** cmdlet will be removed in a future version of Microsoft Exchange. Use the **New-MobileMailboxPolicy** cmdlet instead. If you have any scripts that use the **New-ActiveSyncMailboxPolicy** cmdlet, update them to use the **New-MobileMailboxPolicy** cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-ActiveSyncMailboxPolicy -Name <String> [-AllowApplePushNotifications
<$true | $false>] [-AllowBluetooth <Disable | HandsfreeOnly | Allow>] [-
AllowBrowser <$true | $false>] [-AllowCamera <$true | $false>] [-
AllowConsumerEmail <$true | $false>] [-AllowDesktopSync <$true | $false>]
[-AllowExternalDeviceManagement <$true | $false>] [-AllowHTMLEmail <$true
| $false>] [-AllowInternetSharing <$true | $false>] [-AllowIrDA <$true |
$false>] [-AllowMobileOTAUpdate <$true | $false>] [-
AllowNonProvisionableDevices <$true | $false>] [-AllowPOPIMAPEmail <$true
| $false>] [-AllowRemoteDesktop <$true | $false>] [-
AllowSimpleDevicePassword <$true | $false>] [-
AllowSMIMEEncryptionAlgorithmNegotiation <BlockNegotiation |
OnlyStrongAlgorithmNegotiation | AllowAnyAlgorithmNegotiation>] [-
AllowSMIMESoftCerts <$true | $false>] [-AllowStorageCard <$true | $false>]
[-AllowTextMessaging <$true | $false>] [-AllowUnsignedApplications <$true
| $false>] [-AllowUnsignedInstallationPackages <$true | $false>] [-
AllowWiFi <$true | $false>] [-AlphanumericDevicePasswordRequired <$true |
$false>] [-ApprovedApplicationList <ApprovedApplicationCollection>] [-
AttachmentsEnabled <$true | $false>] [-Confirm [<SwitchParameter>]] [-
DeviceEncryptionEnabled <$true | $false>] [-DevicePasswordEnabled <$true |
$false>] [-DevicePasswordExpiration <Unlimited>] [-DevicePasswordHistory
<Int32>] [-DevicePolicyRefreshInterval <Unlimited>] [-DomainController
<Fqdn>] [-IrmEnabled <$true | $false>] [-IsDefault <$true | $false>] [-
IsDefaultPolicy <$true | $false>] [-MaxAttachmentSize <Unlimited>] [-
MaxCalendarAgeFilter <All | TwoWeeks | OneMonth | ThreeMonths |
SixMonths>] [-MaxDevicePasswordFailedAttempts <Unlimited>] [-
MaxEmailAgeFilter <All | OneDay | ThreeDays | OneWeek | TwoWeeks |
OneMonth>] [-MaxEmailBodyTruncationSize <Unlimited>] [-
MaxEmailHTMLBodyTruncationSize <Unlimited>] [-MaxInactivityTimeDeviceLock
<Unlimited>] [-MinDevicePasswordComplexCharacters <Int32>] [-
MinDevicePasswordLength <Int32>] [-MobileOTAUpdateMode
<MajorVersionUpdates | MinorVersionUpdates | BetaVersionUpdates>] [-
Organization <OrganizationIdParameter>] [-PasswordRecoveryEnabled <$true |
$false>] [-RequireDeviceEncryption <$true | $false>] [-
RequireEncryptedSMIMEMessages <$true | $false>] [-
RequireEncryptionSMIMEAlgorithm <TripleDES | DES | RC2128bit | RC264bit |
RC240bit>] [-RequireManualSyncWhenRoaming <$true | $false>] [-
RequireSignedSMIMEAlgorithm <SHA1 | MD5>] [-RequireSignedSMIMEMessages
<$true | $false>] [-RequireStorageCardEncryption <$true | $false>] [-
UnapprovedInROMApplicationList <MultiValuedProperty>] [-UNCAccessEnabled
<$true | $false>] [-WhatIf [<SwitchParameter>]] [-WSSAccessEnabled <$true
| $false>]
```

Examples

EXAMPLE 1

This example creates the Mobile Device mailbox policy SalesPolicy that has several preconfigured values.

```
New-ActiveSyncMailboxPolicy -Name:"SalesPolicy" -
DevicePasswordEnabled:$true -
AlphanumericDevicePasswordRequired:$true -
PasswordRecoveryEnabled:$true -IsDefault:$false -
AttachmentsEnabled:$false -AllowStorageCard:$true
```

EXAMPLE 2

This example creates the Mobile Device mailbox policy Management that has several preconfigured values. Users assigned to this policy should have an Enterprise client access license (CAL) to use many of these features.

```
New-ActiveSyncMailboxPolicy -Name:"Management" -
AllowBluetooth:Allow -AllowBrowser:$true -AllowCamera:$true
-AllowPOPIMAPEmail:$false -DevicePasswordEnabled:$true -
AlphanumericDevicePasswordRequired:$true -
PasswordRecoveryEnabled:$true -MaxEmailAgeFilter:OneWeek -
AllowWiFi:$true -AllowStorageCard:$true
```

EXAMPLE 3

This example creates the Mobile Device mailbox policy Contoso that has several preconfigured values. This policy is configured to be the default policy for the organization. The default policy will be assigned to all new users.

```
New-ActiveSyncMailboxPolicy -Name:"Contoso" -
DevicePasswordEnabled:$true -
AlphanumericDevicePasswordRequired:$true -
PasswordRecoveryEnabled:$true -
MinDevicePasswordComplexCharacters:3 -IsDefault:$true -
DevicePasswordHistory:10
```

Detailed Description

The **New-ActiveSyncMailboxPolicy** cmdlet creates a Mobile Device mailbox policy for mailboxes accessed by mobile devices.

Note:
Some Mobile Device mailbox policy settings require the mobile device to have certain built-in features that enforce these security and device management settings. If your organization allows all devices, you must set the *AllowNonProvisionableDevices* parameter to `$true`. This allows devices that can't enforce all policy settings to synchronize with your server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile Device mailbox policy settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name of the policy.
AllowApplePushNotifications	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *AllowApplePushNotifications* parameter specifies whether push notifications are allowed for Apple mobile devices. The default value is `$true`.
AllowBluetooth	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.BluetoothType	The *AllowBluetooth* parameter specifies whether the Bluetooth capabilities of the mobile phone are allowed. The available options are `Disable`, `HandsfreeOnly`, and `Allow`. The default value is `Allow`.
AllowBrowser	Optional	System.Boolean	The *AllowBrowser* parameter specifies whether Microsoft Pocket

			Internet Explorer is allowed on the mobile phone. The default value is `$true`. This parameter doesn't affect third-party browsers.
AllowCamera	Optional	System.Boolean	The *AllowCamera* parameter specifies whether the mobile phone's camera is allowed. The default value is `$true`.
AllowConsumerEmail	Optional	System.Boolean	The *AllowConsumerEmail* parameter specifies whether the mobile phone user can configure a personal email account on the device. The default value is `$true`.
AllowDesktopSync	Optional	System.Boolean	The *AllowDesktopSync* parameter specifies whether the mobile phone can synchronize with a desktop computer through a cable. The default value is `$true`.
AllowExternalDeviceManagement	Optional	System.Boolean	The *AllowExternalDeviceManagement* parameter specifies whether an external device

			management program is allowed to manage the device.
AllowHTMLEmail	Optional	System.Boolean	The *AllowHTMLEmail* parameter specifies whether HTML email is enabled on the device. The default value is `$true`.
AllowInternetSharing	Optional	System.Boolean	The *AllowInternetSharing* parameter specifies whether the mobile phone can be used as a modem to connect a computer to the Internet. The default value is `$true`.
AllowIrDA	Optional	System.Boolean	The *AllowIrDA* parameter specifies whether infrared connections are allowed to the mobile phone. The default value is `$true`.
AllowMobileOTAUpdate	Optional	System.Boolean	The *AllowMobileOTAUpdate* parameter specifies whether certain updates are seen by devices that implemented support for this restricting functionality. Further control can be specified via the *MobileOTAUpdateMode*

			parameter.
AllowNonProvisionableDevices	Optional	System.Boolean	When set to `$true`, the *AllowNonProvisionableDevices* parameter enables all devices to synchronize with the computer running Exchange, regardless of whether the device can enforce all the specific settings established in the Mobile Device mailbox policy. This also includes devices managed by a separate device management system. When set to `$false`, this parameter blocks these devices that aren't provisioned from synchronizing with the server running Exchange. The default value is `$false`.
AllowPOPIMAPEmail	Optional	System.Boolean	The *AllowPOPIMAPEmail* parameter specifies whether the user can configure a POP3 or IMAP4 email account on the device. The default value is `$true`.
AllowRemoteDesktop	Optional	System.Boolean	The *AllowRemoteDesktop*

			parameter specifies whether the mobile phone can initiate a remote desktop connection. The default value is `$true`.
AllowSimpleDevicePassword	Optional	System.Boolean	The *AllowSimpleDevicePassword* parameter specifies whether a simple device password is allowed. A simple device password is a password that has a specific pattern, such as 1111 or 1234. The default value is `$true`.
AllowSMIMEEncryptionAlgorithmNegotiation	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.SMIMEEncryptionAlgorithmNegotiationType	The *AllowSMIMEEncryptionAlgorithmNegotiation* parameter specifies whether the messaging application on the device can negotiate the encryption algorithm in case a recipient's certificate doesn't support the specified encryption algorithm.
AllowSMIMESoftCerts	Optional	System.Boolean	The *AllowSMIMESoftCerts* parameter specifies whether S/MIME software certificates are allowed.

			The default value is `$true`.
AllowStorageCard	Optional	System.Boolean	The *AllowStorageCard* parameter specifies whether the device can access information stored on a storage card. The default value is `$true`.
AllowTextMessaging	Optional	System.Boolean	The *AllowTextMessaging* parameter specifies whether text messaging is allowed from the device. The default value is `$true`.
AllowUnsignedApplications	Optional	System.Boolean	The *AllowUnsignedApplications* parameter specifies whether unsigned applications can be installed on the device. The default value is `$true`.
AllowUnsignedInstallationPackages	Optional	System.Boolean	The *AllowUnsignedInstallationPackages* parameter specifies whether unsigned installation packages can be run on the device. The default value is `$true`.
AllowWiFi	Optional	System.Boolean	The *AllowWiFi* parameter specifies whether wireless Internet access is allowed

			on the device. The default value is `$true`.
AlphanumericDevicePasswordRequired	Optional	System.Boolean	The *AlphanumericDevicePasswordRequired* parameter specifies whether the device password must be alphanumeric. The default value is `$false`.
ApprovedApplicationList	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ApprovedApplicationCollection	The *ApprovedApplicationList* parameter specifies a list of approved applications for the device.
AttachmentsEnabled	Optional	System.Boolean	The *AttachmentsEnabled* parameter specifies whether the user can download attachments. When set to `$false`, the user is blocked from downloading attachments. The default value is `$true`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DeviceEncryptionEnabled	Optional	System.Boolean	The *DeviceEncryptionEnabled* parameter, when set to `$true`, enables device encryption on the mobile phone. The default value is `$false`. Currently, only the storage card can be encrypted on devices running Windows Mobile 6.0 or later. We recommend that you don't use this setting and use the *RequireStorageCardEncryption* parameter instead.
DevicePasswordEnabled	Optional	System.Boolean	When set to `$true`, the *DevicePasswordEnabled* parameter specifies that the user set a password for the device. The default value is `$false`.
DevicePasswordExpiration	Optional	Microsoft.Exchange.Data.Unlimited	The *DevicePasswordExpiration* parameter specifies the length of time, in days, that a password can be used. After this length of time, a new password must be created. The

			format of the parameter is *dd.hh.mm:ss*, for example, 24.00:00 = 24 hours.
DevicePasswordHistory	Optional	System.Int32	The *DevicePasswordHistory* parameter specifies the number of previously used passwords to store. When a user creates a password, the user can't reuse a stored password that was previously used.
DevicePolicyRefreshInterval	Optional	Microsoft.Exchange.Data.Unlimited	The *DevicePolicyRefreshInterval* parameter specifies how often the policy is sent from the server to the mobile phone
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IrmEnabled	Optional	System.Boolean	The *IrmEnabled*

			parameter specifies whether Information Rights Management (IRM) is enabled for the mailbox policy.
IsDefault	Optional	System.Boolean	The *IsDefault* parameter specifies whether this policy is the default Mobile Device mailbox policy. The default value is `$false`. If another policy is currently set as the default, setting this parameter replaces the old default policy with this policy.
IsDefaultPolicy	Optional	System.Boolean	The *IsDefault* parameter specifies whether this policy is the default Mobile Device mailbox policy. The default value is `$false`. If another policy is currently set as the default, setting this parameter replaces the old default policy with this policy.
MaxAttachmentSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxAttachmentSize* parameter specifies the maximum size of attachments that can be downloaded to the mobile

			phone. The default value is `Unlimited`.
MaxCalendarAgeFilter	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.CalendarAgeFilterType	The *MaxCalendarAgeFilter* parameter specifies the maximum range of calendar days that can be synchronized to the device. Possible values are: • `All` • `TwoWeeks` • `OneMonth` • `ThreeMonths` • `SixMonths`
MaxDevicePasswordFailedAttempts	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxDevicePasswordFailedAttempts* parameter specifies the number of attempts a user can make to enter the correct password for the device. You can enter any number from 4 through 16. The default value is 8.
MaxEmailAgeFilter	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.EmailAgeFilterType	The *MaxEmailAgeFilter* parameter specifies the maximum number of days of email items to synchronize to the device. Possible values are: • `All` • `OneDay` • `ThreeDays` • `OneWeek` • `TwoWeeks`

			• `OneMonth` • `ThreeMonths` • `SixMonths`
MaxEmailBodyTruncationSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxEmailBodyTruncationSize* parameter specifies the maximum size at which email messages are truncated when synchronized to the device. The value is specified in kilobytes (KB).
MaxEmailHTMLBodyTruncationSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxEmailHTMLBodyTruncationSize* parameter specifies the maximum size at which HTML-formatted email messages are synchronized to the device. The value is specified in KB.
MaxInactivityTimeDeviceLock	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxInactivityTimeDeviceLock* parameter specifies the length of time that the device can be inactive before the password is required to reactivate the device. You can enter any interval between 30 seconds and 1 hour. The default value is 15 minutes. The format of the

			parameter is *hh.mm:ss*, for example, 15:00 = 15 minutes.
MinDevicePasswordComplexCharacters	Optional	System.Int32	The *MinDevicePasswordComplexCharacters* parameter specifies the minimum number of complex characters required in a device password. A complex character isn't a letter.
MinDevicePasswordLength	Optional	System.Int32	The *MinDevicePasswordLength* parameter specifies the minimum number of characters in the device password. You can enter any number from 1 through 16. The maximum length a password can be is 16 characters. The default value is 4.
MobileOTAUpdateMode	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MobileOTAUpdateModeType	The *MobileOTAUpdateMode* parameter specifies which updates can be seen by the devices that implemented support for this restricting

			functionality. It must be used in conjunction with the *AllowMobileOTAUpdate* parameter to function properly.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter specifies the organization in which you'll perform this action. This parameter doesn't accept wildcard characters, and you must use the exact name of the organization.
PasswordRecoveryEnabled	Optional	System.Boolean	The *PasswordRecoveryEnabled* parameter specifies whether you can store the recovery password for the device on an Exchange server. When set to `$true`, you can store the recovery password for the device on an Exchange server. The default value is `$false`. The recovery password can be viewed from either Microsoft Office Outlook Web App or the Exchange Administration Center.

RequireDeviceEncryption	Optional	System.Boolean	The *RequireDeviceEncryption* parameter specifies whether encryption is required on the device. The default value is `$false`.
RequireEncryptedSMIMEMessages	Optional	System.Boolean	The *RequireEncryptedSMIMEMessages* parameter specifies whether you must encrypt S/MIME messages. The default value is `$false`.
RequireEncryptionSMIMEAlgorithm	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.EncryptionSMIMEAlgorithmType	The *RequireEncryptionSMIMEAlgorithm* parameter specifies what required algorithm must be used when encrypting a message.
RequireManualSyncWhenRoaming	Optional	System.Boolean	The *RequireManualSyncWhenRoaming* parameter specifies whether the device must synchronize manually while roaming. The default value is `$false`.
RequireSignedSMIMEAlgorithm	Optional	Microsoft.Exchange.Data.Directory.SystemCo	The *RequireSignedSMIMEAlgo*

		nfiguration.SignedSMIMEAlgorithmType	*rithm* parameter specifies what required algorithm must be used when signing a message.
RequireSignedSMIMEMessages	Optional	System.Boolean	The *RequireSignedSMIMEMessages* parameter specifies whether the device must send signed S/MIME messages.
RequireStorageCardEncryption	Optional	System.Boolean	The *RequireStorageCardEncryption* parameter specifies whether encryption of a storage card is required. The default value is `$true`.
UnapprovedInROMApplicationList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UnapprovedInROMApplicationList* parameter specifies a list of applications that can't be run in ROM.
UNCAccessEnabled	Optional	System.Boolean	The *UNCAccessEnabled* parameter specifies whether access to Microsoft Windows file shares is enabled. Access to specific shares is configured on the Microsoft Exchange ActiveSync virtual

			directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WSSAccessEnabled	Optional	System.Boolean	The *WSSAccessEnabled* parameter specifies whether access to Microsoft Windows SharePoint Services is enabled. Access to specific shares is configured on the Exchange ActiveSync virtual directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn’t accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn’t return data.

Remove-ActiveSyncMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-ActiveSyncMailboxPolicy** cmdlet to remove a specific Microsoft Mobile Device mailbox policy from a computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ActiveSyncMailboxPolicy -Identity <MailboxPolicyIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Force
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the Mobile Device mailbox policy SalesPolicy.

```
Remove-ActiveSyncMailboxPolicy -Identity "SalesPolicy"
```

EXAMPLE 2

This example removes the Mobile Device mailbox policy Default after confirmation is given.

```
Remove-ActiveSyncMailboxPolicy -Identity "Default" -Confirm
$true
```

EXAMPLE 3

This example removes the Mobile Device mailbox policy Management and bypasses any confirmation prompts.

```
Remove-ActiveSyncMailboxPolicy -Identity "Management" -
Force $true
```

Detailed Description

A Mobile Device mailbox policy is a group of settings that specifies how mobile phones connect to

Exchange. Exchange supports multiple Mobile Device mailbox policies. The **Remove-ActiveSyncMailboxPolicy** cmdlet enables you to remove a specific Mobile Device mailbox policy. If any users are assigned to the policy when you remove it, the **Remove-ActiveSyncMailboxPolicy** cmdlet fails.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile Device mailbox policy settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the policy name.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* parameter specifies that the command should run immediately and bypass confirmation prompts.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ActiveSyncMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-ActiveSyncMailboxPolicy** cmdlet to apply a variety of Mobile Device mailbox policy settings to a server. You can set any of the parameters by using one command.

For information about the parameter sets in the Syntax section below, see Syntax.

⚠Warning:

The **Set-ActiveSyncMailboxPolicy** cmdlet will be removed in a future version of Exchange. Use the **Set-MobileMailboxPolicy** cmdlet instead. If you have any scripts that use the **Set-ActiveSyncMailboxPolicy** cmdlet, update them to use the **Set-MobileMailboxPolicy** cmdlet.

```
Set-ActiveSyncMailboxPolicy -Identity <MailboxPolicyIdParameter> [-
AllowApplePushNotifications <$true | $false>] [-AllowBluetooth <Disable |
HandsfreeOnly | Allow>] [-AllowBrowser <$true | $false>] [-AllowCamera
<$true | $false>] [-AllowConsumerEmail <$true | $false>] [-
AllowDesktopSync <$true | $false>] [-AllowExternalDeviceManagement <$true
| $false>] [-AllowHTMLEmail <$true | $false>] [-AllowInternetSharing
<$true | $false>] [-AllowIrDA <$true | $false>] [-AllowMobileOTAUpdate
<$true | $false>] [-AllowNonProvisionableDevices <$true | $false>] [-
AllowPOPIMAPEmail <$true | $false>] [-AllowRemoteDesktop <$true | $false>]
[-AllowSimpleDevicePassword <$true | $false>] [-
AllowSMIMEEncryptionAlgorithmNegotiation <BlockNegotiation |
OnlyStrongAlgorithmNegotiation | AllowAnyAlgorithmNegotiation>] [-
AllowSMIMESoftCerts <$true | $false>] [-AllowStorageCard <$true | $false>]
[-AllowTextMessaging <$true | $false>] [-AllowUnsignedApplications <$true
| $false>] [-AllowUnsignedInstallationPackages <$true | $false>] [-
AllowWiFi <$true | $false>] [-AlphanumericDevicePasswordRequired <$true |
$false>] [-ApprovedApplicationList <ApprovedApplicationCollection>] [-
AttachmentsEnabled <$true | $false>] [-Confirm [<SwitchParameter>]] [-
DeviceEncryptionEnabled <$true | $false>] [-DevicePasswordEnabled <$true |
$false>] [-DevicePasswordExpiration <Unlimited>] [-DevicePasswordHistory
<Int32>] [-DevicePolicyRefreshInterval <Unlimited>] [-DomainController
<Fqdn>] [-IrmEnabled <$true | $false>] [-IsDefault <$true | $false>] [-
IsDefaultPolicy <$true | $false>] [-MaxAttachmentSize <Unlimited>] [-
MaxCalendarAgeFilter <All | TwoWeeks | OneMonth | ThreeMonths |
SixMonths>] [-MaxDevicePasswordFailedAttempts <Unlimited>] [-
MaxEmailAgeFilter <All | OneDay | ThreeDays | OneWeek | TwoWeeks |
OneMonth>] [-MaxEmailBodyTruncationSize <Unlimited>] [-
MaxEmailHTMLBodyTruncationSize <Unlimited>] [-MaxInactivityTimeDeviceLock
<Unlimited>] [-MinDevicePasswordComplexCharacters <Int32>] [-
MinDevicePasswordLength <Int32>] [-MobileOTAUpdateMode
<MajorVersionUpdates | MinorVersionUpdates | BetaVersionUpdates>] [-Name
<String>] [-PasswordRecoveryEnabled <$true | $false>] [-
RequireDeviceEncryption <$true | $false>] [-RequireEncryptedSMIMEMessages
<$true | $false>] [-RequireEncryptionSMIMEAlgorithm <TripleDES | DES |
RC2128bit | RC264bit | RC240bit>] [-RequireManualSyncWhenRoaming <$true |
$false>] [-RequireSignedSMIMEAlgorithm <SHA1 | MD5>] [-
RequireSignedSMIMEMessages <$true | $false>] [-
RequireStorageCardEncryption <$true | $false>] [-
UnapprovedInROMApplicationList <MultiValuedProperty>] [-UNCAccessEnabled
<$true | $false>] [-WhatIf [<SwitchParameter>]] [-WSSAccessEnabled <$true
| $false>]
```

Examples

EXAMPLE 1

This example sets several policy settings for the Mobile Device mailbox policy SalesPolicy.

```
Set-ActiveSyncMailboxPolicy -Identity:SalesPolicy -
DevicePasswordEnabled:$true -
AlphanumericDevicePasswordRequired:$true -
PasswordRecoveryEnabled:$true -AttachmentsEnabled:$true -
MaxInactivityTimeDeviceLock:15:00 -IsDefault:$false
```

EXAMPLE 2

This example sets several policy settings for the Mobile Device mailbox policy Management.

```
Set-ActiveSyncMailboxPolicy -Identity:Management -
DevicePasswordEnabled:$true -
AlphanumericDevicePasswordRequired:$true -
PasswordRecoveryEnabled:$true -AllowCamera:$true  -
AllowWiFi:$false -AllowStorageCard:$true -
AllowPOPIMAPEmail:$false
```

EXAMPLE 3

This example sets several policy settings for the Mobile Device mailbox policy Default and requires confirmation before applying the settings.

```
Set-ActiveSyncMailboxPolicy -Identity:Default -
DevicePasswordEnabled:$true -
AlphanumericDevicePasswordRequired:$true -
PasswordRecoveryEnabled:$true -MaxEmailAgeFilter:ThreeDays
-AllowWiFi:$false -AllowStorageCard:$true -
AllowPOPIMAPEmail:$false -IsDefault:$true -
AllowTextMessaging:$true -Confirm:$true
```

Detailed Description

With the **Set-ActiveSyncMailboxPolicy** cmdlet, you can set each parameter in a mailbox policy.

Note:
Some Microsoft Mobile Device mailbox policy settings require the mobile device to have

specific built-in features that enforce these security and device management settings. If your organization allows all devices, you must set the *AllowNonProvisionableDevices* parameter to `$true`. This applies to devices that can't enforce all policy settings.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync mailbox policy settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the Mobile Device mailbox policy.
AllowApplePushNotifications	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *AllowApplePushNotifications* parameter specifies whether push notifications are allowed to Apple mobile devices.
AllowBluetooth	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.BluetoothType	The *AllowBluetooth* parameter specifies whether the Bluetooth capabilities are allowed on the mobile phone. The available options are `Disable`, `HandsfreeOnly`, and `Allow`. The default value is `Allow`.
AllowBrowser	Optional	System.Boolean	The *AllowBrowser*

			parameter indicates whether Microsoft Pocket Internet Explorer is allowed on the mobile phone. The default value is `$true`. This parameter doesn't affect third-party browsers.
AllowCamera	Optional	System.Boolean	The *AllowCamera* parameter specifies whether the mobile phone's camera is allowed. The default value is `$true`.
AllowConsumerEmail	Optional	System.Boolean	The *AllowConsumerEmail* parameter specifies whether the mobile phone user can configure a personal email account on the mobile phone. The default value is `$true`. This parameter doesn't control access to emails using third-party mobile phone email programs.
AllowDesktopSync	Optional	System.Boolean	The *AllowDesktopSync* parameter specifies whether the mobile phone can synchronize with a desktop computer through a cable. The

			default value is `$true`.
AllowExternalDeviceManagement	Optional	System.Boolean	The *AllowExternalDeviceManagement* parameter specifies whether an external device management program is allowed to manage the mobile phone.
AllowHTMLEmail	Optional	System.Boolean	The *AllowHTMLEmail* parameter specifies whether HTML email is enabled on the mobile phone. The default value is `$true`. If set to `$false`, all email is converted to plain text before synchronization occurs.
AllowInternetSharing	Optional	System.Boolean	The *AllowInternetSharing* parameter specifies whether the mobile phone can be used as a modem to connect a computer to the Internet. The default value is `$true`.
AllowIrDA	Optional	System.Boolean	The *AllowIrDA* parameter specifies whether infrared connections are allowed to the mobile phone. The default value is `$true`.

AllowMobileOTAUpdate	Optional	System.Boolean	The *AllowMobileOTAUpdate* parameter specifies whether the Exchange ActiveSync mailbox policy can be sent to the mobile phone over a cellular data connection.
AllowNonProvisionableDevices	Optional	System.Boolean	The *AllowNonProvisionableDevices* parameter specifies whether all mobile phones can synchronize with the server running Exchange. When set to `$true`, the *AllowNonProvisionableDevices* parameter enables all mobile phones to synchronize with the Exchange server, regardless of whether the phone can enforce all the specific settings established in the Mobile Device mailbox policy. This also includes mobile phones managed by a separate device management system. When set to `$false`, this parameter blocks mobile

			phones that aren't provisioned from synchronizing with the Exchange server. The default value is `$false`.
AllowPOPIMAPEmail	Optional	System.Boolean	The *AllowPOPIMAPEmail* parameter specifies whether the user can configure a POP3 or IMAP4 email account on the mobile phone. The default value is `$true`. This parameter doesn't control access by third-party email programs.
AllowRemoteDesktop	Optional	System.Boolean	The *AllowRemoteDesktop* parameter specifies whether the mobile phone can initiate a remote desktop connection. The default value is `$true`.
AllowSimpleDevicePassword	Optional	System.Boolean	The *AllowSimpleDevicePassword* parameter specifies whether a simple device password is allowed. A simple device password is a password that has a specific pattern, such as 1111 or 1234. The default value is `$true`.

AllowSMIMEEncryptionAlgorithmNegotiation	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.SMIMEEncryptionAlgorithmNegotiationType	The *AllowSMIMEEncryptionAlgorithmNegotiation* parameter specifies whether the messaging application on the mobile phone can negotiate the encryption algorithm if a recipient's certificate doesn't support the specified encryption algorithm.
AllowSMIMESoftCerts	Optional	System.Boolean	The *AllowSMIMESoftCerts* parameter specifies whether S/MIME software certificates are allowed. The default value is `$true`.
AllowStorageCard	Optional	System.Boolean	The *AllowStorageCard* parameter specifies whether the mobile phone can access information stored on a storage card. The default value is `$true`.
AllowTextMessaging	Optional	System.Boolean	The *AllowTextMessaging* parameter specifies whether text messaging is allowed from the mobile phone. The default value is `$true`.
AllowUnsignedApplications	Optional	System.Boolean	The *AllowUnsignedApplication*

			s parameter specifies whether unsigned applications can be installed on the mobile phone. The default value is `$true`.
AllowUnsignedInstallationPackages	Optional	System.Boolean	The *AllowUnsignedInstallationPackages* parameter specifies whether unsigned installation packages can be executed on the mobile phone. The default value is `$true`.
AllowWiFi	Optional	System.Boolean	The *AllowWiFi* parameter specifies whether wireless Internet access is allowed on the mobile phone. The default value is `$true`.
AlphanumericDevicePasswordRequired	Optional	System.Boolean	The *AlphanumericDevicePasswordRequired* parameter specifies whether the password for the mobile phone must be alphanumeric. The default value is `$false`.
ApprovedApplicationList	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ApprovedApplicationCollection	The *ApprovedApplicationList* parameter specifies a list of approved applications

			for the mobile phone.
AttachmentsEnabled	Optional	System.Boolean	The *AttachmentsEnabled* parameter specifies whether attachments can be downloaded. When set to `$false`, the *AttachmentsEnabled* parameter blocks the user from downloading attachments. The default value is `$true`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DeviceEncryptionEnabled	Optional	System.Boolean	The *DeviceEncryptionEnabled* parameter specifies whether encryption is enabled. The *DeviceEncryptionEnabled* parameter, when set to `$true`, enables device encryption on the mobile phone. The default value

			is `$false`.
DevicePasswordEnabled	Optional	System.Boolean	The *DevicePasswordEnabled* parameter specifies whether a password is required. When set to `$true`, the *DevicePasswordEnabled* parameter requires that the user set a password for the mobile phone. The default value is `$false`.
DevicePasswordExpiration	Optional	Microsoft.Exchange.Data.Unlimited	The *DevicePasswordExpiration* parameter specifies the length of time, in days, that a password can be used. After this length of time, a new password must be created. The format of the parameter is *dd.hh.mm*:*ss*, for example, 24.00:00 = 24 hours.
DevicePasswordHistory	Optional	System.Int32	The *DevicePasswordHistory* parameter specifies the number of previously used passwords to store. When a user creates a password, the user can't

			reuse a stored password that was previously used.
DevicePolicyRefreshInterval	Optional	Microsoft.Exchange.Data.Unlimited	The *DevicePolicyRefreshInterval* parameter specifies how often the policy is sent from the server to the mobile phone.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IrmEnabled	Optional	System.Boolean	The *IrmEnabled* parameter specifies whether Information Rights Management (IRM) is enabled for the mailbox policy.
IsDefault	Optional	System.Boolean	The *IsDefault* parameter specifies whether this policy is the default Mobile Device mailbox policy. The default value is `$false`. If another policy

			is currently set as the default, setting this parameter replaces the old default policy with this policy.
IsDefaultPolicy	Optional	System.Boolean	The *IsDefault* parameter specifies whether this policy is the default Mobile Device mailbox policy. The default value is `$false`. If another policy is currently set as the default, setting this parameter replaces the old default policy with this policy.
MaxAttachmentSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxAttachmentSize* parameter specifies the maximum size of attachments that can be downloaded to the mobile phone. The default value is `Unlimited`.
MaxCalendarAgeFilter	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.CalendarAgeFilterType	The *MaxCalendarAgeFilter* parameter specifies the maximum range of calendar days that can be synchronized to the device. The value is specified by entering one of the following values:

			• `All` • `OneDay` • `ThreeDays` • `OneWeek` • `TwoWeeks` • `OneMonth`
MaxDevicePasswordFailedAttempts	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxDevicePasswordFailedAttempts* parameter specifies the number of attempts a user can make to enter the correct password for the mobile phone. You can enter any number from 4 through 16. The default value is `8`.
MaxEmailAgeFilter	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.EmailAgeFilterType	The *MaxEmailAgeFilter* parameter specifies the maximum number of days of email items to synchronize to the mobile phone. The value is specified by entering one of the following values. • `All` • `OneDay` • `ThreeDays` • `OneWeek` • `TwoWeeks` • `OneMonth`
MaxEmailBodyTruncationSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxEmailBodyTruncationSize* parameter specifies the maximum size at which email messages are truncated when synchronized to the

			mobile phone. The value is specified in kilobytes (KB).
MaxEmailHTMLBodyTruncationSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxEmailHTMLBodyTruncationSize* parameter specifies the maximum size at which HTML-formatted email messages are synchronized to the mobile phone. The value is specified in KB.
MaxInactivityTimeDeviceLock	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxInactivityTimeDeviceLock* parameter specifies the length of time that the mobile phone can be inactive before the password is required to reactivate it. You can enter any interval between 30 seconds and 1 hour. The default value is 15 minutes. The format of the parameter is *hh.mm:ss*, for example, 15:00 = 15 minutes.
MinDevicePasswordComplexCharacters	Optional	System.Int32	The *MinDevicePasswordComplexCharacters* parameter specifies the minimum

			number of complex characters required in a mobile phone password. A complex character isn't a letter.
MinDevicePasswordLength	Optional	System.Int32	The *MinDevicePasswordLength* parameter specifies the minimum number of characters in the device password. You can enter any number from 1 through 16. The maximum length a password can be is 16 characters. The default value is 4.
MobileOTAUpdateMode	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MobileOTAUpdateModeType	This parameter is reserved for internal Microsoft use.
Name	Optional	System.String	The *Name* parameter specifies the friendly name of the Mobile Device mailbox policy.
PasswordRecoveryEnabled	Optional	System.Boolean	The *PasswordRecoveryEnabled* parameter specifies whether the recovery password for the mobile phone is stored on an

			Exchange server. When set to `$true`, the *PasswordRecoveryEnabled* parameter enables you to store the recovery password for the mobile phone on an Exchange server. The default value is `$false`. The recovery password can be viewed from either Microsoft Office Outlook Web App or the Exchange Administration Center.
RequireDeviceEncryption	Optional	System.Boolean	The *RequireDeviceEncryption* parameter specifies whether encryption is required on the device. The default value is `$false`.
RequireEncryptedSMIMEMessages	Optional	System.Boolean	The *RequireEncryptedSMIMEMessages* parameter specifies whether you must encrypt S/MIME messages. The default value is `$false`.
RequireEncryptionSMIMEAlgorithm	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.Encryption	The *RequireEncryptionSMIMEAlgorithm* parameter

		SMIMEAlgorithmType	specifies what required algorithm must be used when encrypting a message.
RequireManualSyncWhenRoaming	Optional	System.Boolean	The *RequireManualSyncWhenRoaming* parameter specifies whether the mobile phone must synchronize manually while roaming. The default value is `$false`.
RequireSignedSMIMEAlgorithm	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.SignedSMIMEAlgorithmType	The *RequireSignedSMIMEAlgorithm* parameter specifies what required algorithm must be used when signing a message.
RequireSignedSMIMEMessages	Optional	System.Boolean	The *RequireSignedSMIMEMessages* parameter specifies whether the mobile phone must send signed S/MIME messages.
RequireStorageCardEncryption	Optional	System.Boolean	The *RequireStorageCardEncryption* parameter specifies whether storage card encryption is enabled for the mailbox policy. Setting this parameter to `$true`

			also sets the *DeviceEncryptionEnabled* parameter to `$true`.
UnapprovedInROMApplicationList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UnapprovedInROMApplicationList* parameter contains a list of applications that can't be run in ROM.
UNCAccessEnabled	Optional	System.Boolean	The *UNCAccessEnabled* parameter specifies whether access to Microsoft Windows file shares is enabled. Access to specific shares is configured on the Exchange ActiveSync virtual directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WSSAccessEnabled	Optional	System.Boolean	The *WSSAccessEnabled*

			parameter specifies whether access to Microsoft Windows SharePoint Services is enabled. Access to specific shares is configured on the Exchange ActiveSync virtual directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ActiveSyncOrganizationSettings

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ActiveSyncOrganizationSettings** cmdlet to retrieve the Microsoft Exchange ActiveSync settings for your Microsoft Exchange Server 2013 organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ActiveSyncOrganizationSettings [-Identity
<ActiveSyncOrganizationSettingsIdParameter>] [-DomainController <Fqdn>] [-
Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves the Exchange ActiveSync organization settings for the domain Contoso.com.

```
Get-ActiveSyncOrganizationSettings -Identity "Contoso.com"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ActiveSyncOrganizationSettingsIdParameter	The *Identity* parameter specifies the unique identifier for the Exchange organization.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ActiveSyncOrganizationSettings

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-ActiveSyncOrganizationSettings** cmdlet to set the Microsoft Exchange ActiveSync settings for the organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ActiveSyncOrganizationSettings [-Identity
<ActiveSyncOrganizationSettingsIdParameter>] [-AdminMailRecipients
<MultiValuedProperty>] [-AllowAccessForUnSupportedPlatform <$true |
$false>] [-Confirm [<SwitchParameter>]] [-DefaultAccessLevel <Allow |
Block | Quarantine>] [-DomainController <Fqdn>] [-
OtaNotificationMailInsert <String>] [-UserMailInsert <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the default access level to quarantine and sets two administrative email addresses.

```
Set-ActiveSyncOrganizationSettings -DefaultAccessLevel
Quarantine -AdminMailRecipients
will@contoso.com,roger@contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
AdminMailRecipients	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AdminMailRecipients* parameter specifies the email addresses of the administrators for reporting purposes. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...`. To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...;`

			`Remove="<value1>","<value2>"...}`.
AllowAccessForUnSupportedPlatform	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DefaultAccessLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DeviceAccessLevel	The *DefaultAccessLevel* parameter specifies the access level for new devices. Valid values are `Allow`, `Block` or `Quarantine`. The default value is `Allow`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ActiveSyncOrganizationSettingsIdParameter	The *Identity* parameter specifies the ActiveSync organization settings object. The default name of this object is `Mobile Mailbox Settings`.
OtaNotificationMailInsert	Optional	System.String	The *OtaNotificationMailInsert* parameter specifies a string of information to be stored and included within an email message to users with Windows Mobile 6.1 devices in the event that the devices need to update their Microsoft Outlook Mobile software to use the new Exchange ActiveSync features in Microsoft Exchange Server 2013.
UserMailInsert	Optional	System.String	The *UserMailInsert* parameter specifies an informational footer that's added to the

			email message sent to users when their mobile device isn't synchronized because the device is quarantined.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ActiveSyncVirtualDirectory

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ActiveSyncVirtualDirectory** cmdlet to retrieve the Microsoft Exchange ActiveSync settings configured on the Exchange ActiveSync website.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ActiveSyncVirtualDirectory -Server <ServerIdParameter> <COMMON
PARAMETERS>


Get-ActiveSyncVirtualDirectory [-Identity <VirtualDirectoryIdParameter>]
<COMMON PARAMETERS>


COMMON PARAMETERS: [-ADPropertiesOnly <SwitchParameter>] [-
DomainController <Fqdn>] [-ShowBackEndVirtualDirectories
<SwitchParameter>] [-ShowMailboxVirtualDirectories <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns the settings for the Exchange ActiveSync virtual directory on the server CAS-01.

```
Get-ActiveSyncVirtualDirectory -Server "CAS-01"
```

EXAMPLE 2

This example returns the settings for a specific Exchange ActiveSync virtual directory on the server CAS-01.

```
Get-ActiveSyncVirtualDirectory -Server "CAS-01" -Identity
"Microsoft-Server-ActiveSync"
```

EXAMPLE 3

This example returns the settings for the Exchange ActiveSync virtual directory on the server CAS-01, for the domain controller DOM-01.

```
Get-ActiveSyncVirtualDirectory -Server "CAS-01" -
DomainController "DOM-01"
```

Detailed Description

Microsoft Exchange Server 2013 contains a default virtual directory that Exchange ActiveSync

mobile devices use to synchronize with the server. You can create multiple virtual directories and assign different devices to different directories. The **Get-ActiveSyncVirtualDirectory** cmdlet retrieves a variety of settings for the virtual directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter uniquely identifies a physical server. When this parameter is set, the command returns information configured on all virtual directories available on that server.
ADPropertiesOnly	Optional	System.Management.Automation.SwitchParameter	The *ADPropertiesOnly* switch specifies whether to return only the properties about the virtual directory stored in Active Directory. The properties stored in the Internet Information Services (IIS) metabase aren't returned.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter uniquely identifies an Exchange ActiveSync virtual directory on a physical server. If this value isn't set, this command returns information configured on all Exchange ActiveSync websites available across all Client Access servers in the Exchange organization.
ShowBackEndVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowBackEndVirtualDirectories* parameter controls whether the virtual directories on the Mailbox server are shown in the Exchange Administration Center.
ShowMailboxVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowMailboxVirtualDirectories* parameter controls whether the virtual directories on the Mailbox server are

			shown in the Exchange Administration Center.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ActiveSyncVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-ActiveSyncVirtualDirectory** cmdlet to create Microsoft Exchange ActiveSync virtual directories on your specified websites.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-ActiveSyncVirtualDirectory [-ApplicationRoot <String>] [-AppPoolId
<String>] [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
ExtendedProtectionFlags <MultiValuedProperty>] [-ExtendedProtectionSPNList
<MultiValuedProperty>] [-ExtendedProtectionTokenChecking <None | Allow |
Require>] [-ExternalAuthenticationMethods <MultiValuedProperty>] [-
ExternalUrl <Uri>] [-InstallProxySubDirectory <$true | $false>] [-
InternalAuthenticationMethods <MultiValuedProperty>] [-InternalUrl <Uri>]
[-Path <String>] [-Role <ClientAccess | Mailbox>] [-Server
<ServerIdParameter>] [-WebSiteName <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates an Exchange ActiveSync virtual directory and specifies the external URL used to connect to the virtual directory.

```
New-ActiveSyncVirtualDirectory -WebSiteName "Default Web
```

```
Site" -ExternalUrl http://www.contoso.com/Microsoft-Server-
ActiveSync
```

EXAMPLE 2

This example creates an Exchange ActiveSync virtual directory and specifies the external and internal URLs used to connect to the virtual directory.

```
New-ActiveSyncVirtualDirectory -WebSiteName "Default Web
Site" -ExternalUrl http://www.contoso.com/mail -InternalUrl
http://contoso/mail
```

EXAMPLE 3

This example creates an Exchange ActiveSync virtual directory for the company Fourth Coffee.

```
New-ActiveSyncVirtualDirectory -WebSiteName "Default Web
Site" -ExternalUrl "http://www.fourthcoffee.com/Microsoft-
Server-ActiveSync"
```

Detailed Description

The **New-ActiveSyncVirtualDirectory** cmdlet creates an Exchange ActiveSync virtual directory on the specified server under the specified website. The Exchange ActiveSync virtual directory created is named Microsoft-Server-ActiveSync. Only one Exchange ActiveSync virtual directory can exist in each Exchange ActiveSync website.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
ApplicationRoot	Optional	System.String	The *ApplicationRoot* parameter specifies the metabase path of the virtual directory. By default, this path is the same as the website in which the virtual

			directory is created.
AppPoolId	Optional	System.String	The *AppPoolId* parameter specifies the pool of programs that can be used with the virtual directory.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a

			proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was

			established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an

			FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN` value. You specify the domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionTokenChecking* parameter defines how you want to use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this

			virtual directory. This is the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the client and server will work, but may not be as secure as a connection using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for

			Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalAuthenticationMethods	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalAuthenticationMethods* parameter specifies the authentication methods supported by the server that contains the virtual directory when access is requested from outside the network's firewall. If this

			parameter isn't set, all authentication methods can be used.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the URL used to connect to the virtual directory from outside the network's firewall.
InstallProxySubDirectory	Optional	System.Boolean	The *InstallProxySubDirectory* parameter controls the creation of the new Exchange ActiveSync subdirectory used for proxying between Microsoft Exchange Server 2013 and previous versions of Exchange.
InternalAuthenticationMethods	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *InternalAuthenticationMethods* parameter specifies the authentication methods supported by the server that contains the virtual directory when access is requested from inside the network's firewall. If this parameter isn't set, all authentication methods can be used.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter

			specifies the URL used to connect to the virtual directory from inside the network's firewall.
Path	Optional	System.String	The *Path* parameter sets the directory that contains the virtual directory's system files.
Role	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.VirtualDirectoryRole	The *Role* parameter specifies the configuration that should be used when the virtual directory is created. The following are the values that can be used with this parameter: 1. **FrontEnd** configure the virtual directory for use on a Client Access server 2. **BackEnd** Configure the virtual directory for use on a Mailbox server.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to create the virtual directory. You can use any value that uniquely identifies the server, for example: 1. Name 2. FQDN 3. Distinguished name

			(DN) 4. Exchange Legacy DN If you don't use the *Server* parameter, the virtual directory will be created on the server where the Remote PowerShell session is established This will always be a Mailbox server. If you want to create the virtual directory on a Client Access server or another Mailbox server, you must use the *Server* parameter.
WebSiteName	Optional	System.String	The *WebSiteName* parameter specifies the name of the Exchange ActiveSync website in which to create the virtual directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ActiveSyncVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-ActiveSyncVirtualDirectory** cmdlet to delete an existing Microsoft Exchange ActiveSync virtual directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ActiveSyncVirtualDirectory -Identity <VirtualDirectoryIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the default Exchange ActiveSync virtual directory from the server Contoso.

```
Remove-ActiveSyncVirtualDirectory contoso\Microsoft-Server-
ActiveSync
```

EXAMPLE 2

This example removes the default Exchange ActiveSync virtual directory from the server Contoso after confirmation is given.

```
Remove-ActiveSyncVirtualDirectory contoso\Microsoft-Server-
```

```
ActiveSync -Confirm $true
```

EXAMPLE 3

This example removes a custom Exchange ActiveSync virtual directory from the server Contoso.

```
Remove-ActiveSyncVirtualDirectory contoso\EAS
```

Detailed Description

When the Exchange ActiveSync virtual directory is removed from a specified server and website, the virtual directory is also removed from Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter uniquely identifies the Exchange ActiveSync virtual directory to be deleted.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ActiveSyncVirtualDirectory

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ActiveSyncVirtualDirectory** cmdlet to configure the Microsoft Exchange ActiveSync settings on a specified virtual directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ActiveSyncVirtualDirectory -Identity <VirtualDirectoryIdParameter> [-
ActiveSyncServer <String>] [-BadItemReportingEnabled <$true | $false>] [-
BasicAuthEnabled <$true | $false>] [-ClientCertAuth <Ignore | Accepted |
Required>] [-CompressionEnabled <$true | $false>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ExtendedProtectionFlags
<MultiValuedProperty>] [-ExtendedProtectionSPNList <MultiValuedProperty>]
[-ExtendedProtectionTokenChecking <None | Allow | Require>] [-
ExternalAuthenticationMethods <MultiValuedProperty>] [-ExternalUrl <Uri>]
[-InstallIsapiFilter <$true | $false>] [-InternalAuthenticationMethods
<MultiValuedProperty>] [-InternalUrl <Uri>] [-
MobileClientCertificateAuthorityURL <String>] [-
MobileClientCertificateProvisioningEnabled <$true | $false>] [-
MobileClientCertTemplateName <String>] [-Name <String>] [-
RemoteDocumentsActionForUnknownServers <Allow | Block>] [-
RemoteDocumentsAllowedServers <MultiValuedProperty>] [-
RemoteDocumentsBlockedServers <MultiValuedProperty>] [-
RemoteDocumentsInternalDomainSuffixList <MultiValuedProperty>] [-
SendWatsonReport <$true | $false>] [-WhatIf [<SwitchParameter>]] [-
WindowsAuthEnabled <$true | $false>]
```

Examples

EXAMPLE 1

This example disables Basic authentication on the default Exchange ActiveSync virtual directory on the server Contoso.

```
Set-ActiveSyncVirtualDirectory -Identity "contoso
\Microsoft-Server-ActiveSync" -BasicAuthEnabled:$false
```

EXAMPLE 2

This example enables bad item reporting and turns on the option to send Watson reports for errors on the server Contoso.

```
Set-ActiveSyncVirtualDirectory -Identity "contoso
\Microsoft-Server-ActiveSync" -
BadItemReportingEnabled:$true -SendWatsonReport:$true
```

EXAMPLE 3

This example configures the external URL on the default Exchange ActiveSync virtual directory on

the server Contoso.

```
Set-ActiveSyncVirtualDirectory -Identity "contoso
\Microsoft-Server-ActiveSync" -ExternalUrl "http://
contoso.com/mail"
```

Detailed Description

The **Set-ActiveSyncVirtualDirectory** cmdlet configures a variety of settings on the virtual directory used for Exchange ActiveSync including security, authentication, and internal and external URL settings.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter uniquely identifies the Exchange ActiveSync virtual directory to be configured. This value must be *website* \Microsoft-Server-ActiveSync.
ActiveSyncServer	Optional	System.String	The *ActiveSyncServer* parameter specifies the URL of the Client Access server. This value is in the following format: https://*servername*/Microsoft-Server-ActiveSync.
BadItemReportingEna	Optional	System.Boolean	The

bled			*BadItemReportingEnabled* parameter specifies whether items that can't be synchronized should be reported to the user. If set to `$true`, the user receives a notification when an item can't be synchronized to the mobile phone.
BasicAuthEnabled	Optional	System.Boolean	The *BasicAuthEnabled* parameter enables or disables Basic authentication. The default setting is `$true`.
ClientCertAuth	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ClientCertAuthTypes	The *ClientCertAuth* parameter specifies the status of client certificate authentication. By default, client certificate authentication is disabled. The default setting is `Ignore`.
CompressionEnabled	Optional	System.Boolean	The *CompressionEnabled* parameter is a Boolean value that identifies the compression applied to the specified Exchange ActiveSync virtual directory. The default setting is `$true`.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in

			the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a

			secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example,

			ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN value`. You specify the domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionTokenChecking* parameter defines how you want to use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is

			the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the client and server will work, but may not be as secure as a connection using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for

			Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalAuthenticationMethods	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalAuthenticationMethods* parameter specifies the authentication methods supported by the server that contains the virtual directory when access is requested from

			outside the network firewall. If this parameter isn't set, all authentication methods can be used.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the URL used to connect to the virtual directory from outside the network firewall.
InstallIsapiFilter	Optional	System.Boolean	The *InstallIsapiFilter* parameter specifies whether the Internet Server API (ISAPI) filter is installed.
InternalAuthenticationMethods	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *InternalAuthenticationMethods* parameter specifies the authentication methods supported by the server that contains the virtual directory when access is requested from inside the network firewall. If this parameter isn't set, all authentication methods can be used.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter is used for the Client Access server HTTP Proxy feature. If it's set, the Client Access server

			receiving the incoming client connections proxies the requests to this URL. This can be an Internet-facing FQDN URL or an internal host name.
MobileClientCertificateAuthorityURL	Optional	System.String	The *MobileClientCertificateAuthorityURL* parameter specifies the URL for the certification authority (CA) used by the mobile phone.
MobileClientCertificateProvisioningEnabled	Optional	System.Boolean	The *MobileClientCertificateProvisioningEnabled* parameter specifies whether the Autodiscover service returns the Certificate Services server URL in the XML file.
MobileClientCertTemplateName	Optional	System.String	The *MobileClientCertTemplateName* parameter specifies the template name for the client certificate.
Name	Optional	System.String	The *Name* parameter specifies the name of the Exchange ActiveSync virtual directory.
RemoteDocumentsAct	Optional	Microsoft.Exchange.Da	The

ionForUnknownServers		ta.Directory.SystemConfiguration.RemoteDocumentsActions	*RemoteDocumentsActionForUnknownServers* parameter specifies the action that occurs when a Microsoft Windows SharePoint Services or Microsoft Windows file share request comes in via Exchange ActiveSync. When a request arrives, Exchange ActiveSync looks for the requested host name in the Allow and Block lists. If the host name isn't found in either list, the action specified in this parameter, either `Block` or `Allow`, is performed.
RemoteDocumentsAllowedServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RemoteDocumentsAllowedServers* parameter is a multivalued property that lists all the allowed servers for remote document access.
RemoteDocumentsBlockedServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RemoteDocumentsBlockedServers* parameter is a multivalued property that lists all the blocked servers for remote

			document access.
RemoteDocumentsInternalDomainSuffixList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RemoteDocumentsInternalDomainSuffixList* parameter is used in organizations that don't run Windows Internet Name Service (WINS) in their network. In these environments, you can specify one or more FQDNs that Exchange ActiveSync treats as internal when a request for remote file access is received.
SendWatsonReport	Optional	System.Boolean	The *SendWatsonReport* parameter specifies whether a Watson report is sent for errors and events.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.
WindowsAuthEnabled	Optional	System.Boolean	The *WindowsAuthEnabled* parameter specifies whether Integrated Windows authentication is enabled. The default value is `$false`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AuthRedirect

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-07*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-AuthRedirect** cmdlet to view OAuth redirection settings for Microsoft Exchange 2010 Client Access servers in your Microsoft Exchange 2013 organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-AuthRedirect [-Identity <AuthRedirectIdParameter>] [-DomainController
<Fqdn>]
```

Examples

Example 1

This example shows a summary list of all the OAuth redirection objects.

```
Get-AuthRedirect
```

Example 2

This example retrieves details about the OAuth redirection object named `AuthRedirect-Bearer-C0B7AC3F-FE64-4B4B-A907-9226F8027CCE`

```
Get-AuthRedirect AuthRedirect-Bearer-C0B7AC3F-FE64-4B4B-A907-9226F8027CCE | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "OAuth authentication redirection settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.AuthRedirectIdParameter	The *Identity* parameter specifies the existing OAuth redirection object that you want to view. The object name uses the syntax `AuthRedirect-Bearer-<GUID>`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-AuthRedirect

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-07*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-AuthRedirect** cmdlet to create OAuth redirection settings for Microsoft Exchange 2010 Client Access servers in your Microsoft Exchange 2013 organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-AuthRedirect -AuthScheme <Unknown | Bearer> -TargetUrl <String> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example creates an OAuth redirection object with the following settings:

- **AuthScheme** Bearer
- **TargetURL** http://cas.contoso.com

```
New-AuthRedirect -AuthScheme Bearer -TargetURL http://
cas.contoso.com
```

Detailed Description

Exchange 2010 Client Access servers don't support OAuth authentication requests. Use this cmdlet

to redirect OAuth authentication requests to an Exchange 2013 Client Access server. This cmdlet is only useful if your organization has Exchange 2010 Client Access servers.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "OAuth authentication redirection settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
AuthScheme	Required	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthScheme	The *AuthScheme* parameter specifies the authentication scheme that's used by the authentication redirection object. Typically, this value is `Bearer`.
TargetUrl	Required	System.String	The *TargetUrl* parameter specifies the FQDN of the Exchange 2013 Client Access server that will process the Oauth request.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value

			with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-AuthRedirect

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-07*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-AuthRedirect** cmdlet to remove OAuth redirection settings for Microsoft Exchange 2010 Client Access servers in your Microsoft Exchange 2013 organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-AuthRedirect -Identity <AuthRedirectIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example removes the existing OAuth redirection object named `AuthRedirect-Bearer-C0B7AC3F-FE64-4B4B-A907-9226F8027CCE`.

```
Remove-AuthRedirect AuthRedirect-Bearer-C0B7AC3F-FE64-4B4B-A907-9226F8027CCE
```

Example 2

This example is the same as the first example. However, if you have only one authentication redirection object in your organization, you can use **Get-AuthRedirect** to pipe the *Identity* value to the **Remove-AuthRedirect** cmdlet.

```
Get-AuthRedirect | Remove-AuthRedirect
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "OAuth authentication redirection settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AuthRedirectIdParameter	The *Identity* parameter specifies the existing OAuth redirection object that you want to remove. The object name uses the syntax `AuthRedirect-Bearer-<GUID>`. The easiest way to find the name of the OAuth redirection object is to run **Get-AuthRedirect**.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AuthRedirect

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-07*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-AuthRedirect** cmdlet to modify the existing OAuth redirection settings for Microsoft Exchange 2010 Client Access servers in your Microsoft Exchange 2013 organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-AuthRedirect -Identity <AuthRedirectIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-TargetUrl <String>] [-
WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example sets the *TargetURL* value to http://cas01.contoso.com for the existing OAuth redirection object named `AuthRedirect-Bearer-C0B7AC3F-FE64-4B4B-A907-9226F8027CCE`.

```
Set-AuthRedirect AuthRedirect-Bearer-C0B7AC3F-FE64-4B4B-
A907-9226F8027CCE -TargetUrl http://cas01.contoso.com
```

Example 2

This example is the same as the first example. However, if you have only one authentication redirection object in your organization, you can use **Get-AuthRedirect** to pipe the *Identity* value to the **Set-AuthRedirect** cmdlet.

```
Get-AuthRedirect | Set-AuthRedirect -TargetUrl http://
cas01.contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "OAuth authentication redirection settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AuthRedirectIdParameter	The *Identity* parameter specifies the existing OAuth redirection object that you want to modify. The object name uses the syntax `AuthRedirect-Bearer-<GUID>`. The easiest way to find the name of the

			OAuth redirection object is to run **Get-AuthRedirect**.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
TargetUrl	Optional	System.String	The *TargetUrl* parameter specifies the FQDN of the Exchange 2013 Client Access server that will process the Oauth request.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-AutoDiscoverConfig

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Export-AutoDiscoverConfig** cmdlet to create or update a service connection point for an Autodiscover service pointer in a target Exchange forest on a computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Export-AutoDiscoverConfig -TargetForestDomainController <String> [-Confirm
[<SwitchParameter>]] [-DeleteConfig <$true | $false>] [-DomainController
<Fqdn>] [-MultipleExchangeDeployments <$true | $false>] [-
PreferredSourceFqdn <Fqdn>] [-SourceForestCredential <PSCredential>] [-
TargetForestCredential <PSCredential>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a service connection point object to connect to another Active Directory forest so that Outlook 2010 or Outlook 2007 clients can automatically connect to their mailbox without having to set up a profile.

```
Export-AutoDiscoverConfig -TargetForestDomainController
targetForestName
```

EXAMPLE 2

This example specifies that Exchange 2013 is deployed in more than one Active Directory forest while establishing an Autodiscover service connection point to the target domain controller in another Active Directory forest.

```
Export-AutoDiscoverConfig -TargetForestDomainController
targetForestName -MultipleExchangeDeployments $true
```

Detailed Description

The Autodiscover service connection point pointer resides in Active Directory and contains the names of the Exchange Web Services URLs.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Autodiscover service configuration settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
TargetForestDomainController	Required	System.String	The *TargetForestDomainController* parameter specifies the domain controller that you want to export the

			Autodiscover configuration to.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DeleteConfig	Optional	System.Boolean	The *DeleteConfig* parameter causes the command to delete your configuration settings on the service connection point object.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
MultipleExchangeDeployments	Optional	System.Boolean	The *MultipleExchangeDeployments* parameter

			specifies whether multiple Exchange deployments exist. This setting should be set to `$true` only if Exchange 2013 is deployed in more than one Active Directory forest, and the forests are connected. If set to `$true`, the list of authoritative accepted domains for the source forest is written to the Autodiscover service connection point object. Microsoft Outlook 2010 and Office Outlook 2007 clients use this object to select the most appropriate forest to search for the Autodiscover service.
PreferredSourceFqdn	Optional	Microsoft.Exchange.Data.Fqdn	The *PreferredSourceFqdn* parameter specifies the FQDN of the Active Directory domain for the Autodiscover pointer service connection point

			object.
SourceForestCredential	Optional	System.Management.Automation.PSCredential	The *SourceForestCredential* parameter specifies the credentials to use when connecting to the source forest.
TargetForestCredential	Optional	System.Management.Automation.PSCredential	The *TargetForestCredential* parameter specifies the credentials to use to connect to the target forest.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AutodiscoverVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-AutodiscoverVirtualDirectory** cmdlet to retrieve the settings for the Autodiscover virtual directory on a computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-AutodiscoverVirtualDirectory -Server <ServerIdParameter> <COMMON
PARAMETERS>


Get-AutodiscoverVirtualDirectory [-Identity <VirtualDirectoryIdParameter>]
<COMMON PARAMETERS>


COMMON PARAMETERS: [-ADPropertiesOnly <SwitchParameter>] [-
DomainController <Fqdn>] [-ShowBackEndVirtualDirectories
<SwitchParameter>] [-ShowMailboxVirtualDirectories <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns settings for the Autodiscover virtual directory under the default website in IIS on the Client Access server CAS01 for the autodiscover.contoso.com site.

```
Get-AutodiscoverVirtualDirectory -Server CAS01 -Identity
"CAS01\autodiscover(autodiscover.contoso.com)"
```

EXAMPLE 2

This example returns settings for the Autodiscover virtual directory located on the Client Access server CAS01 by querying Active Directory using the domain controller specified.

```
Get-AutodiscoverVirtualDirectory -DomainController Exch1 -
```

```
Server CAS01
```

Detailed Description

You can run the **Get-AutodiscoverVirtualDirectory** cmdlet on a local server or run it remotely if the server name is specified in the *Identity* or *Server* parameters. You can also run this cmdlet without parameters to retrieve the configuration settings from all Autodiscover virtual directories on all Internet Information Services (IIS) websites located on the Client Access servers in the organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Autodiscover virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name or GUID of the Client Access server that hosts the virtual directories that you want to display.
ADPropertiesOnly	Optional	System.Management.Automation.SwitchParameter	The *ADPropertiesOnly* switch specifies whether to return only the properties about the virtual directory stored in Active Directory. The properties stored in the Internet Information Services (IIS) metabase aren't returned.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the virtual directory and website for the Autodiscover virtual directory.
ShowBackendVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowBackendVirtualDirectories* switch specifies whether to list the virtual directories located on the Mailbox servers within the organization.
ShowMailboxVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowMailboxVirtualDirectories* switch specifies whether to list the virtual directories located on the Mailbox servers within the organization.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-AutodiscoverVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-AutodiscoverVirtualDirectory** cmdlet to create an Autodiscover virtual directory on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-AutodiscoverVirtualDirectory [-ApplicationRoot <String>] [-AppPoolId
<String>] [-BasicAuthentication <$true | $false>] [-Confirm
[<SwitchParameter>]] [-DigestAuthentication <$true | $false>] [-
DomainController <Fqdn>] [-ExtendedProtectionFlags <MultiValuedProperty>]
[-ExtendedProtectionSPNList <MultiValuedProperty>] [-
ExtendedProtectionTokenChecking <None | Allow | Require>] [-ExternalUrl
<Uri>] [-InternalUrl <Uri>] [-OAuthAuthentication <$true | $false>] [-Path
<String>] [-Role <ClientAccess | Mailbox>] [-Server <ServerIdParameter>]
[-WebSiteName <String>] [-WhatIf [<SwitchParameter>]] [-
WindowsAuthentication <$true | $false>] [-WSSecurityAuthentication <$true
| $false>]
```

Examples

EXAMPLE 1

This example creates the virtual directory autodiscover under the website autodiscover.contoso.com and requires that the user connect using Integrated Windows authentication or Digest authentication.

```
New-AutodiscoverVirtualDirectory -WebSiteName
"autodiscover.contoso.com" -WindowsAuthentication $true -
DigestAuthentication $true
```

Detailed Description

If your organization has multiple email domains and each requires its own Autodiscover site and corresponding virtual directory, use the **New-AutodiscoverVirtualDirectory** cmdlet to create an Autodiscover virtual directory under a new website.

Note:
When you're creating an Autodiscover virtual directory, we recommend that you enable Secure Sockets Layer (SSL) for the Autodiscover service.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Autodiscover virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
ApplicationRoot	Optional	System.String	The *ApplicationRoot* parameter specifies the metabase path of the virtual directory. By default, this path is the same as the website in which the virtual directory is created.
AppPoolId	Optional	System.String	The *AppPoolId* parameter specifies the pool of programs that can be used with the virtual directory.
BasicAuthentication	Optional	System.Boolean	The *BasicAuthentication* parameter specifies whether Basic authentication is enabled on the Autodiscover virtual directory.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DigestAuthentication	Optional	System.Boolean	The *DigestAuthentication* parameter specifies whether Digest authentication is enabled on the Autodiscover virtual directory.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are:

			• **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless

			certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully

			qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN` `value`. You specify the domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionTokenChecking* parameter defines how you want to use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be

			used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the client and server will work, but may not be as secure as a connection using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you

			must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter

			specifies the URL used to connect to the virtual directory from outside the network firewall.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the URL used to connect to the virtual directory from inside the network firewall.
OAuthAuthentication	Optional	System.Boolean	The *OAuthAuthentication* parameter specifies whether OAuth authentication is enabled.
Path	Optional	System.String	The *Path* parameter specifies the directory that contains the system files for the virtual directory.
Role	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.VirtualDirectoryRole	The *Role* parameter specifies the configuration that should be used when the virtual directory is created. The following are the values that can be used with this parameter: • `Client Access` Configures the virtual directory for use on a Client Access server. • `Mailbox` Configures the virtual directory for use on a Mailbox server.
Server	Optional	Microsoft.Exchange.Co	The *Server* parameter specifies the Exchange

		nfiguration.Tasks.ServerIdParameter	server on which you want to create the virtual directory. You can use any value that uniquely identifies the server, for example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the virtual directory will be created on the server where the Remote PowerShell session is established This will always be a Mailbox server. If you want to create the virtual directory on a Client Access server or another Mailbox server, you must use the *Server* parameter.
WebSiteName	Optional	System.String	The *WebSiteName* parameter specifies the name of the Internet Information Services (IIS) website under which to create the virtual directory.
WhatIf	Optional	System.Management.	The *WhatIf* switch

		Automation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsAuthentication	Optional	System.Boolean	The *WindowsAuthentication* parameter specifies whether Integrated Windows authentication is enabled on the Autodiscover virtual directory.
WSSecurityAuthentication	Optional	System.Boolean	The *WSSecurityAuthentication* parameter enables or disables the Web Services Security (WS-Security) endpoint element in the Autodiscover Web.config file. The value can be set to `$true` or `$false`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-AutodiscoverVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-AutodiscoverVirtualDirectory** cmdlet to remove the Autodiscover virtual directory associated with the Autodiscover service on a computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-AutodiscoverVirtualDirectory -Identity
<VirtualDirectoryIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the Autodiscover virtual directory from the site autodiscover.contoso.com on the Client Access server CAS01.

```
Remove-AutodiscoverVirtualDirectory -Identity "CAS01
\autodiscover(autodiscover.contoso.com)"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Autodiscover virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name or GUID of an Autodiscover virtual directory. The *Identity* parameter is represented as: *ServerName\VirtualDirectoryName (WebsiteName)*.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AutodiscoverVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-07*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-AutodiscoverVirtualDirectory** cmdlet to configure settings on the Autodiscover virtual directory on a server running Microsoft Exchange Server 2013. You can run this cmdlet on the local Client Access server or from another Exchange 2013 server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-AutodiscoverVirtualDirectory -Identity <VirtualDirectoryIdParameter>
[-BasicAuthentication <$true | $false>] [-Confirm [<SwitchParameter>]] [-
DigestAuthentication <$true | $false>] [-DomainController <Fqdn>] [-
ExtendedProtectionFlags <MultiValuedProperty>] [-ExtendedProtectionSPNList
<MultiValuedProperty>] [-ExtendedProtectionTokenChecking <None | Allow |
Require>] [-ExternalUrl <Uri>] [-InternalUrl <Uri>] [-
LiveIdBasicAuthentication <$true | $false>] [-
LiveIdNegotiateAuthentication <$true | $false>] [-OAuthAuthentication
<$true | $false>] [-WhatIf [<SwitchParameter>]] [-WindowsAuthentication
<$true | $false>] [-WSSecurityAuthentication <$true | $false>]
```

Examples

EXAMPLE 1

This example sets the authentication method to Digest authentication for the Autodiscover virtual directory.

```
Set-AutodiscoverVirtualDirectory -Identity
'autodiscover(default Web Site)' -WindowsAuthentication
$false -BasicAuthentication $false -DigestAuthentication
$true
```

EXAMPLE 2

This example sets Integrated Windows authentication for the Autodiscover virtual directory.

```
Set-AutodiscoverVirtualDirectory -Identity 'autodiscover
(default Web Site)' -WindowsAuthentication $true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Autodiscover virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name of the virtual directory and the website. This parameter

			can also be used to specify the Client Access server if it's being run from another Exchange 2013 server.
BasicAuthentication	Optional	System.Boolean	The *BasicAuthentication* parameter specifies Basic authentication on the Autodiscover virtual directory.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DigestAuthentication	Optional	System.Boolean	The *DigestAuthentication* parameter specifies Digest authentication on the Autodiscover virtual directory.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration

			change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the

			fully qualified domain name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal

			Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN value`. You specify the domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedP	The *ExtendedProtectionTokenChecking* parameter

		rotectionTokenCheckingMode	defines how you want to use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for

			Authentication on the client and server will work, but may not be as secure as a connection using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNL*

			ist parameter. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalUrl	Optional	System.Uri	This parameter is reserved for internal Microsoft use.
InternalUrl	Optional	System.Uri	This parameter is reserved for internal Microsoft use.
LiveIdBasicAuthentication	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LiveIdNegotiateAuthentication	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
OAuthAuthentication	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.
WindowsAuthentication	Optional	System.Boolean	The *WindowsAuthentication* parameter specifies Integrated Windows authentication on the Autodiscover virtual directory.
WSSecurityAuthentication	Optional	System.Boolean	The *WSSecurityAuthentication* parameter specifies whether to enable the WS-Security (Web Services Security) endpoint element in the Autodiscover Web.config file. The value can be set to `$true` or `$false`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-CalendarConnectivity

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-25*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-CalendarConnectivity** cmdlet to verify that anonymous calendar sharing is enabled and working properly. The Calendar virtual directory is a subdirectory of the Microsoft Office Outlook Web App virtual directory. When you run this command without any parameters, the command tests against all Outlook Web App calendar virtual directories.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-CalendarConnectivity [-ClientAccessServer <ServerIdParameter>] [-
TestType <Internal | External>] [-VirtualDirectoryName <String>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-LightMode
<SwitchParameter>] [-MailboxServer <ServerIdParameter>] [-
MonitoringContext <SwitchParameter>] [-ResetTestAccountCredentials
<SwitchParameter>] [-Timeout <UInt32>] [-TrustAnySSLCertificate
<SwitchParameter>] [-UserType <LEGACY | EDU | BPOS>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the connectivity for the Client Access server CAS01.

```
Test-CalendarConnectivity -ClientAccessServer CAS01
```

Detailed Description

To test virtual directories on a Client Access server, there must be a test Active Directory account. There must also be a test mailbox in each Active Directory site that hosts mailboxes that can be accessed through the virtual directories being tested. You can create the test account by running the New-TestCasConnectivityUser.ps1 script. The default location for this file is: \\Program Files \Microsoft\Exchange Server\V15\Scripts.

If the test environment wasn't created during the Mailbox server setup, you're prompted to run the script that creates the test mailboxes and test users when you run this command.

If the server hosting the test mailbox isn't available, the command returns an error that might not clearly identify the problem. To avoid this, use the **Test-MapiConnectivity** cmdlet to verify that the server that hosts the test mailbox is running and that the mailbox is available before you run this command.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook

Web App virtual directories" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
ClientAccessServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *ClientAccessServer* parameter specifies the name of the Client Access server to test. If this parameter is included, all Outlook Web App calendar virtual directories on the Client Access server are tested against all Exchange Mailbox servers in the local Active Directory site.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory.
LightMode	Optional	System.Management.Automation.SwitchParameter	The *LightMode* parameter isn't implemented for this diagnostic command. Using this parameter doesn't change the behavior of the command. **Note:** This parameter is implemented for other Exchange diagnostic commands where it's used to run a less intensive version of a command.
MailboxServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *MailboxServer* parameter specifies the name of the Mailbox server to test. If not specified, all Mailbox servers in the local Active Directory site are tested.
MonitoringContext	Optional	System.Management.Automation.SwitchParameter	The *MonitoringContext* switch includes the associated monitoring events and performance counters in the results. You don't need to specify a value with this

			switch. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
ResetTestAccountCredentials	Optional	System.Management.Automation.SwitchParameter	The *ResetTestAccountCredentials* switch resets the password for the test account that's used to run this command. The password for the test account is typically reset every seven days. Use this switch to force a password reset any time it's required for security reasons.
TestType	Optional	Microsoft.Exchange.Monitoring.OwaConnectivityTestType	The *TestType* parameter specifies whether the command tests internal or external URLs. Values are `Internal` and `External`. If you don't specify this parameter, the default is

			`TestType:Internal`.
Timeout	Optional	System.UInt32	The *Timeout* parameter isn't implemented for this diagnostic command. Using this parameter doesn't change the behavior of the command. **Note:** This parameter is implemented for other Exchange diagnostic commands where it's used to run a less intensive version of a command.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	The *TrustAnySSLCertificate* parameter isn't implemented for this diagnostic command. Using this parameter doesn't change the behavior of the command. **Note:** This parameter is implemented for other Exchange diagnostic commands where it's used to run a less intensive version of a command.
UserType	Optional	Microsoft.Exchange.Monitoring.DatacenterUse	This parameter is reserved for internal

		rType	Microsoft use.
VirtualDirectoryName	Optional	System.String	The *VirtualDirectoryName* parameter specifies the name of the virtual directory to test on a particular Client Access server. If this parameter isn't included, all Outlook Web App calendar virtual directories that support calendar sharing are tested.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-CASMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-CASMailbox** cmdlet to view the client access settings that are configured on a mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-CASMailbox [-Identity <MailboxIdParameter>] <COMMON PARAMETERS>

Get-CASMailbox [-Anr <String>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
ActiveSyncDebugLogging <SwitchParameter>] [-Credential <PSCredential>] [-
DomainController <Fqdn>] [-Filter <String>] [-GetImapProtocolLog
<SwitchParameter>] [-GetPopProtocolLog <SwitchParameter>] [-
IgnoreDefaultScope <SwitchParameter>] [-Monitoring <SwitchParameter>] [-
Organization <OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-ProtocolSettings <SwitchParameter>] [-
ReadFromDomainController <SwitchParameter>] [-
RecalculateHasActiveSyncDevicePartnership <SwitchParameter>] [-ResultSize
<Unlimited>] [-SendLogsTo <MultiValuedProperty>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example returns the values of the following client access settings for the user named Jeff Hay.

- *ActiveSyncEnabled*
- *OWAEnabled*
- *PopEnabled*
- *ImapEnabled*
- *MapiEnabled*

```
Get-CASMailbox "Jeff Hay"
```

EXAMPLE 2

This example returns all IMAP4 settings for the user tony@contoso.com.

```
Get-CASMailbox tony@contoso.com | Format-List Imap*
```

EXAMPLE 3

This example returns all Exchange Web Services settings for the user chris@contoso.com.

```
Get-CASMailbox chris@contoso.com | Format-List Ews*
```

Detailed Description

This cmdlet returns a variety of client access settings for one or more mailboxes. These settings include options for Microsoft Outlook Web App, Microsoft Exchange ActiveSync, POP3, and IMAP4.

The **Get-CASMailbox** cmdlet can run only on a Mailbox server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access user settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
ActiveSyncDebugLogging	Optional	System.Management.Automation.SwitchParameter	The *ActiveSyncDebugLogging* switch shows the actual value of the **ActiveSyncDebugLogging** property for the mailbox. If you don't use this switch, the value always appears as `$false`.

			To see this value, you need to use a formatting cmdlet. For example, `Get-CasMailbox laura@contoso.com -ActiveSyncDebugLogging | Format-List`.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password that's used to run this command. Typically, you use this parameter in scripts or when you need to provide different credentials that have the required permissions. This parameter requires

			the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
GetImapProtocolLog	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
GetPopProtocolLog	Optional	System.Management.Automation.SwitchPar	This parameter is reserved for internal Microsoft use.

		ameter	
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox that you want to view. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com

			• `User Principal Name` Example: JPhillips@contoso.com
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and

			Identity parameters together. • You can't use the *Credential* parameter.
Monitoring	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Monitoring* switch includes mailboxes that were created by monitoring accounts in the results. By default, these mailboxes aren't included in the results. You don't have to specify a value with this switch.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter limits the search to the specified organizational unit (OU). You can use any value that uniquely identifies the OU. For example: • Distinguished name (DN) Example: `OU=Users,DC=contoso,DC=com` • Canonical name (CN) Example: `contoso.com/`

			`Users` • GUID Example: `8d3d920d-fe9c-4432-8b94-d84027f5d627`
ProtocolSettings	Optional	System.Management.Automation.SwitchParameter	The *ProtocolSettings* switch returns the server names, TCP ports and encryption methods for the following settings: • *ExternalImapSettings* • *InternalImapSettings* • *ExternalPopSettings* • *InternalPopSettings* • *ExternalSmtpSettings* • *InternalSmtpSettings* To see these values, you need to use a formatting cmdlet. For example, `Get-CasMailbox laura@contoso.com -ProtocolSettings | Format-List`.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* switch specifies that information should be read from a domain controller in the user's domain. If you run the command `Set-AdServerSettings -ViewEntireForest $true`

			to include all objects in the forest and you don't use the *ReadFromDomainController* switch, it's possible that information will be read from a global catalog that has outdated information. When you use the *ReadFromDomainController* switch, multiple reads might be necessary to get the information. You don't have to specify a value with this switch. **Note:** By default, the recipient scope is set to the domain that hosts your Exchange servers.
RecalculateHasActiveSyncDevicePartnership	Optional	System.Management.Automation.SwitchParameter	The *RecalculateHasActiveSyncDevicePartnership* switch recalculates the value of the **HasActiveSyncDevicePartnership** property on the mailbox. The value is automatically updated if it's found to be incorrect. You don't have to specify a value with this switch.

ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
SendLogsTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. You can sort by only one attribute at a time. The results are sorted in ascending order.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-CASMailbox

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-CASMailbox** cmdlet to configure client access settings on a mailbox. For example, you can configure settings for Microsoft Exchange ActiveSync, Microsoft Outlook Web App, POP3, and IMAP4.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-CASMailbox -Identity <MailboxIdParameter> [-ActiveSyncAllowedDeviceIDs
<MultiValuedProperty>] [-ActiveSyncBlockedDeviceIDs <MultiValuedProperty>]
[-ActiveSyncDebugLogging <$true | $false>] [-ActiveSyncEnabled <$true |
$false>] [-ActiveSyncMailboxPolicy <MailboxPolicyIdParameter>] [-Confirm
[<SwitchParameter>]] [-DisplayName <String>] [-DomainController <Fqdn>] [-
ECPEnabled <$true | $false>] [-EmailAddresses <ProxyAddressCollection>] [-
EwsAllowEntourage <$true | $false>] [-EwsAllowList <MultiValuedProperty>]
[-EwsAllowMacOutlook <$true | $false>] [-EwsAllowOutlook <$true | $false>]
[-EwsApplicationAccessPolicy <EnforceAllowList | EnforceBlockList>] [-
EwsBlockList <MultiValuedProperty>] [-EwsEnabled <$true | $false>] [-
IgnoreDefaultScope <SwitchParameter>] [-ImapEnabled <$true | $false>] [-
ImapEnableExactRFC822Size <$true | $false>] [-
ImapForceICalForCalendarRetrievalOption <$true | $false>] [-
ImapMessagesRetrievalMimeFormat <TextOnly | HtmlOnly |
HtmlAndTextAlternative | TextEnrichedOnly | TextEnrichedAndTextAlternative
| BestBodyFormat | Tnef>] [-ImapSuppressReadReceipt <$true | $false>] [-
ImapUseProtocolDefaults <$true | $false>] [-MAPIBlockOutlookNonCachedMode
<$true | $false>] [-MAPIBlockOutlookRpcHttp <$true | $false>] [-
MAPIBlockOutlookVersions <String>] [-MAPIEnabled <$true | $false>] [-Name
<String>] [-OWAEnabled <$true | $false>] [-OWAforDevicesEnabled <$true |
$false>] [-OwaMailboxPolicy <MailboxPolicyIdParameter>] [-PopEnabled
<$true | $false>] [-PopEnableExactRFC822Size <$true | $false>] [-
PopForceICalForCalendarRetrievalOption <$true | $false>] [-
PopMessagesRetrievalMimeFormat <TextOnly | HtmlOnly |
HtmlAndTextAlternative | TextEnrichedOnly | TextEnrichedAndTextAlternative
| BestBodyFormat | Tnef>] [-PopSuppressReadReceipt <$true | $false>] [-
PopUseProtocolDefaults <$true | $false>] [-PrimarySmtpAddress
<SmtpAddress>] [-ResetAutoBlockedDevices <SwitchParameter>] [-
SamAccountName <String>] [-ShowGalAsDefaultView <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables Outlook Web App and POP3 access for the user adam@contoso.com.

```
Set-CASMailbox adam@contoso.com -OWAEnabled $false -
PopEnabled $false
```

EXAMPLE 2

This example enables Exchange ActiveSync debug logging and specifies the Exchange ActiveSync mailbox policy named Management for the user adam@contoso.com.

```
Set-CASMailbox adam@contoso.com -ActiveSyncDebugLogging
$true -ActiveSyncMailboxPolicy Management
```

EXAMPLE 3

This example sets the display name and disables Outlook Anywhere access for the user tony@contoso.com.

```
Set-CASMailbox tony@contoso.com -DisplayName "Tony Smith" -
MAPIBlockOutlookRpcHttp $true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client protocol settings" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox that you want to configure. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account`

			Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
ActiveSyncAllowedDeviceIDs	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ActiveSyncAllowedDeviceIDs* parameter specifies one or more Exchange ActiveSync device IDs that are allowed to synchronize with the mailbox. A device ID is a text string that uniquely identifies the device. Use the **Get-MobileDevice** cmdlet to see the devices that have Exchange ActiveSync partnerships with the mailbox.

			To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. To clear the list of device IDs, use the value `$null` for this parameter.
ActiveSyncBlockedDeviceIDs	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ActiveSyncBlockedDeviceIDs* parameter specifies one or more Exchange ActiveSync device IDs that aren't allowed to synchronize with the mailbox. A device ID is a text string that uniquely identifies the device. Use

			the **Get-MobileDevice** cmdlet to see the devices that have Exchange ActiveSync partnerships with the mailbox. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. To clear the list of device IDs, use the value `$null` for this parameter.
ActiveSyncDebugLogging	Optional	System.Boolean	The *ActiveSyncDebugLogging* parameter enables or disables Exchange

			ActiveSync debug logging for the mailbox. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
ActiveSyncEnabled	Optional	System.Boolean	The *ActiveSyncEnabled* parameter enables or disables Exchange ActiveSync for the mailbox. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. Note that when you set this parameter to `$false`, the other Exchange ActiveSync settings in this cmdlet are ignored.
ActiveSyncMailboxPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *ActiveSyncMailboxPolicy* parameter specifies the Exchange ActiveSync mailbox policy for the mailbox. You can use any value that uniquely identifies the Exchange ActiveSync mailbox policy. For example: • Name • Distinguished name (DN) • GUID

			The name of the default Exchange ActiveSync mailbox policy is `Default`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisplayName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *DisplayName* parameter specifies the display name for the mailbox.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

ECPEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *ECPEnabled* parameter enables or disables access to the Exchange admin center (EAC) for the specified user. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
EmailAddresses	Optional	Microsoft.Exchange.Data.ProxyAddressCollection	This parameter is available only in on-premises Exchange 2013. The *EmailAddresses* parameter specifies all the proxy addresses of the recipient. It includes the primary SMTP address as one of the proxy addresses. Typically, the primary SMTP address and other proxy address are set by email address policies. However, you can use this parameter to configure other proxy addresses for the recipient. For more information, see Email address policies.

			Valid syntax for this parameter is `[<Type>]:<emailaddress1>, [<Type>]:<emailaddress2>...`. The optional *<Type>* value indicates the type of email address. Some examples of valid type values include: • `SMTP` The primary SMTP address • `smtp` Other SMTP email addresses • `X400` X.400 addresses • `X500` X.500 addresses If you don't include a *<Type>* value for an email address, the value `smtp` is assumed. To specify the primary SMTP email address, you can use any of the following methods: • Use the *<Type>* value `SMTP`. • The first email address when you don't use the *<Type>* value `SMTP`, and when you don't use any *<Type>* values, or when there are multiple *<Type>* values of `smtp` • Use the *PrimarySmtpAddress*

			parameter instead. You can't use the *EmailAddresses* parameter and the *PrimarySmtpAddress* parameter in the same command. When you specify one or more proxy address by using the *EmailAddresses* parameter, those values replace any exiting proxy addresses that are configured for the recipient. To add or remove specify proxy addresses without affecting other values, use the following syntax: `@{Add="[<Type>]:<emailaddress1>","[<Type>]:<emailaddress2>"...; Remove="[<Type>]:<emailaddress2>","[<Type>]:<emailaddress2>"...}` **Important:** Exchange doesn't validate custom address types (including X.400 addresses) for proper formatting. You need to ensure that any custom addresses comply with the format requirements for that address type.
EwsAllowEntourage	Optional	System.Boolean	The *EwsAllowEntourage* parameter enables or disables access to the

			mailbox by Microsoft Entourage clients that use Exchange Web Services (for example, Entourage 2008 for Mac, Web Services Edition). Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
EwsAllowList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *EwsAllowList* parameter specifies the Exchange Web Services applications (user agent strings) that are allowed to access the mailbox. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following

			syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. This parameter is meaningful only when the *EwsEnabled* parameter is set to `$true`, and the *EwsApplicationAccessPolicy* parameter is set to `EnforceAllowList`.
EwsAllowMacOutlook	Optional	System.Boolean	The *EwsAllowMacOutlook* parameter enables or disables access to the mailbox by Microsoft Outlook for Mac clients that use Exchange Web Services (for example, Outlook for Mac 2011). Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
EwsAllowOutlook	Optional	System.Boolean	The *EwsAllowOutlook* parameter enables or disables access to the mailbox by Microsoft Outlook clients that use Exchange Web Services. Outlook uses Exchange Web Services for free/

			busy, out-of-office settings, and calendar sharing.
EwsApplicationAccessPolicy	Optional	Microsoft.Exchange.Data.Directory.EwsApplicationAccessPolicy	The *EwsApplicationAccessPolicy* parameter controls access to the mailbox by using Exchange Web Services applications. Valid values for this parameter are: • `EnforceAllowList` Only applications specified in the *EwsAllowList* parameter are allowed to access the mailbox. • `EnforceBlockList` Applications specified in the *EwsBlockList* parameter aren't allowed to access the mailbox, but any other applications can access the mailbox. This parameter doesn't affect access to the mailbox by using Entourage, Outlook for Mac, and Outlook. Access to the mailbox by using these clients is controlled by the *EwsAllowEntourage*, *EwsAllowMacOutlook* and

			EwsAllowOutlook parameters.
EwsBlockList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *EwsBlockList* parameter specifies the Exchange Web Services applications (user agent strings) that aren't allowed to access the mailbox by using Exchange Web Services. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. This parameter is meaningful only when

			the *EwsEnabled* parameter is set to `$true`, and the *EwsApplicationAccessPolicy* parameter is set to `EnforceBlockList`.
EwsEnabled	Optional	System.Boolean	The *EwsEnabled* parameter enables or disables access to the mailbox by using Exchange Web Services clients. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. Note that when you set this parameter to `$false`, the other Exchange Web Services settings in this cmdlet are ignored.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This

			allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
ImapEnabled	Optional	System.Boolean	The *ImapEnabled* parameter enables or disables access to the mailbox by using IMAP4 clients. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. Note that when you set this parameter to `$false`, the other IMAP4

			settings in this cmdlet are ignored.
ImapEnableExactRFC822Size	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *ImapEnableExactRFC822Size* parameter specifies how message sizes are presented to IMAP4 clients that access the mailbox. Valid values for this parameter are: • `$true` Calculate the exact message size. • `$false` Use an estimated message size. The default value is `$false`. We don't recommend changing this value unless you determine that the default setting causes problems for IMAP4 clients. To change the value of this parameter, you also need to set the value of the *ImapUseProtocolDefaults* parameter to `$false`.
ImapForceICalForCale	Optional	System.Boolean	The

ndarRetrievalOption			*ImapForceICalForCalendarRetrievalOption* parameter specifies how meeting requests are presented to IMAP4 clients that access the mailbox. Valid values for this parameter are: • `$true` All meeting requests are in the iCal format. • `$false` All meeting requests appear as Outlook Web App links. The default value is `$false`. To change the value of this parameter, you also need to set the value of the *ImapUseProtocolDefaults* parameter to `$false`.
ImapMessagesRetrievalMimeFormat	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MimeTextFormat	The *ImapMessagesRetrievalMimeFormat* parameter specifies the message format for IMAP4 clients that access the mailbox. You can use an integer or a text value. Valid values are: • 0: `TextOnly`

			• 1: `HtmlOnly` • 2: `HtmlAndTextAlternative` • 3: `TextEnrichedOnly` • 4: `TextEnrichedAndTextAlternative` • 5: `BestBodyFormat` • 6: `Tnef` The default value is `BestBodyFormat`. To change the value of this parameter, you also need to set the value of the *ImapUseProtocolDefaults* parameter to `$false`.
ImapSuppressReadReceipt	Optional	System.Boolean	The *ImapSuppressReadReceipt* parameter controls the behavior of read receipts for IMAP4 clients that access the mailbox. Valid values for this parameter are: • `$true` The user receives a read receipt when the recipient opens the message. • `$false` The user receives two read receipts: one when the message is downloaded and another when the message is opened.

			The default value is `$false`. To change the value of this parameter, you also need to set the value of the *ImapUseProtocolDefaults* parameter to `$false`.
ImapUseProtocolDefaults	Optional	System.Boolean	The *ImapUseProtocolDefaults* parameter specifies whether to use the IMAP4 protocol defaults for the mailbox. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. You need to set the *ImapUseProtocolDefaults* parameter to `$false` when you use any of the following IMAP4 parameters: • *ImapEnableExactRFC822Size* • *ImapForceICalForCalendarRetrievalOption* • *ImapMessagesRetrievalMimeFormat* • *ImapSuppressReadReceipt*
MAPIBlockOutlookNo	Optional	System.Boolean	This parameter is

nCachedMode			available only in on-premises Exchange 2013. The *MAPIBlockOutlookNonCachedMode* parameter controls access to the mailbox by using Microsoft Outlook in online or offline mode. Valid values for this parameter are: • `$true` Only Outlook clients that are configured to use Cached Exchange Mode (offline mode) are allowed to access the mailbox. • `$false` The state of the Cached Exchange Mode setting isn't checked before Outlook clients are allowed to access the mailbox (online mode and offline mode are allowed). The default value is `$false`.
MAPIBlockOutlookRpcHttp	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *MAPIBlockOutlookRpcHttp* parameter enables or disables access to the

			mailbox by using Outlook Anywhere (RPC over HTTP) in Microsoft Outlook. Valid values for this parameter are: • `$true` Only Outlook clients that aren't configured to use Outlook Anywhere (RPC over HTTP) are allowed to access the mailbox. By default, Outlook 2013 is configured to use Outlook Anywhere. • `$false` Outlook clients that are configured to use Outlook Anywhere (RPC over HTTP) are allowed to access the mailbox. The default value is `$false`.
MAPIBlockOutlookVersions	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *MAPIBlockOutlookVersions* parameter blocks access to the mailbox for specific versions of Outlook. For example, if you specify the value `15.0.4569.1503`, only Outlook 2013 Service Pack 1 (SP1) or

			later clients are allowed to access the mailbox. Earlier versions of Outlook are blocked.
MAPIEnabled	Optional	System.Boolean	The *MAPIEnabled* parameter enables or disables access to the mailbox by using MAPI clients (for example, Microsoft Outlook). Valid input for this parameter is `$true` or `$false`. The default value is `$true`. Note that when you set this parameter to `$false`, the other MAPI settings in this cmdlet are ignored.
Name	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *Name* parameter specifies the name of the mailbox.
OWAEnabled	Optional	System.Boolean	The *OWAEnabled* parameter enables or disables access to the mailbox by using Outlook Web App. Valid input for this parameter is `$true` or

			`$false`. The default value is `$true`. Note that when you set this parameter to `$false`, the other Outlook Web App settings in this cmdlet are ignored.
OWAforDevicesEnabled	Optional	System.Boolean	The *OWAforDevicesEnabled* parameter enables or disables access to the mailbox by using OWA for Devices. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
OwaMailboxPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *OwaMailboxPolicy* parameter specifies the Outlook Web App mailbox policy for the mailbox. You can use any value that uniquely identifies the Outlook Web App mailbox policy. For example: • Name • Distinguished name (DN) • GUID The name of the default Outlook Web App mailbox

			policy is `Default`.
PopEnabled	Optional	System.Boolean	The *PopEnabled* parameter enables or disables access to the mailbox by using POP3 clients. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. Note that when you set this parameter to `$false`, the other POP3 settings in this cmdlet are ignored.
PopEnableExactRFC822Size	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *PopEnableExactRFC822Size* parameter specifies how message sizes are presented to POP3 clients that access the mailbox. Valid values for this parameter are: • `$true` Calculate the exact message size. • `$false` Use an estimated message size. The default value is `$false`. We don't recommend

			changing this value unless you determine that the default setting causes problems for POP3 clients. To change the value of this parameter, you also need to set the value of the *PopUseProtocolDefaults* parameter to `$false`.
PopForceICalForCalendarRetrievalOption	Optional	System.Boolean	The *PopForceICalForCalendarRetrievalOption* parameter specifies how meeting requests are presented to POP3 clients that access the mailbox. Valid values for this parameter are: • `$true` All meeting requests are in the iCal format. • `$false` All meeting requests appear as Outlook Web App links. The default value is `$false`. To change the value of this parameter, you also need to set the value of the *PopUseProtocolDefaults* parameter to `$false`.

PopMessagesRetrievalMimeFormat	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MimeTextFormat	The *PopMessagesRetrievalMimeFormat* parameter specifies the message format for POP3 clients that access the mailbox. You can use an integer or a text value. Valid values are: • 0: `TextOnly` • 1: `HtmlOnly` • 2: `HtmlAndTextAlternative` • 3: `TextEnrichedOnly` • 4: `TextEnrichedAndTextAlternative` • 5: `BestBodyFormat` • 6: `Tnef` The default value is `BestBodyFormat`. To change the value of this parameter, you also need to set the value of the *PopUseProtocolDefaults* parameter to `$false`.
PopSuppressReadReceipt	Optional	System.Boolean	The *PopSuppressReadReceipt* parameter controls the behavior of read receipts for POP3 clients that access the mailbox.

			Valid values for this parameter are: • `$true` The user receives a read receipt when the recipient opens the message. • `$false` The user receives two read receipts: one when the message is downloaded and another when the message is opened. The default value is `$false`. To change the value of this parameter, you also need to set the value of the *PopUseProtocolDefaults* parameter to `$false`.
PopUseProtocolDefaults	Optional	System.Boolean	The *PopUseProtocolDefaults* parameter specifies whether to use the POP3 protocol defaults for the mailbox. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. You need to set the *PopUseProtocolDefaults* parameter to `$false` when you use any of

			following parameters: • *PopEnableExactRFC822Size* • *PopForceICalForCalendarRetrievalOption* • *PopMessagesRetrievalMimeFormat* • *PopSuppressReadReceipt*
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in on-premises Exchange 2013. The *PrimarySmtpAddress* parameter specifies the primary SMTP email address for the mailbox. This is the From address that external recipients see when they receive a message from this mailbox. You can't use the *PrimarySmtpAddress* parameter and the *EmailAddresses* parameter in the same command.
ResetAutoBlockedDevices	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ResetAutoBlockedDevices*

			switch resets the status of blocked mobile devices that have exceeded the limits defined by the **Set-ActiveSyncDeviceAutoblockThreshold** cmdlet.
SamAccountName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *SamAccountName* parameter specifies an account identifier that's compatible with older operating systems. The value of this parameter must be less than 20 characters, and can contain letters, numbers, and the following characters: • ! • # • $ • % • ^ • & • - • . • _ • { • } • |

			• ~
ShowGalAsDefaultView	Optional	System.Boolean	The *ShowGalAsDefaultView* parameter shows the global address list (GAL) as the default recipient picker for messages. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ClientAccessArray

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ClientAccessArray** cmdlet to return an object that represents a load-balanced array of Client Access servers.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ClientAccessArray [-Identity <ClientAccessArrayIdParameter>] [-
DomainController <Fqdn>] [-Site <AdSiteIdParameter>]
```

Examples

EXAMPLE 1

This example returns the array of Client Access servers with the unique identifier CASArray01 in the current Active Directory site.

```
Get-ClientAccessArray -Identity "CASArray01"
```

EXAMPLE 2

This example returns the array of Client Access servers with the unique identifier CASArray01 in the Active Directory site eur.contoso.com.

```
Get-ClientAccessArray -Identity "CASArray01" -Site
"eur.contoso.com"
```

EXAMPLE 3

This example returns the array of Client Access servers with the unique identifier Array12032 in the current Active Directory site.

```
Get-ClientAccessArray -Identity "Array12032"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Client Access server array settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ClientAccessArrayIdParameter	The *Identity* parameter specifies a unique identifier for the array of Client Access servers.
Site	Optional	Microsoft.Exchange.Configuration.Tasks.AdSiteIdParameter	The *Site* parameter specifies the Active Directory site that contains the specified array of Client Access servers.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ClientAccessServer

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ClientAccessServer** cmdlet to return information for the Client Access server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ClientAccessServer [-Identity <ClientAccessServerIdParameter>] [-
DomainController <Fqdn>] [-
IncludeAlternateServiceAccountCredentialPassword <SwitchParameter>] [-
IncludeAlternateServiceAccountCredentialStatus <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns information about the Exchange Client Access server with the FQDN of mail.contoso.com.

```
Get-ClientAccessServer -Identity "mail.contoso.com"
```

EXAMPLE 2

This example returns information about all computers running Exchange that have the Client Access server role installed.

```
Get-ClientAccessServer
```

EXAMPLE 3

This example returns information about the Exchange Client Access server with the FQDN of email.fourthcoffee.com.

```
Get-ClientAccessServer -Identity "email.fourthcoffee.com"
```

Detailed Description

The Client Access server is one of two server types that can be installed with Microsoft Exchange Server 2013. The **Get-ClientAccessServer** cmdlet returns information about all Client Access servers in the organization.

The **ExchangeVersion** attribute returned is the minimum version of Exchange that you can use to manage the returned object.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access server settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ClientAccessServerIdParameter	The *Identity* parameter specifies the name of the Exchange Client Access server.
IncludeAlternateServiceAccountCredentialPassword	Optional	System.Management.Automation.SwitchParameter	The *IncludeAlternateServiceAccountCredentialPassword* parameter specifies whether the credential password should be included with the request.
IncludeAlternateServiceAccountCredentialStatus	Optional	System.Management.Automation.SwitchParameter	The *IncludeAlternateServiceAccountCredentialStatus* parameter specifies

			whether the status of the service account credential should be included with the request.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ClientAccessServer

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ClientAccessServer** cmdlet to set properties on specified Client Access server objects.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ClientAccessServer [-Array <ClientAccessArrayIdParameter>] [-
AutoDiscoverServiceInternalUri <Uri>] [-AutoDiscoverSiteScope
<MultiValuedProperty>] <COMMON PARAMETERS>


Set-ClientAccessServer [-AlternateServiceAccountCredential <PSCredential[]
>] [-CleanUpInvalidAlternateServiceAccountCredentials <SwitchParameter>]
[-RemoveAlternateServiceAccountCredentials <SwitchParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: -Identity <ClientAccessServerIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IsOutOfService <$true |
$false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets two properties on the Client Access server CAS-01.

```
Set-ClientAccessServer -Identity "CAS-01" -
AutoDiscoverServiceInternalUri "https://cas01.contoso.com/
autodiscover/autodiscover.xml" -AutoDiscoverSiteScope
"Mail"
```

EXAMPLE 2

This example sets two properties on the Client Access server CASMail.

```
Set-ClientAccessServer -Identity "CASMail" -
AutoDiscoverServiceInternalUri "https://
casmail.contoso.com/autodiscover/autodiscover.xml" -
AutoDiscoverSiteScope "Mail"
```

EXAMPLE 3

This example sets two properties on the Client Access server WebMail.

```
Set-ClientAccessServer -Identity "WebMail" -
AutoDiscoverServiceInternalUri "https://
webmail.contoso.com/autodiscover/autodiscover.xml" -
AutoDiscoverSiteScope "Mail"
```

Detailed Description

You can run the **Set-ClientAccessServer** cmdlet for a single Client Access server or for all Client Access servers in your Exchange organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access server settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.Client	The *Identity* parameter specifies an individual

		AccessServerIdParameter	Client Access server.
AlternateServiceAccountCredential	Optional	System.Management.Automation.PSCredential[]	The *AlternateServiceAccountCredential* parameter specifies a credential (consisting of a user name and password) to distribute to all Client Access servers in an organization.
Array	Optional	Microsoft.Exchange.Configuration.Tasks.ClientAccessArrayIdParameter	This parameter is reserved for internal Microsoft use.
AutoDiscoverServiceInternalUri	Optional	System.Uri	The *AutoDiscoverServiceInternalUri* parameter specifies the internal URL of the Autodiscover service.
AutoDiscoverSiteScope	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AutoDiscoverSiteScope* parameter specifies the site for which the Autodiscover service is authoritative. Clients that connect to the Autodiscover service by using the internal URL must belong to a listed

			site.
CleanUpInvalidAlternateServiceAccountCredentials	Optional	System.Management.Automation.SwitchParameter	The *CleanUpInvalidAlternateServiceAccountCredentials* parameter specifies whether to remove a previously set alternate service account credential that's no longer valid.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IsOutOfService	Optional	System.Boolean	This parameter is reserved for internal

			Microsoft use.
RemoveAlternateServiceAccountCredentials	Optional	System.Management.Automation.SwitchParameter	The *RemoveAlternateServiceAccountCredentials* parameter specifies whether to remove a previously distributed alternate service account credential.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-EcpConnectivity

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-EcpConnectivity** cmdlet to verify that the Exchange Administration Center is running as expected.

Warning:
The Exchange Control Panel (ECP) is the web-based user interface developed for Microsoft Exchange Server 2010. The Exchange Server 2013 Exchange Administration Center cmdlets for virtual directory still use ECP in the name, and the ECP cmdlets can be used to manage Exchange 2010 and Exchange 2013 ECP virtual directories.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-EcpConnectivity [-ClientAccessServer <ServerIdParameter>] [-
RSTEndpoint <String>] [-TestType <Internal | External>] [-
VirtualDirectoryName <String>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-LightMode <SwitchParameter>] [-MailboxServer
<ServerIdParameter>] [-MonitoringContext <SwitchParameter>] [-
ResetTestAccountCredentials <SwitchParameter>] [-Timeout <UInt32>] [-
TrustAnySSLCertificate <SwitchParameter>] [-UserType <LEGACY | EDU |
BPOS>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests user connectivity to the Exchange Administration Center on Server01.

```
Test-EcpConnectivity -ClientAccessServer Server01
```

Detailed Description

The **Test-EcpConnectivity** cmdlet can be used to test Exchange Administration Center connectivity for all ECP virtual directories on a specified Client Access server for all mailboxes on servers running Exchange in the same Active Directory site. The **Test-EcpConnectivity** cmdlet can also be used to test the connectivity for an individual Exchange Administration Center URL.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange

Administration Center connectivity" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
ClientAccessServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *ClientAccessServer* parameter specifies the name of the Client Access server to test. If this parameter is included, all Exchange ECP virtual directories on the Client Access server are tested against all Exchange Mailbox servers in the local Active Directory site. Mailboxes that aren't on Exchange Mailbox servers are tested.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that retrieves data from Active Directory.
LightMode	Optional	System.Management.Automation.SwitchParameter	The *LightMode* parameter isn't implemented for this diagnostic command. Using this parameter doesn't change the behavior of the command. **Note:** This parameter is implemented for other Exchange diagnostic commands where it's used to run a less intensive version of a command.
MailboxServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *MailboxServer* parameter specifies the name of the Mailbox server to test. If not specified, all Mailbox servers in the local Active Directory site are tested.
MonitoringContext	Optional	System.Management.Automation.SwitchParameter	The *MonitoringContext* switch includes the associated monitoring events and performance counters in the results.

			You don't need to specify a value with this switch. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
ResetTestAccountCredentials	Optional	System.Management.Automation.SwitchParameter	The *ResetTestAccountCredentials* parameter resets the password for the test account used to run this command. The password for the test account is usually reset every 7 days. Use this parameter to force a password reset any time that a password reset is required for security reasons.
RSTEndpoint	Optional	System.String	This parameter is reserved for internal Microsoft use.
TestType	Optional	Microsoft.Exchange.Mo	The *TestType* parameter

		nitoring.OwaConnectivityTestType	determines whether the command tests internal or external URLs. Values are `Internal` and `External`.
Timeout	Optional	System.UInt32	The *Timeout* parameter specifies the amount of time, in seconds, to wait for the test operation to finish. The default value for the *Timeout* parameter is 30 seconds. You must specify a time-out value greater than 0 seconds and less than 1 hour (3,600 seconds). We recommend that you always configure this parameter with a value of 5 seconds or more.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	The *TrustAnySSLCertificate* parameter specifies whether Secure Sockets Layer (SSL) certificate validation failures are reported. If the *TrustAnySSLCertificate* parameter is used, SSL certificate validation failures aren't reported.

			This is useful for testing internal URLs because Internet Information Services (IIS) doesn't support assigning multiple certificates for a single virtual directory. If a directory has different URLs for internal and external access and has a certificate, that certificate is usually for the external URL. This parameter lets the task check an internal URL without generating an error when the certificate doesn't match the URL.
UserType	Optional	Microsoft.Exchange.Monitoring.DatacenterUserType	This parameter is reserved for internal Microsoft use.
VirtualDirectoryName	Optional	System.String	The *VirtualDirectoryName* parameter specifies the name of the virtual directory to test on a particular Client Access server. If this parameter isn't included, all Exchange Control Panel

			virtual directories that support Exchange mailboxes are tested.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-EcpVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-EcpVirtualDirectory** cmdlet to retrieve all configuration data for Exchange Control Panel (ECP) virtual directories. The ECP virtual directory manages the Exchange Administration Center.

For information about the parameter sets in the Syntax section below, see Syntax.

Note:
The ECP is the web-based user interface developed for Microsoft Exchange Server 2010. The Exchange Server 2013 Exchange Administration Center cmdlets for virtual directory still use ECP in the name, and the ECP cmdlets can be used to manage Exchange 2010 and Exchange 2013 ECP virtual directories.

```
Get-EcpVirtualDirectory -Server <ServerIdParameter> <COMMON PARAMETERS>

Get-EcpVirtualDirectory [-Identity <VirtualDirectoryIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-ADPropertiesOnly <SwitchParameter>] [-
DomainController <Fqdn>] [-ShowBackEndVirtualDirectories
<SwitchParameter>] [-ShowMailboxVirtualDirectories <SwitchParameter>]
```

Examples

EXAMPLE 1

This example displays the configuration data for the Exchange Control Panel virtual directory on Server01.

```
Get-EcpVirtualDirectory -Server Server01
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Administration Center connectivity" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name or GUID of the server that hosts the ECP virtual

			directories that you want to display. If you don't specify a value for the *Server* parameter, all ECP virtual directories are returned.
ADPropertiesOnly	Optional	System.Management.Automation.SwitchParameter	The *ADPropertiesOnly* switch specifies whether to return only the properties about the virtual directory stored in Active Directory. The properties stored in the Internet Information Services (IIS) metabase aren't returned.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name or GUID of an ECP virtual directory. The *Identity* parameter is represented as:

			ServerName\ECP (*WebsiteName*). If you don't specify a server name, the command returns the ECP virtual directory on the local server.
ShowBackEndVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowBackEndVirtualDirectories* switch specifies whether to return virtual directories on Client Access servers when used in a query. If you don't use this parameter, only virtual directories on Mailbox servers are returned.
ShowMailboxVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowMailboxVirtualDirectories* switch specifies whether to return the ECP virtual directories that are located on servers running the Mailbox server role. This switch should only be used with the direction of Microsoft support.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-EcpVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-EcpVirtualDirectory** cmdlet to create an Exchange Control Panel (ECP) virtual directory. This is a command that Microsoft Exchange Server Setup runs when you install Exchange in your organization. The ECP virtual directory manages the Exchange admin center.

Warning:
The ECP is the web-based user interface developed for Microsoft Exchange Server 2010. The Exchange Server 2013 Exchange Admin center cmdlets for virtual directory still use ECP in the name, and the ECP cmdlets can be used to manage Exchange 2010 and Exchange 2013 ECP virtual directories.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-EcpVirtualDirectory [-AppPoolId <String>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ExtendedProtectionFlags
<MultiValuedProperty>] [-ExtendedProtectionSPNList <MultiValuedProperty>]
[-ExtendedProtectionTokenChecking <None | Allow | Require>] [-ExternalUrl
<Uri>] [-InternalUrl <Uri>] [-Path <String>] [-Role <ClientAccess |
Mailbox>] [-Server <ServerIdParameter>] [-WebSiteName <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the Exchange Control Panel virtual directory on the Exchange server Server01.

```
New-EcpVirtualDirectory -Server SERVER01 -ExternalURL
https://mail.contoso.com/ecp -InternalURL https://
mail.contoso.com/ecp
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Administration Center connectivity" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
AppPoolId	Optional	System.String	The *AppPoolId* parameter sets the Internet Information Services (IIS) application pool in which the Exchange Control Panel virtual directory runs. We recommend that you leave this parameter at its default setting.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Da	The *DomainController*

		ta.Fqdn	parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the

			Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.

ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN value`. You specify the domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/

			mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionTokenChecking* parameter defines how you want to use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for

			Authentication. Connections that don't support Extended Protection for Authentication on the client and server will work, but may not be as secure as a connection using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail.

			If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the host name used to connect to the server running Exchange from outside the firewall. This setting is important when Secure Sockets Layer (SSL) is used. This parameter must be set to allow the Autodiscover service to return the URL for the Exchange Control Panel virtual directory.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the host name used to connect to the server running Exchange from inside the firewall.

			This setting is important when SSL is used. This parameter must be set to allow the Autodiscover service to return the URL for the Exchange Control Panel virtual directory.
Path	Optional	System.String	The *Path* parameter sets the file system path of the Exchange Control Panel virtual directory. This parameter should be used with care and only when you must use a different file system path than the default.
Role	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.VirtualDirectoryRole	The *Role* parameter specifies the configuration that should be used when the virtual directory is created. The following are the values that can be used with this parameter: • `ClientAccess` Configure the virtual directory for use on a Client Access server. • `Mailbox` Configure the virtual directory for use on a Mailbox server.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name or GUID of the server that

			hosts the Exchange Control Panel virtual directories that you want to create. You can't create an Exchange Control Panel virtual directory remotely. You can only create an Exchange Control Panel virtual directory on the local computer.
WebSiteName	Optional	System.String	The *WebSiteName* parameter specifies the name of the IIS website under which the Exchange Control Panel virtual directory is created.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-EcpVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-EcpVirtualDirectory** cmdlet to remove Exchange Control Panel (ECP) virtual directories located in the Internet Information Services (IIS) website on a server running Exchange. The ECP virtual directory manages the Exchange Administration Center.

⚠**Warning:**
The ECP is the web-based user interface developed for Microsoft Exchange Server 2010. The Exchange Server 2013 Exchange Administration Center cmdlets for virtual directory still use ECP in the name, and the ECP cmdlets can be used to manage Exchange 2010 and Exchange 2013 ECP virtual directories.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-EcpVirtualDirectory -Identity <VirtualDirectoryIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the ECP virtual directory ecp located on the default IIS website on the Exchange server Server01.

```
Remove-EcpVirtualDirectory -Identity "Server01\ecp (default
Web site)"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Administration Center connectivity" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name or GUID of an ECP virtual directory. The *Identity* parameter is represented as: *ServerName*\ECP (*WebsiteName*). Remote removal of an ECP virtual directory isn't supported. You must run this command from the local computer.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-EcpVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-07*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-EcpVirtualDirectory** cmdlet to modify the properties of an Exchange Control Panel (ECP) virtual directory. The ECP virtual directory manages the Exchange Administration Center (EAC).

⚠**Warning:**
The ECP is the web-based user interface developed for Microsoft Exchange Server 2010. The Exchange Server 2013 Exchange Administration Center cmdlets for virtual directory still use ECP in the name, and the ECP cmdlets can be used to manage Exchange 2010 and Exchange 2013 ECP virtual directories.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-EcpVirtualDirectory -Identity <VirtualDirectoryIdParameter> [-
AdfsAuthentication <$true | $false>] [-AdminEnabled <$true | $false>] [-
BasicAuthentication <$true | $false>] [-Confirm [<SwitchParameter>]] [-
DigestAuthentication <$true | $false>] [-DomainController <Fqdn>] [-
ExtendedProtectionFlags <MultiValuedProperty>] [-ExtendedProtectionSPNList
<MultiValuedProperty>] [-ExtendedProtectionTokenChecking <None | Allow |
Require>] [-ExternalAuthenticationMethods <MultiValuedProperty>] [-
ExternalUrl <Uri>] [-FormsAuthentication <$true | $false>] [-GzipLevel
<Off | Low | High | Error>] [-InternalUrl <Uri>] [-LiveIdAuthentication
<$true | $false>] [-OwaOptionsEnabled <$true | $false>] [-WhatIf
[<SwitchParameter>]] [-WindowsAuthentication <$true | $false>]
```

Examples

EXAMPLE 1

This example disables Basic authentication on the default ECP virtual directory on the server Server01.

```
Set-EcpVirtualDirectory -Identity "Server01\ecp (default
Web site)" -BasicAuthentication:$false
```

EXAMPLE 2

This example turns off the Internet access to the EAC on server CAS01.

```
Set-EcpVirtualDirectory -Identity "CAS01\ecp (default Web
site)" -AdminEnabled $false
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Administration Center connectivity" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name or GUID of an ECP virtual directory. The *Identity* parameter is represented as: *ServerName*\ECP (*WebsiteName*). To manage the first ECP virtual directory created in an Exchange organization, you must run the **Set-EcpVirtualDirectory** cmdlet on the computer that includes the first ECP virtual directory. If you create additional ECP virtual directories, you can manage those remotely.
AdfsAuthentication	Optional	System.Boolean	The *AdfsAuthentication* parameter specifies that the ECP virtual directory allows users to authenticate through Active Directory

			Federation Services (AD FS) authentication. This parameter accepts `$true` or `$false`. The default value is `$false`. The ADFS authentication settings for **Set-OwaVirtualDirectory** and **Set-EcpVirtualDirectory** are related. You need to set the *AdfsAuthentication* parameter on **Set-EcpVirtualDirectory** to `$true` before you can set the *AdfsAuthentication* parameter on **Set-OwaVirtualDirectory** to `$true`. Likewise, you need to set the *AdfsAuthentication* parameter on **Set-OwaVirtualDirectory** to `$false` before you can set the *AdfsAuthentication* parameter on **Set-EcpVirtualDirectory** to `$false`.
AdminEnabled	Optional	System.Boolean	The *AdminEnabled* parameter specifies that the EAC isn't able to be accessed through the

			Internet. For more information, see Turn off access to the Exchange admin center. This parameter accepts `$true` or `$false`.
BasicAuthentication	Optional	System.Boolean	The *BasicAuthentication* parameter specifies whether Basic authentication is enabled on the ECP virtual directory. This parameter can be used with the *FormsAuthentication* parameter or with the *DigestAuthentication* and *WindowsAuthentication* parameters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DigestAuthentication	Optional	System.Boolean	The *DigestAuthentication* parameter specifies whether Digest

			authentication is enabled on the ECP virtual directory.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic

			may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended

			Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN` value. You specify the

			domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionTokenChecking* parameter defines how you want to use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and

			Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the client and server will work, but may not be as secure as a connection using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or

			server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalAuthenticationMethods	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalAuthenticationMethods* parameter specifies the authentication methods supported on the Exchange server from outside the firewall.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the host name used to connect to the Exchange server from outside the firewall. This setting is also important

			when Secure Sockets Layer (SSL) is used. This parameter must be set to allow the Autodiscover service to return the URL for the ECP virtual directory.
FormsAuthentication	Optional	System.Boolean	The *FormsAuthentication* parameter specifies whether forms-based authentication is enabled on the ECP virtual directory. If the *FormsAuthentication* parameter is set to `$true`, the *BasicAuthentication* parameter is set to `$true`, and the *DigestAuthentication* and *WindowsAuthentication* parameters are set to `$false`.
GzipLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.GzipLevel	The *GzipLevel* parameter sets Gzip configuration information for the ECP virtual directory.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the host name of the Exchange server for connections from inside

			the firewall. This setting is also important when SSL is used. This parameter must be set to allow the Autodiscover service to return the URL for the ECP virtual directory.
LiveIdAuthentication	Optional	System.Boolean	The *LiveIdAuthentication* parameter specifies whether Microsoft account (formerly known as Windows Live ID) authentication is enabled for the ECP virtual directory.
OwaOptionsEnabled	Optional	System.Boolean	The *OwaOptionsEnabled* parameter specifies that Outlook Web Access Options is enabled for end users. If this parameter is set to `$false`, users aren't able to access Outlook Web Access Options. You may want to disable access if your organization uses third-party provider tools. This parameter accepts `$true` or `$false`.
WhatIf	Optional	System.Management.Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsAuthenticati on	Optional	System.Boolean	The *WindowsAuthentication* parameter specifies whether Integrated Windows authentication is permitted on the ECP virtual directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-ImapConnectivity

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-ImapConnectivity** cmdlet to verify that the IMAP4 service is running as expected. The **Test-ImapConnectivity** cmdlet can be used to test the IMAP4 functionality for a specified Client Access server for all mailboxes on servers running Microsoft Exchange Server 2013 in the same Active Directory site.

```
Test-ImapConnectivity [-ClientAccessServer <ServerIdParameter>] [-
ConnectionType <Plaintext | Tls | Ssl>] [-MailboxCredential
<PSCredential>] [-PerConnectionTimeout <Int32>] [-PortClientAccessServer
<Int32>] [-TrustAnySSLCertificate <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-LightMode
<SwitchParameter>] [-MailboxServer <ServerIdParameter>] [-
MonitoringContext <SwitchParameter>] [-ResetTestAccountCredentials
<SwitchParameter>] [-Timeout <UInt32>] [-UserType <LEGACY | EDU | BPOS>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the IMAP4 connectivity for the Client Access server Contoso12 by using the credentials for the user contoso\kweku.

```
Test-ImapConnectivity -ClientAccessServer:Contoso12 -
MailboxCredential:(Get-Credential contoso\kweku)
```

EXAMPLE 2

This example tests the IMAP4 connectivity of the specific Client Access server Contoso12 and tests all Exchange mailboxes.

```
Test-ImapConnectivity -ClientAccessServer:Contoso12
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Test IMAP4 settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
ClientAccessServer	Optional	Microsoft.Exchange.Configuration.Tasks.Serve	The *ClientAccessServer* parameter specifies the

		rIdParameter	name of the Client Access server to test.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ConnectionType	Optional	Microsoft.Exchange.Monitoring.ProtocolConnectionType	The *ConnectionType* parameter specifies the type of connection used to connect to the Client Access server. This setting can be set to `Plaintext`, `Tls`, or `Ssl`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
LightMode	Optional	System.Management.Automation.SwitchParameter	The *LightMode* parameter instructs the command to perform

			only a test logon to the server using the IMAP4 protocol. If you don't use this parameter, the test also tests the sending and receiving of a message using the IMAP4 protocol.
MailboxCredential	Optional	System.Management.Automation.PSCredential	The *MailboxCredential* parameter specifies the mailbox credential for a single mailbox test.
MailboxServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *MailboxServer* parameter specifies the name of the Mailbox server to test. If not specified, all Mailbox servers in the local Active Directory site are tested.
MonitoringContext	Optional	System.Management.Automation.SwitchParameter	The *MonitoringContext* switch includes the associated monitoring events and performance counters in the results. You don't need to specify a value with this switch. Typically, you include the monitoring events and performance

			counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
PerConnectionTimeout	Optional	System.Int32	The *PerConnectionTimeout* parameter specifies the amount of time, in seconds, to wait per connection for the test operation to finish. The default value for the *PerConnectionTimeout* parameter is 120 seconds. You must specify a time-out value greater than 0 seconds and less than 120 seconds. We recommend that you configure this parameter with a value of 5 seconds or more.
PortClientAccessServer	Optional	System.Int32	The *PortClientAccessServer* parameter specifies the port to use to connect to the Client Access

			server. The default port is 143 for IMAP4. The valid range is from 0 through 65,535.
ResetTestAccountCredentials	Optional	System.Management.Automation.SwitchParameter	The *ResetTestAccountCredentials* switch resets the password for the test account that's used to run this command. The password for the test account is typically reset every seven days. Use this switch to force a password reset any time it's required for security reasons.
Timeout	Optional	System.UInt32	The *Timeout* parameter specifies the amount of time, in seconds, to wait for the test operation to finish. The default value for the *Timeout* parameter is 180 seconds. You must specify a time-out value greater than 0 seconds and less than 1 hour (3,600 seconds). We recommend that you configure this parameter with a value

			of 5 seconds or more.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	The *TrustAnySSLCertificate* parameter specifies whether Secure Sockets Layer (SSL) certificate validation failures are reported. This parameter instructs the command to check the IMAP4 service without generating an error when the certificate doesn't match the URL of the Client Access server.
UserType	Optional	Microsoft.Exchange.Monitoring.DatacenterUserType	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IMAPSettings

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-IMAPSettings** cmdlet to display the properties of a single server running Microsoft Exchange Server 2013 that has the Client Access server role installed and is running the IMAP4 service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ImapSettings [-DomainController <Fqdn>] [-Server <ServerIdParameter>]
```

Examples

EXAMPLE 1

This example displays the parameters and values for the Client Access server CAS01 running the IMAP4 service.

```
Get-IMAPSettings -Server CAS01
```

Detailed Description

With the **Get-IMAPSettings** cmdlet, you can view the settings for the IMAP4 service running on an Exchange Client Access server. Information can only be returned about servers located in the Exchange organization from which the cmdlet is being run.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "IMAP4 settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies whether the command returns the properties of an individual Client Access server in your organization for which you're viewing IMAP4 settings.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ImapSettings

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ImapSettings** cmdlet to set specific IMAP4 settings for the server running Microsoft Exchange Server 2013 that has the Client Access server role installed and that's running the Microsoft Exchange IMAP4 service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ImapSettings [-AuthenticatedConnectionTimeout <EnhancedTimeSpan>] [-
Banner <String>] [-CalendarItemRetrievalOption <iCalendar | intranetUrl |
InternetUrl | Custom>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-EnableExactRFC822Size <$true | $false>] [-
EnableGSSAPIAndNTLMAuth <$true | $false>] [-EnforceCertificateErrors
<$true | $false>] [-ExtendedProtectionPolicy <None | Allow | Require>] [-
ExternalConnectionSettings <MultiValuedProperty>] [-
InternalConnectionSettings <MultiValuedProperty>] [-
LiveIdBasicAuthReplacement <$true | $false>] [-LogFileLocation <String>]
[-LogFileRollOverSettings <Hourly | Daily | Weekly | Monthly>] [-LoginType
<PlainTextLogin | PlainTextAuthentication | SecureLogin>] [-
LogPerFileSizeQuota <Unlimited>] [-MaxCommandSize <Int32>] [-
MaxConnectionFromSingleIP <Int32>] [-MaxConnections <Int32>] [-
MaxConnectionsPerUser <Int32>] [-MessageRetrievalMimeFormat <TextOnly |
HtmlOnly | HtmlAndTextAlternative | TextEnrichedOnly |
TextEnrichedAndTextAlternative | BestBodyFormat | Tnef>] [-OwaServerUrl
<Uri>] [-PreAuthenticatedConnectionTimeout <EnhancedTimeSpan>] [-
ProtocolLogEnabled <$true | $false>] [-ProxyTargetPort <Int32>] [-Server
<ServerIdParameter>] [-ShowHiddenFoldersEnabled <$true | $false>] [-
SSLBindings <MultiValuedProperty>] [-SuppressReadReceipt <$true | $false>]
[-UnencryptedOrTLSBindings <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]] [-X509CertificateName <String>]
```

Examples

EXAMPLE 1

This example sets the plain text or TLS connection to the Client Access server CAS01. In this example, the connection uses an IP address of 10.0.0.0 and a port number of 995.

```
Set-ImapSettings -Server "CAS01" -UnencryptedOrTLSBindings
10.0.0.0:995
```

EXAMPLE 2

This example turns on IMAP4 protocol logging. It also changes the IMAP4 protocol logging directory to C:\Imap4Logging.

```
Set-ImapSettings -ProtocolLogEnabled $true -LogFileLocation
"C:\Imap4Logging"
```

EXAMPLE 3

This example changes the IMAP4 protocol logging to create a new log file when a log file reaches 2 megabytes (MB).

```
Set-ImapSettings -LogPerFileSizeQuota 2000000
```

EXAMPLE 4

This example changes the IMAP4 protocol logging to create a new log file every hour.

```
Set-ImapSettings -LogPerFileSizeQuota 0 -
LogFileRollOverSettings Hourly
```

Detailed Description

You can run the **Set-ImapSettings** cmdlet for a single Client Access server that has the Microsoft Exchange IMAP4 service installed, or for all Exchange Client Access servers that have the Microsoft Exchange IMAP4 service installed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "IMAP4 settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
AuthenticatedConnectionTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *AuthenticatedConnectionTimeout* parameter specifies the period of time to wait before closing an idle authenticated connection.

			To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Valid input for this parameter is `00:00:30` to `1:00:00`. The default setting is `00:30:00` or 30 minutes.
Banner	Optional	System.String	The *Banner* parameter specifies the banner string displayed after a connection to a Client Access server has been established.
CalendarItemRetrievalOption	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.CalendarItemRetrievalOptions	The *CalendarItemRetrievalOption* parameter specifies the type of calendar item returned when the calendar is accessed by using IMAP4. The default value is `iCalendar`. You can specify the value for this parameter by using a numerical value or text string. The following values are available: • `0` or `iCalendar` • `1` or `intranetUrl` • `2` or `InternetUrl` • `3` or `Custom`

			If you're using `3` or `custom`, you must specify the *OwaServerUrl* parameter setting.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EnableExactRFC822Size	Optional	System.Boolean	The *EnableExactRFC822Size* parameter calculates the exact size of each MIME message that can be retrieved from the server. When you set this parameter to `$true`, the exact size of MIME messages stored on the

			Exchange server is available to POP3 or IMAP4 client programs that rely on knowing the exact size of each MIME message. **Note:** This parameter is set to `$false` by default. If you don't set this option to `$true`, the size of each MIME message that the Exchange server returns to POP3 and IMAP4 client programs may be slightly different than the exact size of the message. Because setting this option to `$true` can negatively affect performance, you should only use this option if many of your users are using a client that requires knowing the exact size of MIME messages.
EnableGSSAPIAndNTLMAuth	Optional	System.Boolean	The *EnableGSSAPIAndNTLMAuth* parameter specifies whether connections can use Integrated Windows authentication (NTLM) using the Generic Security Services application programming interface (GSSAPI). This setting applies to connections

			where Transport Layer Security (TLS) is disabled. This parameter is set to `$true` by default. You can disable NTLM for IMAP4 connections by setting the parameter value to `$false`. NTLM authentication isn't supported for IMAP4 connections in Microsoft Exchange Server 2010 release to manufacturing (RTM). Support for NTLM authentication for IMAP4 connections was brought back in Exchange 2010 Service Pack 1 (SP1).
EnforceCertificateErrors	Optional	System.Boolean	The *EnforceCertificateErrors* parameter specifies whether to enforce valid Secure Sockets Layer (SSL) certificates. To use this parameter, specify the destination Client Access server for which you want to enforce valid SSL certificates. If the *EnforceCertificateErrors* parameter is set to `$true` and the proxy's target certificate isn't valid, the

			proxy logon attempt fails. The default setting is `$false`.
ExtendedProtectionPolicy	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionPolicy* parameter specifies how you want to use Extended Protection for Authentication for IMAP4 connections. By default, this parameter is set to `None`. The possible values are: • `None` Extended Protection for Authentication won't be used. • `Allow` Extended Protection for Authentication will be used only if the connecting IMAP4 connection supports it. Otherwise, the connections will be established without Extended Protection for Authentication. • `Require` Extended Protection for Authentication will be required for all IMAP4 connections. If the connecting host doesn't support Extended Protection for Authentication, the connection will be rejected.

			Extended Protection for Authentication enhances the protection and handling of credentials when authenticating network connections using Integrated Windows authentication. Integrated Windows authentication is also known as NTLM. We strongly recommend that you use Extended Protection for Authentication if you're using Integrated Windows authentication. To use Extended Protection for Authentication, the client and server computers must meet specific requirements. These include operating system requirements and security update requirements. In addition, the IMAP4 client program must support the use of Extended Protection for Authentication.
ExternalConnectionSettings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalConnectionSettings* parameter specifies the host name, port, and

			encryption type that Exchange uses when IMAP4 clients connect to their email from outside the corporate network. Enter a value using the format: *<HostName>:<Port>:<Encryption Type>*. The *<Encryption Type>* part of the multivalued value is optional. Valid values for *<Encryption Type>* are either TLS (Transport Layer Security) or SSL.
InternalConnectionSettings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *InternalConnectionSettings* parameter specifies the host name, port, and encryption type used when IMAP4 clients connect to their email from inside the corporate network. This setting is also used to specify the host name, port, and encryption type used when a user connection is forwarded to another Client Access server. Enter a value using the format:

			<HostName>:*<Port>*:*<Encryption Type>*. The *<Encryption Type>* part of the multivalued value is optional. Valid values for *<Encryption Type>* are either TLS or SSL. If the server name is Server01 and the domain is Contoso.com, the default value is `Server01.Contoso.com:993:SSL, Server01.Contoso.com:143:TLS`.
LiveIdBasicAuthReplacement	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LogFileLocation	Optional	System.String	The *LogFileLocation* parameter specifies the location for the IMAP4 protocol log files. By default, IMAP4 protocol log files are located in the C:\Program Files\Microsoft\Exchange Server\V15\Logging\Imap4 directory.
LogFileRollOverSettings	Optional	Microsoft.Exchange.Diagnostics.LogFileRollOver	The *LogFileRollOverSettings* parameter specifies how frequently IMAP4 protocol logging creates a new log file. By default, a new log

			file is created daily. You can specify the value for this parameter by using a numerical value or text string. The possible values are: • 1 or Hourly • 2 or Daily • 3 or Weekly • 4 or Monthly This setting only applies when the value for the *LogPerFileSizeQuota* parameter is set to 0.
LoginType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.LoginOptions	The *LoginType* parameter specifies the authentication setting used for the Client Access server running the Microsoft Exchange IMAP4 service. The default value is SecureLogin. You can specify the value for this parameter by using a numerical value or text string. The possible values are: • 1 or PlainTextLogin • 2 or PlainTextAuthentication • 3 or SecureLogin
LogPerFileSizeQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *LogPerFileSizeQuota* parameter specifies the maximum size of an

			IMAP4 protocol log file in bytes. By default, this value is set to 0. When this value is set to 0, a new protocol log file is created at the frequency specified by the *LogFileRollOverSettings* parameter.
MaxCommandSize	Optional	System.Int32	The *MaxCommandSize* parameter specifies the maximum size of a single command. The default size is 10240 bytes. The possible values are from 1024 through 16384 bytes.
MaxConnectionFromSingleIP	Optional	System.Int32	The *MaxConnectionFromSingleIP* parameter specifies the number of connections that the specified server accepts from a single IP address. The default value is 2147483647. The possible values are from 1 through 2147483647.
MaxConnections	Optional	System.Int32	The *MaxConnections* parameter specifies the total number of

			connections that the specified server accepts. This includes authenticated and unauthenticated connections. The default value is 2147483647. The possible values are from 1 through 2147483647.
MaxConnectionsPerUser	Optional	System.Int32	The *MaxConnectionsPerUser* parameter specifies the maximum number of connections that the Client Access server accepts from a particular user. The default value is 16. The possible values are from 1 through 2147483647.
MessageRetrievalMimeFormat	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MimeTextFormat	The *MessageRetrievalMimeFormat* parameter specifies the format of the messages retrieved from the server. The default value is `BestBodyFormat`. You can specify the value for this parameter by using a numerical value or text string. The possible values are:

			• 0 or `TextOnly` • 1 or `HtmlOnly` • 2 or `HtmlAndTextAlternative` • 3 or `TextEnrichedOnly` • 4 or `TextEnrichedAndTextAlternative` • 5 or `BestBodyFormat` • 6 or `Tnef` For more information, see Configure POP3 and IMAP4 message retrieval format options.
OwaServerUrl	Optional	System.Uri	The *OwaServerUrl* parameter specifies the Client Access server from which to retrieve calendar information for instances of custom Microsoft Office Outlook Web App calendar items.
PreAuthenticatedConnectionTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *PreAuthenticatedConnectionTimeout* parameter specifies the period of time to wait before closing an idle connection that isn't authenticated. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.

			Valid input for this parameter is `00:00:30` to `1:00:00`.The default value is `00:01:00` or one minute.
ProtocolLogEnabled	Optional	System.Boolean	The *ProtocolLogEnabled* parameter specifies whether to enable protocol logging. For more information, see Configure protocol logging for POP3 and IMAP4.
ProxyTargetPort	Optional	System.Int32	The *ProxyTargetPort* parameter specifies the port on the Exchange Server 2003 back-end server to which the Microsoft Exchange IMAP4 service on a Client Access server relays commands. The default port is 9933.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies an individual Client Access server in your organization for which you're specifying IMAP4 settings.
ShowHiddenFoldersEnabled	Optional	System.Boolean	The *ShowHiddenFoldersEnabl*

			ed parameter specifies whether hidden folders are visible. If the value is set to `$true`, hidden folders are visible. The default value is `$false`.
SSLBindings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SSLBindings* parameter specifies whether the command sets the IP address and port number to use for communication for an SSL session. This is a multivalued property. Enter a value using the format: <*IP address*>:<*Port number*>. The default value is `0.0.0.0:993`.
SuppressReadReceipt	Optional	System.Boolean	The *SuppressReadReceipt* parameter specifies whether to stop duplicate read receipts from being sent to IMAP4 senders that are using the **Send read receipts for messages I send** option in their IMAP4 email program. By default, this option is set to `$false`. By default, IMAP4 senders that use the **Send read**

			receipts for messages I send option receive a read receipt in both of the following circumstances: • When messages they send are downloaded by the recipient. • When the recipient opens the message. The following are valid values and descriptions for this parameter: • `$false` IMAP4 users are sent a read receipt each time a recipient downloads a message. IMAP4 users are also sent a read receipt when the user opens the message. • `$true` IMAP4 users that use the **Send read receipts for messages I send** option in their email client programs only receive a read receipt when the recipient opens the message.
UnencryptedOrTLSBindings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UnencryptedOrTLSBindings* parameter specifies the IP address and port number for communication over the TLS-encrypted connection

			or the connection that isn't encrypted. This is a multivalued property. Enter a value using the format: *<IP address>*:*<Port number>*. The default value is `0.0.0.0:143`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
X509CertificateName	Optional	System.String	The *X509CertificateName* parameter specifies the host name in the SSL certificate from the **Associated Subject** field. This is a multivalued property that contains both the IP address and the port setting. Enter a value using the format: *<IP address>*:*<Port number>*.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxCalendarConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxCalendarConfiguration** cmdlet to show the calendar settings for a specified mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxCalendarConfiguration -Identity <MailboxIdParameter> [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves all the calendar settings for Kai's mailbox where the *Identity* parameter is specified in the alias format.

```
Get-MailboxCalendarConfiguration -Identity kai | Format-
List
```

EXAMPLE 2

This example uses the *Identity* parameter specified in the *domain\account* format and returns the calendar settings for Tony's mailbox.

```
Get-MailboxCalendarConfiguration -Identity contoso\tony
```

EXAMPLE 3

This example requests that the domain controller DC1 retrieves calendar settings for Kai's mailbox from Active Directory.

```
Get-MailboxCalendarConfiguration -Identity kai -
DomainController DC1
```

Detailed Description

The **Get-MailboxCalendarConfiguration** cmdlet returns settings for the calendar of the specified mailbox, including the following:

- **Workdays** Days that appear in the calendar as work days in Microsoft Office Outlook Web App
- **WorkingHoursStartTime** Time that the calendar work day starts
- **WorkingHoursEndTime** Time that the calendar work day ends
- **WorkingHoursTimeZone** Time zone set on the mailbox for the working hours start and end times
- **WeekStartDay** First day of the calendar work week
- **ShowWeekNumbers** Number for each week ranging from 1 through 52 for the calendar while in month view in Outlook Web App
- **TimeIncrement** Increments in minutes in which the calendar displays the time in Outlook Web App
- **RemindersEnabled** Whether Outlook Web App provides a visual cue when a calendar reminder is due
- **ReminderSoundEnabled** Whether a sound is played when a calendar reminder is due
- **DefaultReminderTime** Length of time before each meeting or appointment that the calendar in Outlook Web App shows the reminder

To see all of the settings returned, pipeline the command to the **Format-List** command. To view a code sample, see "EXAMPLE 1."

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Calendar configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.Mail	The *Identity* parameter specifies a unique

		boxIdParameter	identifier for the mailbox. You can use the following values: • GUID • ADObjectID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxCalendarConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailboxCalendarConfiguration** cmdlet to apply calendar settings for users using Microsoft Office Outlook Web App calendars. This affects how the user's calendar looks and how reminders work in Outlook Web App. This also affects settings that define how meeting invitations, responses, and notifications are sent to the user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxCalendarConfiguration -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DefaultReminderTime <TimeSpan>] [-DomainController
<Fqdn>] [-FirstWeekOfYear <LegacyNotSet | FirstDay | FirstFourDayWeek |
FirstFullWeek>] [-RemindersEnabled <$true | $false>] [-
ReminderSoundEnabled <$true | $false>] [-ShowWeekNumbers <$true | $false>]
[-TimeIncrement <FifteenMinutes | ThirtyMinutes>] [-WeekStartDay <Sunday |
Monday | Tuesday | Wednesday | Thursday | Friday | Saturday>] [-WhatIf
[<SwitchParameter>]] [-WorkDays <None | Sunday | Monday | Tuesday |
Wednesday | Thursday | Friday | Saturday | Weekdays | WeekendDays |
AllDays>] [-WorkingHoursEndTime <TimeSpan>] [-WorkingHoursStartTime
<TimeSpan>] [-WorkingHoursTimeZone <ExTimeZoneValue>]
```

Examples

EXAMPLE 1

This example disables the calendar reminders for the user Peter.

```
Set-MailboxCalendarConfiguration -Identity Peter -
RemindersEnabled $false
```

EXAMPLE 2

This example sets the time zone of the work hours' start and end times to Pacific Standard Time for the user Peter.

```
Set-MailboxCalendarConfiguration -Identity Peter -
WorkingHoursTimeZone "Pacific Standard Time"
```

EXAMPLE 3

This example sets the working day's starting hour in Tony's calendar to the specified time.

```
Set-MailboxCalendarConfiguration -Identity tony -
WorkingHoursStartTime 07:00:00
```

Detailed Description

The **Set-MailboxCalendarConfiguration** cmdlet was created primarily to allow users to manage their calendar settings. However, administrators who have the Organization Management or Recipient Management management roles may configure the calendar settings for users by using the Exchange Management Shell.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Calendar configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the user account to be set.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DefaultReminderTime	Optional	System.TimeSpan	The *DefaultReminderTime* parameter specifies the length of time before each meeting or appointment

			that Outlook Web App should show the reminder. Values are expressed in "*DD.HH:MM:SS*" time span format within quotation marks, where *DD* refers to days, *HH* refers to hours, *MM* refers to minutes, and *SS* refers to seconds. Valid values are as follows: 00:00:00, 00:05:00, 00:10:00, 00:15:00, 00:30:00, 01:00:00, 02:00:00, 03:00:00, 04:00:00, 08:00:00, 12:00:00, 1.00:00:00, 2.00:00:00, 3.00:00:00, 7.00:00:00, 14.00:00:00
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

FirstWeekOfYear	Optional	Microsoft.Exchange.Data.Storage.Management.FirstWeekRules	The *FirstWeekOfYear* parameter specifies when the first week of the year will start when show week numbers has been turned on in Outlook Web App by the user, or by using the Shell to set the *ShowWeekNumbers* parameter to `$true`. The *FirstWeekOfYear* parameter can have the following values: • `LegacyNotSet` A null value that appears only when the mailbox has been moved from an earlier version of Exchange. It can't be set using the Shell. • `FirstDay` Causes the week numbers to start on the first day of the year. • `FirstFourDayWeek` Causes the week numbers to start on the first week that has at least four days. • `FirstFullWeek` Causes the week numbers to start on the first seven day week. **Note:** The first day of the week can be set by using the *WeekStartDay* parameter.
RemindersEnabled	Optional	System.Boolean	The *RemindersEnabled*

			parameter specifies whether Outlook Web App provides a visual indicator when a calendar reminder is due. Reminders are enabled by default (`$true`).
ReminderSoundEnabled	Optional	System.Boolean	The *ReminderSoundEnabled* parameter specifies whether a sound is played when a reminder is due. The reminder sound is enabled by default (`$true`).
ShowWeekNumbers	Optional	System.Boolean	The *ShowWeekNumbers* parameter specifies whether the date picker in the Outlook Web App calendar shows the week number.
TimeIncrement	Optional	Microsoft.Exchange.Data.Storage.Management.HourIncrement	The *TimeIncrement* parameter specifies the minutes in which the Outlook Web App calendar shows time. For example, `FifteenMinutes` is mapped to 15 minutes.
WeekStartDay	Optional	Microsoft.Exchange.Data.Storage.Management.DayOfWeek	The *WeekStartDay* parameter specifies the first day of the work week.

			The valid values for this parameter are `Sunday`, `Monday`, `Tuesday`, `Wednesday`, `Thursday`, `Friday`, and `Saturday`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WorkDays	Optional	Microsoft.Exchange.Data.DaysOfWeek	The *WorkDays* parameter specifies the work days in the calendar. Valid values for this parameter are `Weekdays`, `AllDays`, `WeekEndDays`, `Sunday`, `Monday`, `Tuesday`, `Wednesday`, `Thursday`, `Friday`, `Saturday` and `None`. The default value is `Weekdays`. You can specify multiple values separated by commas, but redundant

			values are ignored. For example, entering `weekdays,Monday` results in the value `weekdays`.
WorkingHoursEndTime	Optional	System.TimeSpan	The *WorkingHoursEndTime* parameter specifies the time in hours, minutes, and seconds (*hh:mm:ss*) that the work day ends. For example, to state the time as 5:00 P.M., use 17:00:00.
WorkingHoursStartTime	Optional	System.TimeSpan	The *WorkingHoursStartTime* parameter specifies the time in hours, minutes, and seconds (*hh:mm:ss*) that the work day starts. For example, to state the time as 8:00 A.M., use 08:00:00.
WorkingHoursTimeZone	Optional	Microsoft.Exchange.Data.Storage.Management.ExTimeZoneValue	The *WorkingHoursTimeZone* parameter specifies the time zone used by the user's working hour start and end times. Two types of formats are supported as follows: • GMT (Greenwich Mean

			Time), for example, GMT-08:00 • Time zone key name, for example, Pacific Standard Time (PST)

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxMessageConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxMessageConfiguration** cmdlet to view the Microsoft Outlook Web App settings that are applied to specific mailboxes.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxMessageConfiguration -Identity <MailboxIdParameter> [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-
ReadFromDomainController <SwitchParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns the Outlook Web App settings for Tony's mailbox.

```
Get-MailboxMessageConfiguration tony@contoso.com
```

EXAMPLE 2

This example returns the Outlook Web App settings for Tony's mailbox, and specifies the domain controller that's used to get those settings.

```
Get-MailboxMessageConfiguration tony@contoso.com -
DomainController DC1
```

Detailed Description

The **Get-MailboxMessageConfiguration** cmdlet shows Outlook Web App settings for the specified mailbox. These settings are not used in Microsoft Outlook, Microsoft Exchange ActiveSync, or other email clients. These settings are applied in Outlook Web App only. Settings that contain the word *Mobile* are applied in Microsoft OWA for Devices only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Message configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox. You can use any value that uniquely identifies the mailbox. For example: • Alias • Distinguished name (DN) • GUID • Name • Display name • LegacyExchangeDN • Email address

Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest, and if you don't use this

			parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Exchange.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxMessageConfiguration

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailboxMessageConfiguration** cmdlet to configure the Microsoft Outlook Web App settings that are applied to specific mailboxes.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxMessageConfiguration -Identity <MailboxIdParameter> [-
AfterMoveOrDeleteBehavior <OpenPreviousItem | OpenNextItem |
ReturnToView>] [-AlwaysShowBcc <$true | $false>] [-AlwaysShowFrom <$true |
$false>] [-AutoAddSignature <$true | $false>] [-AutoAddSignatureOnMobile
<$true | $false>] [-CheckForForgottenAttachments <$true | $false>] [-
Confirm [<SwitchParameter>]] [-ConversationSortOrder <Chronological | Tree
| NewestOnTop | NewestOnBottom | ChronologicalNewestOnTop |
ChronologicalNewestOnBottom | TreeNewestOnBottom>] [-DefaultFontColor
<String>] [-DefaultFontFlags <Normal | Bold | Italic | Underline | All>]
[-DefaultFontName <String>] [-DefaultFontSize <Int32>] [-DefaultFormat
<Html | PlainText>] [-DomainController <Fqdn>] [-EmailComposeMode <Inline
| SeparateForm>] [-EmptyDeletedItemsOnLogoff <$true | $false>] [-
HideDeletedItems <$true | $false>] [-IgnoreDefaultScope <SwitchParameter>]
[-NewItemNotification <None | Sound | EMailToast | VoiceMailToast |
FaxToast | All>] [-PreviewMarkAsReadBehavior <Delayed | OnSelectionChange
| Never>] [-PreviewMarkAsReadDelaytime <Int32>] [-ReadReceiptResponse
<DoNotAutomaticallySend | AlwaysSend | NeverSend>] [-SendAddressDefault
<String>] [-ShowConversationAsTree <$true | $false>] [-SignatureHtml
<String>] [-SignatureText <String>] [-SignatureTextOnMobile <String>] [-
UseDefaultSignatureOnMobile <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets items deleted from a conversation thread to not show in the list view of the conversation in Outlook Web App for Kai's mailbox.

```
Set-MailboxMessageConfiguration kai@contoso.com -
HideDeletedItems $true
```

EXAMPLE 2

This example sets the compose email message form to always show the Bcc field in Outlook Web App for Kai's mailbox.

```
Set-MailboxMessageConfiguration kai@contoso.com -
AlwaysShowBcc $true
```

Detailed Description

The **Set-MailboxMessageConfiguration** cmdlet configures Outlook Web App settings for the specified mailbox. These settings include email signature, message format, message options, read receipts, reading pane, and conversations. These settings are not used in Microsoft Outlook, Exchange ActiveSync, or other email clients. These settings are applied in Outlook Web App only. Settings that contain the word *Mobile* are applied in Microsoft OWA for Devices only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "User options" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox. You can use any value that uniquely identifies the mailbox. For example: • Alias • Distinguished name (DN) • GUID • Name • Display name • LegacyExchangeDN • Email address
AfterMoveOrDeleteBehavior	Optional	Microsoft.Exchange.Data.Storage.Management.AfterMoveOrDeleteBehavior	The *AfterMoveOrDeleteBehavior* parameter specifies the behavior after moving or deleting an email item in Outlook Web App. You

			can use the following values: • `OpenPreviousItem` • `OpenNextItem` • `ReturnToView` The default value is `OpenNextItem`.
AlwaysShowBcc	Optional	System.Boolean	The *AlwaysShowBcc* parameter shows or hides the blind carbon copy (**Bcc**) field when the user creates messages in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
AlwaysShowFrom	Optional	System.Boolean	The *AlwaysShowFrom* parameter shows or hides the **From** field when the user creates messages in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
AutoAddSignature	Optional	System.Boolean	The *AutoAddSignature* parameter automatically adds the email signature specified by the *SignatureText* or *SignatureHTML*

			parameters to messages when the user creates messages in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. The email signature specified by the *SignatureText* parameter is added to plain text messages. The email signature specified by the *SignatureHTML* parameter is added to HTML-formatted messages.
AutoAddSignatureOnMobile	Optional	System.Boolean	The *AutoAddSignatureOnMobile* parameter automatically adds the signature specified by the *SignatureTextOnMobile* parameter to messages when the user creates messages in OWA for Devices. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.

CheckForForgottenAttachments	Optional	System.Boolean	The *CheckForForgottenAttachments* parameter shows or hides the attachment warning prompt when the user creates messages in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. For example, the user creates a message that includes the text "Please see the attached Word document", but the user doesn't attach a file, and clicks **Send**. If this value is set to `$true`, the user gets a warning prompt so they can go back to the message and attach a file. If this value is set to `$false`, the user doesn't get the warning prompt.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You

			don't have to specify a value with the *Confirm* switch.
ConversationSortOrder	Optional	Microsoft.Exchange.Data.Storage.Management.ConversationSortOrder	The *ConversationSortOrder* parameter specifies the sorting of messages in the reading pane in **Conversation** view for the user in Outlook Web App. You can use the following values: • `Chronological` • `Tree` • `NewestOnTop` • `NewestOnBottom` • `ChronologicalNewestOnTop` • `ChronologicalNewestOnBottom` • `TreeNewestOnBottom` The default value is `ChronologicalNewestOnTop`.
DefaultFontColor	Optional	System.String	The *DefaultFontColor* parameter specifies the default text color when the user creates messages in Outlook Web App. This parameter accepts a hexadecimal color code value in the format `#xxxxxx`. The default value is `#000000`. If the string value is unrecognized, the

			browser application uses a default font color to display the text.
DefaultFontFlags	Optional	Microsoft.Exchange.Data.Storage.Management.FontFlags	The *DefaultFontFlags* parameter specifies the default text effect when the user creates messages in Outlook Web App. You can use the following values: • `Normal` • `Bold` • `Italic` • `Underline` • `All` The default value is `Normal`.
DefaultFontName	Optional	System.String	The *DefaultFontName* parameter specifies the default font when the user creates messages in Outlook Web App. The default value is `Calibri`. If the font name value is unrecognized, the browser application uses a default font to display the text.
DefaultFontSize	Optional	System.Int32	The *DefaultFontSize* parameter specifies the default text size when the user creates messages in Outlook Web App.

			Valid input for this parameter is an integer between 1 and 7. The default value is 3, which corresponds to a 12 point font size.
DefaultFormat	Optional	Microsoft.Exchange.Data.Storage.Management.MailFormat	The *DefaultFormat* parameter specifies the default message format when the user creates messages in Outlook Web App. Accepted values are `Html` and `PlainText`. The default value is `Html`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EmailComposeMode	Optional	Microsoft.Exchange.Data.Storage.Management.EmailComposeMode	The *EmailComposeMode* parameter specifies how the user creates messages in Outlook Web App. You can use the following values: • `Inline` New messages

			and replies are created in a new browser window. • `SeparateForm` New messages and replies are created in the preview pane. The default value is `Inline`.
EmptyDeletedItemsOnLogoff	Optional	System.Boolean	The *EmptyDeletedItemsOnLogoff* parameter specifies whether to delete items from the Deleted Items folder when the user logs out of Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
HideDeletedItems	Optional	System.Boolean	The *HideDeletedItems* parameter shows or hides deleted messages in **Conversation** view for the user in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchPar	The *IgnoreDefaultScope* parameter instructs the command to ignore the

		ameter	default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
NewItemNotification	Optional	Microsoft.Exchange.Data.Storage.Management.NewItemNotification	The *NewItemNotification* parameter specifies how to provide notification for the arrival of new items for the user in Outlook Web App. You can use the following values:

			• `Sound` • `EMailToast` • `VoiceMailToast` • `FaxToast` • `None` • `All` The default value is `All`.
PreviewMarkAsReadBehavior	Optional	Microsoft.Exchange.Data.Storage.Management.PreviewMarkAsReadBehavior	The *PreviewMarkAsReadBehavior* parameter specifies the options for marking an item as Read in the reading pane for the user in Outlook Web App. You can use the following values: • `Delayed` This value uses the delay interval specified by the *PreviewMarkAsReadDelaytime* parameter. • `OnSelectionChange` • `Never` The default value is `OnSelectionChange`.
PreviewMarkAsReadDelaytime	Optional	System.Int32	The *PreviewMarkAsReadDelaytime* parameter specifies the time in seconds to wait before marking an item as **Read** for the user in Outlook Web App. Valid input for this parameter is an integer between 0 and 30. The default value is 5 seconds.

			This parameter is meaningful only if you set the *PreviewMarkAsReadBehavior* parameter to the value `Delayed`.
ReadReceiptResponse	Optional	Microsoft.Exchange.Data.Storage.Management.ReadReceiptResponse	The *ReadReceiptResponse* parameter specifies how to respond to requests for read receipts for the user in Outlook Web App. You can use the following values: • `DoNotAutomaticallySend` • `AlwaysSend` • `NeverSend` The default value is `DoNotAutomaticallySend`.
SendAddressDefault	Optional	System.String	This parameter is available only in the cloud-based service. The *SendAddressDefault* parameter specifies the default From email address when the user has POP, IMAP, or Hotmail subscriptions configured on their mailbox. Users can override the default From address when they create an email message in Outlook Web App.

			You can use one of the following values: • Blank, which is represented by the value `$null`. This indicates no default From address is specified. • The user's primary email address. For example, bob@contoso.com. • The GUID of a POP, IMAP, or Hotmail subscription that's configured on the user's mailbox. By default, no default From address is specified on the mailbox. When no default From address is specified, the default behavior is: • The primary email address on the mailbox is used for all new messages. • The To address of the incoming message is used as the From address for all replies or forwarded messages.

			You can find the available values for *SendAddressDefault* on a mailbox by running the command **Get-SendAddress -Mailbox <mailbox>**.
ShowConversationAsTree	Optional	System.Boolean	The *ShowConversationAsTree* parameter specifies how to sort messages in the list view in an expanded conversation for the user in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
SignatureHtml	Optional	System.String	The *SignatureHtml* parameter specifies the email signature that's available to the user in HTML-formatted messages in Outlook Web App. You can use plain text or text with HTML tags. However, any JavaScript code is removed. To automatically add this email signature to HTML-

			formatted messages created by the user in Outlook Web App, the *AutoAddSignature* parameter must be set to `$true`.
SignatureText	Optional	System.String	The *SignatureText* parameter specifies the email signature that's available to the user in plain text messages in Outlook Web App. This parameter supports all Unicode characters. To automatically add the email signature to plain text messages created by the user in Outlook Web App, the *AutoAddSignature* parameter must be set to the value `$true`.
SignatureTextOnMobile	Optional	System.String	The *SignatureTextOnMobile* parameter specifies the email signature that's available in messages created by the user in OWA for Devices. This parameter supports all Unicode characters.

			To automatically add the email signature to messages created by the user in OWA for Devices, the *AutoAddSignatureOnMobile* parameter must be set to the value `$true`.
UseDefaultSignatureOnMobile	Optional	System.Boolean	The *UseDefaultSignatureOnMobile* parameter specifies whether to add the default email signature to messages created by the user in OWA for Devices. The user configures the default signature in Microsoft Outlook. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxRegionalConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxRegionalConfiguration** cmdlet to retrieve regional settings such as time zone, time format, date, and language settings for a specified mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxRegionalConfiguration -Identity <MailboxIdParameter> [-
DomainController <Fqdn>] [-VerifyDefaultFolderNameLanguage
<SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves regional settings for Tony's mailbox.

```
Get-MailboxRegionalConfiguration -Identity tony
```

EXAMPLE 2

This example retrieves regional settings for Tony's mailbox, requesting information from the

domain controller closest to Tony's mailbox.

```
Get-MailboxRegionalConfiguration -Identity tony -
DomainController "DC1"
```

EXAMPLE 3

This example, in addition to returning regional settings for Tony's mailbox, indicates whether the default folder names of the mailbox are localized in the locale selected for the mailbox.

```
Get-MailboxRegionalConfiguration -Identity tony -
VerifyDefaultFolderNameLanguage $true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Message configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter identifies the mailbox. You can use one of the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-

			premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
VerifyDefaultFolderNameLanguage	Optional	System.Management.Automation.SwitchParameter	If the *VerifyDefaultFolderNameLanguage* parameter is set to `$true`, the **DefaultFolderNameMatchingUserLanguage** property returned by the task indicates whether the mailbox default folder names are localized with the language specified for the mailbox. If this parameter isn't specified, the property isn't returned by the task. This parameter can only be used by the mailbox owner. If a non-mailbox owner tries to run the command on the mailbox with this parameter, an error is reported. The default value is `$false`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxRegionalConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailboxRegionalConfiguration** cmdlet to set regional settings such as time, date, or language for the specified mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxRegionalConfiguration -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DateFormat <String>] [-DomainController <Fqdn>] [-
Language <CultureInfo>] [-LocalizeDefaultFolderName <SwitchParameter>] [-
TimeFormat <String>] [-TimeZone <ExTimeZoneValue>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets Tony's mailbox to have the language set as Brazilian Portuguese.

```
Set-MailboxRegionalConfiguration -Identity Tony -Language
pt-br
```

EXAMPLE 2

This example sets the date format for Tony's mailbox.

```
Set-MailboxRegionalConfiguration -Identity Tony -DateFormat
"d/m/yyyy"
```

EXAMPLE 3

This example sets Tony's mailbox to have the language set as Danish Denmark and the date set in the format of day/month/year.

```
Set-MailboxRegionalConfiguration -Identity Tony -Language
da-dk -DateFormat "dd-mm-yyyy"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Spelling configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox. You can use the following values: • GUID • ADObjectID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
Confirm	Optional	System.Management.Automation.SwitchPar	The *Confirm* switch can be used to suppress the

		ameter	confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DateFormat	Optional	System.String	The *DateFormat* parameter specifies the format for displaying the date, for example, *m/d/yyyy*, in the mailbox for a specified region based on the *Language* selection.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Language	Optional	System.Globalization.CultureInfo	The *Language* parameter specifies the language setting such as en-us that would apply for the mailbox.

LocalizeDefaultFolderName	Optional	System.Management.Automation.SwitchParameter	The *LocalizeDefaultFolderName* parameter specifies whether the default folder names of the mailbox are localized with the current or specified language. When the parameter is set to `$true`, the default folder names of the mailbox are localized with the current or specified language. This parameter can only be used by the mailbox owner running the task. If a non-mailbox owner tries to configure this setting on the mailbox, an error is reported. By default, the value is set to `$false`.
TimeFormat	Optional	System.String	The *TimeFormat* parameter specifies the format (for example, *h:mm tt*, as in 3:45 A.M.) used by the mailbox to display time for the specified region.
TimeZone	Optional	Microsoft.Exchange.Data.Storage.Management.ExTimeZoneValue	The *TimeZone* parameter specifies the time zone, such as Pacific Standard Time, that the mailbox in

			the specified region uses. The default value is the time zone setting on the server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxSpellingConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxSpellingConfiguration** cmdlet to retrieve the Microsoft Office Outlook Web App spelling checker settings of a specified user. For example, users can set their dictionary language and configure the spelling checker to ignore mixed digits and words in all uppercase.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxSpellingConfiguration -Identity <MailboxIdParameter> [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves the Outlook Web App options of user Tony.

```
Get-MailboxSpellingConfiguration -Identity Tony
```

EXAMPLE 2

This example returns the Outlook Web App spelling checker options for Tony's mailbox by specifying domain controller DC1 to get the information from Active Directory.

```
Get-MailboxSpellingConfiguration -Identity Tony -
DomainController DC1
```

EXAMPLE 3

This example returns the Outlook Web App spelling checker options for Tony's mailbox by specifying the identity of the mailbox in the format *domain\account*.

```
Get-MailboxSpellingConfiguration -Identity contoso\tony
```

Detailed Description

The **Get-MailboxSpellingConfiguration** cmdlet is primarily used to populate the spelling checker settings for end users in Outlook Web App. Administrators can also view users' settings by running this cmdlet in the Exchange Management Shell. The following spelling checker settings are retrieved by the cmdlet for the specified mailbox:

- **Identity** This setting specifies the mailbox identity.
- **CheckBeforeSend** This setting specifies whether Outlook Web App checks the spelling of every message when the user clicks **Send** in the new message form.
- **DictionaryLanguage** This setting specifies the dictionary language used when the spelling checker checks the spelling in messages.
- **IgnoreMixedDigits** This setting specifies whether the spelling checker ignores words that

contain numbers.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Spelling configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox. You can use one of the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxSpellingConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailboxSpellingConfiguration** cmdlet to modify Microsoft Office Outlook Web App spelling checker options for a specified user. For example, you can set the dictionary language and configure the spelling checker to ignore mixed digits or words in all uppercase.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxSpellingConfiguration -Identity <MailboxIdParameter> [-
CheckBeforeSend <$true | $false>] [-Confirm [<SwitchParameter>]] [-
DictionaryLanguage <Spanish | Arabic | Danish | Dutch | EnglishAustralia |
EnglishCanada | EnglishUnitedKingdom | EnglishUnitedStates | Finnish |
French | GermanPostReform | GermanPreReform | Hebrew | Italian | Korean |
NorwegianBokmal | NorwegianNynorsk | PortuguesePortugal | PortugueseBrasil
| Swedish | Catalan>] [-DomainController <Fqdn>] [-IgnoreMixedDigits
<$true | $false>] [-IgnoreUppercase <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the spelling checker of user Peter to ignore words in all uppercase letters.

```
Set-MailboxSpellingConfiguration -Identity Peter -
IgnoreUppercase $true
```

EXAMPLE 2

This example sets the spelling checker to ignore words that contain only uppercase letters for messages sent from Kai's mailbox.

```
Set-MailboxSpellingConfiguration -Identity kai -
IgnoreUppercase $true
```

EXAMPLE 3

This example sets the spelling checker to ignore words containing numbers for messages sent from Kai's mailbox.

```
Set-MailboxSpellingConfiguration -IgnoreMixedDigits $true -
Identity kai
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Spelling configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox. You can use the following values: • GUID • ADObjectID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
CheckBeforeSend	Optional	System.Boolean	The *CheckBeforeSend* parameter specifies

			whether Outlook Web App checks the spelling for every message when the user clicks **Send** in the new message form. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DictionaryLanguage	Optional	Microsoft.Exchange.Data.Storage.Management.SpellcheckerSupportedLanguage	The *DictionaryLanguage* parameter specifies the dictionary language to use when the spelling checker checks the spelling in messages. You can use the following values: • `Arabic` • `Catalan` • `Danish` • `Dutch` • `EnglishAustralia` • `EnglishCanada` • `EnglishUnitedKingdom` • `EnglishUnitedStates` • `Finnish` • `French` • `GermanPreReform`

			• GermanPostReform • Hebrew • Italian • Korean • NorwegianBokMal • NorwegianNyorsk • PortuguesePortugal • PortugueseBrasil • Spanish • Swedish
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreMixedDigits	Optional	System.Boolean	The *IgnoreMixedDigits* parameter specifies whether the spelling checker ignores words that contain numbers. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`.
IgnoreUppercase	Optional	System.Boolean	The *IgnoreUppercase* parameter specifies whether the spelling checker ignores words that contain only uppercase letters, for

			example, acronyms. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MapiVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MapiVirtualDirectory** cmdlet to view Message Application Programming Interface (MAPI) virtual directories on Microsoft Exchange 2013 servers. A MAPI virtual directory is used by supported versions of Microsoft Outlook to connect to mailboxes by using the MAPIHTTP protocol.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MapiVirtualDirectory [-Identity <VirtualDirectoryIdParameter>] <COMMON
PARAMETERS>

Get-MapiVirtualDirectory -Server <ServerIdParameter> <COMMON PARAMETERS>

COMMON PARAMETERS: [-ADPropertiesOnly <SwitchParameter>] [-
DomainController <Fqdn>] [-ShowBackEndVirtualDirectories
<SwitchParameter>] [-ShowMailboxVirtualDirectories <SwitchParameter>]
```

Examples

Example 1

This example returns the settings of the MAPI virtual directory on the server named ContosoMail.

```
Get-MapiVirtualDirectory -Server ContosoMail
```

Example 2

These examples return the settings of the MAPI virtual directory on the local server named ContosoMail. All three commands do the same thing.

```
Get-MapiVirtualDirectory -Identity "ContosoMail\mapi
(Default Web Site)"
Get-MapiVirtualDirectory "mapi (Default Web Site)"
Get-MapiVirtualDirectory mapi*
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description

Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server that will host the virtual directory. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the virtual directory will be created on the server where the remote PowerShell session is established. This will always be a Mailbox server. If you want to create the virtual directory on a Client Access server or another Mailbox server, you need to use the *Server* parameter.
ADPropertiesOnly	Optional	System.Management.Automation.SwitchParameter	The *ADPropertiesOnly* switch specifies whether to return only the properties about the virtual directory stored in Active Directory. The properties stored in the

			Internet Information Services (IIS) metabase aren't returned.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the MAPI virtual directory that you want to view. You can use any value that uniquely identifies the virtual directory. For example: • Name or <Server>\Name • Distinguished name (DN) • GUID The **Name** value uses the syntax "`<VirtualDirectoryName>(<WebsiteName>)`" from the properties of the virtual directory. You can specify the wildcard character (*) instead of the

			default website by using the syntax `<VirtualDirectoryName>*`.
ShowBackEndVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ShowMailboxVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MapiVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-MapiVirtualDirectory** cmdlet to create Messaging Application Programming Interface (MAPI) virtual directories on Microsoft Exchange 2013 servers A MAPI virtual directory is used by supported versions of Microsoft Outlook to connect to mailboxes by using the MAPIHTTP protocol.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MapiVirtualDirectory [-Confirm [<SwitchParameter>]] [-DomainController
```

```
<Fqdn>] [-ExtendedProtectionFlags <MultiValuedProperty>] [-
ExtendedProtectionSPNList <MultiValuedProperty>] [-
ExtendedProtectionTokenChecking <None | Allow | Require>] [-ExternalUrl
<Uri>] [-IISAuthenticationMethods <MultiValuedProperty>] [-InternalUrl
<Uri>] [-Role <ClientAccess | Mailbox>] [-Server <ServerIdParameter>] [-
WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example creates a new MAPI virtual directory that has the following configuration:

- **Internal URL** https://contoso.com/mapi
- **IIS authentication methods** NTLM and Negotiate.

```
New-MapiVirtualDirectory -InternalUrl https://contoso.com/
mapi -IISAuthenticationMethods NTLM,Negotiate
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	This parameter is reserved for internal Microsoft use.
ExternalUrl	Optional	System.Uri	The *ExternalURL* parameter specifies the URL that's used to connect to the virtual directory from outside the firewall. This setting enforces the Secure Sockets Layer (SSL) protocol and uses the default SSL port. Valid input for this parameter uses the syntax `https://<Domain Name>/mapi`.

			When you use the *InternalUrl* or *ExternalUrl* parameters, you need to specify one or more authentication values by using the *IISAuthenticationMethods* parameter.
IISAuthenticationMethods	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *IISAuthenticationMethods* parameter specifies the authentication methods that are enabled on the virtual directory in Internet Information Services (IIS). Valid values for this parameter are: • Basic • Negotiate • NTLM You can specify multiple values separated by commas.
InternalUrl	Optional	System.Uri	The *InternalURL* parameter specifies the URL that's used to connect to the virtual directory from inside the firewall. This setting enforces the Secure Sockets Layer (SSL) protocol and uses

			the default SSL port. Valid input for this parameter uses the syntax `https://<Domain Name>/mapi`. When you use the *InternalUrl* or *ExternalUrl* parameters, you need to specify one or more authentication values by using the *IISAuthenticationMethods* parameter.
Role	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.VirtualDirectoryRole	This parameter is reserved for internal Microsoft use.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server that will host the virtual directory. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the virtual directory will be created

			on the server where the remote PowerShell session is established. This will always be a Mailbox server. If you want to create the virtual directory on a Client Access server or another Mailbox server, you need to use the *Server* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MapiVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-MapiVirtualDirectory** cmdlet to remove Message Application Programming Interface (MAPI) virtual directories from Microsoft Exchange 2013 servers. A MAPI virtual directory is used by supported versions of Microsoft Outlook to connect to mailboxes by using the MAPIHTTP protocol.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MapiVirtualDirectory -Identity <VirtualDirectoryIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example removes the MAPI virtual directory from the local server named ContosoMail. All three commands do the same thing.

```
Remove-MapiVirtualDirectory -Identity "ContosoMail\mapi
(Default Web Site)"
Remove-MapiVirtualDirectory "mapi (Default Web Site)"
Remove-MapiVirtualDirectory mapi*
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the MAPI virtual directory that you want to remove. You can use any value that uniquely identifies the virtual directory. For example: • Name or <Server>\Name • Distinguished name (DN) • GUID The **Name** value uses the syntax "<VirtualDirectoryName> (<WebsiteName>)" from the properties of the virtual directory. You can specify the wildcard character (*) instead of the default website by using the syntax <VirtualDirectoryName>*.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax --

			`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MapiVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-MapiVirtualDirectory** cmdlet to modify Messaging Application Programming Interface (MAPI) virtual directories on Microsoft Exchange 2013 servers. A MAPI virtual directory is used by supported versions of Microsoft Outlook to connect to mailboxes by using the MAPIHTTP protocol.

```
Set-MapiVirtualDirectory -Identity <VirtualDirectoryIdParameter> -
IISAuthenticationMethods <MultiValuedProperty> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ExtendedProtectionFlags
<MultiValuedProperty>] [-ExtendedProtectionSPNList <MultiValuedProperty>]
[-ExtendedProtectionTokenChecking <None | Allow | Require>] [-ExternalUrl
<Uri>] [-InternalUrl <Uri>] [-Name <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example makes the following configuration changes to the MAPI virtual directory on the default web site of the server named ContosoMail:

- **Internal URL** https://contoso/mapi.
- **IIS authentication methods** NTLM and Negotiate.

```
Set-MapiVirtualDirectory -Identity "ContosoMail\mapi
(Default Web Site)" -InternalUrl https://contoso.com/mapi -
IISAuthenticationMethods NTLM,Negotiate
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the MAPI virtual directory that you want to configure. You can use any value that uniquely identifies the virtual directory. For example: • Name or <Server>\Name • Distinguished name (DN) • GUID The **Name** value uses the syntax "`<VirtualDirectoryName>(<WebsiteName>)`" from the properties of the virtual directory. You can specify the wildcard character (*) instead of the default website by using the syntax `<VirtualDirectoryName>*`.
IISAuthenticationMethods	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *IISAuthenticationMethods* parameter specifies the authentication methods that are enabled on the virtual directory in Internet Information Services (IIS). Valid values

			for this parameter are: • Basic • Negotiate • NTLM To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	This parameter is reserved for internal Microsoft use.
ExternalUrl	Optional	System.Uri	The *ExternalURL* parameter specifies the URL that's used to connect to the virtual directory from outside the firewall. This setting enforces the Secure Sockets Layer (SSL) protocol and uses the default SSL port. Valid

			input for this parameter uses the syntax `https://<Domain Name>/mapi`. When you use the *InternalUrl* or *ExternalUrl* parameters, you need to specify one or more authentication values by using the *IISAuthenticationMethods* parameter.
InternalUrl	Optional	System.Uri	The *InternalURL* parameter specifies the URL that's used to connect to the virtual directory from inside the firewall. This setting enforces the Secure Sockets Layer (SSL) protocol and uses the default SSL port. Valid input for this parameter uses the syntax `https://<Domain Name>/mapi`. When you use the *InternalUrl* or *ExternalUrl* parameters, you need to specify one or more authentication values by using the *IISAuthenticationMethods* parameter.

Name	Optional	System.String	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Clear-MobileDevice

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Clear-MobileDevice** cmdlet to delete all data from a mobile phone.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Clear-MobileDevice -Identity <MobileDeviceIdParameter> [-Cancel
<SwitchParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-NotificationEmailAddresses <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example clears all data from the mobile device with the Identity WM_JeffHay.

```
Clear-MobileDevice -Identity WM_JeffHay
```

EXAMPLE 2

This example clears all data from the mobile device for Tony Smith and sends a confirmation email message to tony@contoso.com.

```
Clear-MobileDevice -Identity WM_TonySmith -
NotificationEmailAddresses "tony@contoso.com"
```

EXAMPLE 3

This example cancels a previously sent **Clear-MobileDevice** command request for Tony Smith's mobile device.

```
Clear-MobileDevice -Identity WM_TonySmith -Cancel $true
```

Detailed Description

The **Clear-MobileDevice** cmdlet deletes all user data from a mobile device the next time that the device receives data from the server running Microsoft Exchange Server 2013. This cmdlet sets the *DeviceWipeStatus* parameter to `$true`. The mobile device acknowledges the cmdlet and records the time stamp in the *DeviceWipeAckTime* parameter.

After you run this cmdlet, you receive a warning that states: "This command will force all the data on the device to be permanently deleted. Do you want to continue?" You must respond to the warning for the cmdlet to run on the mobile phone.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange ActiveSync user settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MobileDeviceIdParameter	The *Identity* parameter specifies the identity of the device that you want to reset.
Cancel	Optional	System.Management.Automation.SwitchParameter	The *Cancel* parameter specifies whether the command should be canceled. If you use the *Cancel* parameter, a cancellation request is issued for the remote device wipe.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
NotificationEmailAddresses	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *NotificationEmailAddresses* parameter specifies the notification email address for the remote device wipe confirmation.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MobileDevice

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MobileDevice** cmdlet to get the list of devices in your organization that have active Microsoft Exchange ActiveSync partnerships.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MobileDevice [-Identity <MobileDeviceIdParameter>] <COMMON PARAMETERS>

Get-MobileDevice -Mailbox <MailboxIdParameter> <COMMON PARAMETERS>

COMMON PARAMETERS: [-ActiveSync <SwitchParameter>] [-DomainController
<Fqdn>] [-Filter <String>] [-Monitoring <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-OWAforDevices <SwitchParameter>] [-
ResultSize <Unlimited>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example returns all the Exchange ActiveSync mobile devices that Tony Smith has used that are associated with his mailbox.

```
Get-MobileDevice -Identity "TonySmith"
```

EXAMPLE 2

This example returns all the Exchange ActiveSync mobile devices that Tony Smith has used that are associated with his mailbox.

```
Get-MobileDevice -Mailbox "Redmond\TonySmith"
```

Detailed Description

The **Get-MobileDevice** cmdlet returns identification, configuration, and status information for each mobile device.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile devices user settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Mailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the user whose mobile devices you want to retrieve.
ActiveSync	Optional	System.Management.Automation.SwitchParameter	The *ActiveSync* switch specifies whether to include mobile devices that synchronize with Exchange ActiveSync.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter specifies a set of attributes

			used to filter the list of returned mobile devices.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MobileDeviceIdParameter	The *Identity* parameter specifies the device to retrieve. One of the following values is used to identify a mobile device in Active Directory: • `GUID` • `DeviceIdentity` • `Multi-TenantID`
Monitoring	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Monitoring* parameter specifies whether mobile devices that are created by monitoring accounts are exposed by the **Get-MobileDevice** cmdlet. The default value is *$false*.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies the organizational unit (OU) where the task is run.
OWAforDevices	Optional	System.Management.Automation.SwitchParameter	The *OWAforDevices* parameter specifies whether OWA for Devices is enabled for the specific

			mobile phone or device.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute to sort by.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MobileDevice

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MobileDevice** cmdlet to remove the mobile device partnership information that you specify from a user's mobile device list stored in a mailbox on a computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MobileDevice -Identity <MobileDeviceIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the mobile device partnership for the device WM_JeffHay.

```
Remove-MobileDevice -Identity WM_JeffHay
```

EXAMPLE 2

This example removes the mobile device partnership for the device iPhone_TonySmith after displaying the confirm prompt.

```
Remove-MobileDevice -Identity iPhone_TonySmith -Confirm
$true
```

EXAMPLE 3

This example removes the mobile device partnership for the device Tablet_JeffHay after displaying the confirm prompt.

```
Remove-MobileDevice -Identity Tablet_JeffHay -Confirm $true
```

Detailed Description

The **Remove-MobileDevice** cmdlet is useful for removing mobile devices that no longer synchronize successfully with the server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile Device user settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MobileDeviceIdParameter	The *Identity* parameter uniquely identifies the specific device partnership to be removed.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of

			those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MobileDeviceMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MobileDeviceMailboxPolicy** cmdlet to retrieve the Mobile Device mailbox policy settings for a specific Mobile Device mailbox policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MobileDeviceMailboxPolicy [-Identity <MailboxPolicyIdParameter>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example returns the policy settings for the Mobile Device mailbox policy SalesPolicy.

```
Get-MobileDeviceMailboxPolicy -Identity "SalesPolicy"
```

EXAMPLE 2

This example returns the policy settings for the Mobile Device mailbox policy Default.

```
Get-MobileDeviceMailboxPolicy -Identity "Default"
```

Detailed Description

A Mobile Device mailbox policy is a group of settings that specifies how mobile devices enabled for Exchange ActiveSync connect to the computer running Exchange. Exchange supports multiple Mobile Device mailbox policies. The **Get-MobileDeviceMailboxPolicy** cmdlet displays all the policy settings for the specified policy. These settings include password settings, file access settings, and attachment settings.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile Device mailbox policy settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the policy name.
Organization	Optional	Microsoft.Exchange.Co	The *Organization*

		nfiguration.Tasks.OrganizationIdParameter	parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MobileDeviceMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MobileDeviceMailboxPolicy** cmdlet to create Microsoft mobile device mailbox policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MobileDeviceMailboxPolicy -Name <String> [-AllowApplePushNotifications
<$true | $false>] [-AllowBluetooth <Disable | HandsfreeOnly | Allow>] [-
AllowBrowser <$true | $false>] [-AllowCamera <$true | $false>] [-
AllowConsumerEmail <$true | $false>] [-AllowDesktopSync <$true | $false>]
[-AllowExternalDeviceManagement <$true | $false>] [-
AllowGooglePushNotifications <$true | $false>] [-AllowHTMLEmail <$true |
$false>] [-AllowInternetSharing <$true | $false>] [-AllowIrDA <$true |
$false>] [-AllowMicrosoftPushNotifications <$true | $false>] [-
AllowMobileOTAUpdate <$true | $false>] [-AllowNonProvisionableDevices
<$true | $false>] [-AllowPOPIMAPEmail <$true | $false>] [-
AllowRemoteDesktop <$true | $false>] [-AllowSimplePassword <$true |
$false>] [-AllowSMIMEEncryptionAlgorithmNegotiation <BlockNegotiation |
OnlyStrongAlgorithmNegotiation | AllowAnyAlgorithmNegotiation>] [-
AllowSMIMESoftCerts <$true | $false>] [-AllowStorageCard <$true | $false>]
[-AllowTextMessaging <$true | $false>] [-AllowUnsignedApplications <$true
| $false>] [-AllowUnsignedInstallationPackages <$true | $false>] [-
AllowWiFi <$true | $false>] [-AlphanumericPasswordRequired <$true |
$false>] [-ApprovedApplicationList <ApprovedApplicationCollection>] [-
AttachmentsEnabled <$true | $false>] [-Confirm [<SwitchParameter>]] [-
DeviceEncryptionEnabled <$true | $false>] [-DevicePolicyRefreshInterval
<Unlimited>] [-DomainController <Fqdn>] [-IrmEnabled <$true | $false>] [-
```

```
IsDefault <$true | $false>] [-MaxAttachmentSize <Unlimited>] [-
MaxCalendarAgeFilter <All | TwoWeeks | OneMonth | ThreeMonths |
SixMonths>] [-MaxEmailAgeFilter <All | OneDay | ThreeDays | OneWeek |
TwoWeeks | OneMonth>] [-MaxEmailBodyTruncationSize <Unlimited>] [-
MaxEmailHTMLBodyTruncationSize <Unlimited>] [-MaxInactivityTimeLock
<Unlimited>] [-MaxPasswordFailedAttempts <Unlimited>] [-
MinPasswordComplexCharacters <Int32>] [-MinPasswordLength <Int32>] [-
MobileOTAUpdateMode <MajorVersionUpdates | MinorVersionUpdates |
BetaVersionUpdates>] [-Organization <OrganizationIdParameter>] [-
PasswordEnabled <$true | $false>] [-PasswordExpiration <Unlimited>] [-
PasswordHistory <Int32>] [-PasswordRecoveryEnabled <$true | $false>] [-
RequireDeviceEncryption <$true | $false>] [-RequireEncryptedSMIMEMessages
<$true | $false>] [-RequireEncryptionSMIMEAlgorithm <TripleDES | DES |
RC2128bit | RC264bit | RC240bit>] [-RequireManualSyncWhenRoaming <$true |
$false>] [-RequireSignedSMIMEAlgorithm <SHA1 | MD5>] [-
RequireSignedSMIMEMessages <$true | $false>] [-
RequireStorageCardEncryption <$true | $false>] [-
UnapprovedInROMApplicationList <MultiValuedProperty>] [-UNCAccessEnabled
<$true | $false>] [-WhatIf [<SwitchParameter>]] [-WSSAccessEnabled <$true
| $false>]
```

Examples

EXAMPLE 1

This example creates the mobile device mailbox policy Sales Policy that has several preconfigured values.

```
New-MobileDeviceMailboxPolicy -Name "Sales Policy" -
PasswordEnabled $true -AlphanumericPasswordRequired $true -
PasswordRecoveryEnabled $true -IsDefault $false -
AttachmentsEnabled $false -AllowStorageCard $true
```

EXAMPLE 2

This example creates the mobile device mailbox policy Management that has several preconfigured values. Users assigned to this policy should have an Enterprise client access license (CAL) to use many of these features.

```
New-MobileDeviceMailboxPolicy -Name Management -
AllowBluetooth $true -AllowBrowser $true -AllowCamera $true
-AllowPOPIMAPEmail $false -PasswordEnabled $true -
AlphanumericPasswordRequired $true -PasswordRecoveryEnabled
$true -MaxEmailAgeFilter TwoWeeks -AllowWiFi $true -
AllowStorageCard $true
```

EXAMPLE 3

This example creates the mobile device mailbox policy Contoso Policy that has several preconfigured values. This policy is configured to be the default policy for the organization. The default policy is assigned to all new users.

```
New-MobileDeviceMailboxPolicy -Name "Contoso Policy" -
PasswordEnabled $true -AlphanumericPasswordRequired $true -
PasswordRecoveryEnabled $true -MinPasswordComplexCharacters
3 -IsDefault $true -PasswordHistory 10
```

Detailed Description

Mobile device mailbox policies define settings for mobile devices that are used to access mailboxes in your organization. The default mobile device mailbox policy is applied to all new mailboxes that you create. You can assign a mobile device mailbox policy to existing mailboxes by using the **Set-CASMailbox** cmdlet, or by editing the mailbox properties in the Exchange admin center (EAC).

Note:

Some mobile device mailbox policy settings require the mobile device to have certain built-in features that enforce these security and device management settings. If your organization allows all devices, you need to set the *AllowNonProvisionableDevices* parameter to `$true`. This allows devices that can't enforce all policy settings to synchronize with your server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile device mailbox policy settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name of the mobile device mailbox policy. You can use any value that uniquely identifies the policy. For example: • Name • Distinguished name (DN) • GUID The name of the built-in

			mobile device mailbox policy is `Default`.
AllowApplePushNotifications	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *AllowApplePushNotifications* parameter specifies whether push notifications are allowed for Apple mobile devices. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowBluetooth	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.BluetoothType	The *AllowBluetooth* parameter specifies whether the Bluetooth capabilities of the mobile device are allowed. The available options are `Disable`, `HandsfreeOnly`, and `Allow`. The default value is `Allow`.
AllowBrowser	Optional	System.Boolean	The *AllowBrowser* parameter specifies whether Microsoft Pocket Internet Explorer is allowed on the mobile device. Valid input for this parameter is `$true` or

			`$false`. The default value is `$true`. This parameter doesn't affect third-party browsers.
AllowCamera	Optional	System.Boolean	The *AllowCamera* parameter specifies whether the mobile device's camera is allowed. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowConsumerEmail	Optional	System.Boolean	The *AllowConsumerEmail* parameter specifies whether the user can configure a personal email account on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. This parameter doesn't control access to email accounts using third-party mobile device email programs.
AllowDesktopSync	Optional	System.Boolean	The *AllowDesktopSync* parameter specifies whether the mobile device can synchronize with a desktop computer through a cable. Valid

			input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowExternalDeviceManagement	Optional	System.Boolean	The *AllowExternalDeviceManagement* parameter specifies whether an external device management program is allowed to manage the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
AllowGooglePushNotifications	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *AllowGooglePushNotifications* parameter controls whether the user can receive push notifications from Google for OWA for Devices. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowHTMLEmail	Optional	System.Boolean	The *AllowHTMLEmail* parameter specifies whether HTML-formatted email is enabled on the

			mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. If set to `$false`, all email is converted to plain text before synchronization occurs.
AllowInternetSharing	Optional	System.Boolean	The *AllowInternetSharing* parameter specifies whether the mobile device can be used as a modem to connect a computer to the Internet. This process is also known as *tethering*. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowIrDA	Optional	System.Boolean	The *AllowIrDA* parameter specifies whether infrared connections are allowed to the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowMicrosoftPushNotifications	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *AllowMicrosoftPushNotific*

			ations parameter specifies whether push notifications are enabled on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowMobileOTAUpdate	Optional	System.Boolean	The *AllowMobileOTAUpdate* parameter specifies whether the policy can be sent to the mobile device over a cellular data connection. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowNonProvisionableDevices	Optional	System.Boolean	The *AllowNonProvisionableDevices* parameter specifies whether all mobile devices can synchronize with Exchange. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. When set to `$true`, this parameter enables all mobile devices to synchronize with Exchange, regardless of

			whether the device can enforce all settings that are defined by the policy. This also includes mobile devices managed by a separate device management system. When set to `$false`, this parameter blocks mobile devices that aren't provisioned from synchronizing with Exchange.
AllowPOPIMAPEmail	Optional	System.Boolean	The *AllowPOPIMAPEmail* parameter specifies whether the user can configure a POP3 or IMAP4 email account on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. This parameter doesn't control access by third-party email programs.
AllowRemoteDesktop	Optional	System.Boolean	The *AllowRemoteDesktop* parameter specifies whether the mobile device can initiate a remote desktop connection. Valid input for this parameter is `$true` or `$false`. The

			default value is `$true`.
AllowSimplePassword	Optional	System.Boolean	The *AllowSimplePassword* parameter specifies whether a simple password is allowed on the mobile device. A simple password is a password that has a specific pattern, such as 1111 or 1234. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowSMIMEEncryptionAlgorithmNegotiation	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.SMIMEEncryptionAlgorithmNegotiationType	The *AllowSMIMEEncryptionAlgorithmNegotiation* parameter specifies whether the messaging application on the mobile device can negotiate the encryption algorithm if a recipient's certificate doesn't support the specified encryption algorithm. Valid values for this parameter are: • `AllowAnyAlgorithmNegotiation` • `BlockNegotiation` • `OnlyStrongAlgorithmNegotiation` The default value is `AllowAnyAlgorithmNegotiation`.

AllowSMIMESoftCerts	Optional	System.Boolean	The *AllowSMIMESoftCerts* parameter specifies whether S/MIME software certificates are allowed on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowStorageCard	Optional	System.Boolean	The *AllowStorageCard* parameter specifies whether the mobile device can access information stored on a storage card. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowTextMessaging	Optional	System.Boolean	The *AllowTextMessaging* parameter specifies whether text messaging is allowed from the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowUnsignedApplications	Optional	System.Boolean	The *AllowUnsignedApplications* parameter specifies whether unsigned applications can be installed on the mobile

			device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowUnsignedInstallationPackages	Optional	System.Boolean	The *AllowUnsignedInstallationPackages* parameter specifies whether unsigned installation packages are allowed to run on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowWiFi	Optional	System.Boolean	The *AllowWiFi* parameter specifies whether wireless Internet access is allowed on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AlphanumericPasswordRequired	Optional	System.Boolean	The *AlphanumericPasswordRequired* parameter specifies whether the password for the mobile device must be alphanumeric. Valid input for this parameter is

			`$true` or `$false`. The default value is `$false`.
ApprovedApplicationList	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ApprovedApplicationCollection	The *ApprovedApplicationList* parameter specifies a configured list of approved applications for the device.
AttachmentsEnabled	Optional	System.Boolean	The *AttachmentsEnabled* parameter specifies whether attachments can be downloaded on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. When set to `$false`, this parameter blocks the user from downloading attachments on the mobile device.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DeviceEncryptionEnabled	Optional	System.Boolean	The *DeviceEncryptionEnabled* parameter specifies whether encryption is enabled on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When this parameter is set to `$true`, device encryption is enabled on the mobile device.
DevicePolicyRefreshInterval	Optional	Microsoft.Exchange.Data.Unlimited	The *DevicePolicyRefreshInterval* parameter specifies how often the policy is sent to the mobile device. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The default value is `Unlimited`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IrmEnabled	Optional	System.Boolean	The *IrmEnabled* parameter specifies whether Information Rights Management (IRM) is enabled for the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
IsDefault	Optional	System.Boolean	The *IsDefault* parameter specifies whether this policy is the default mobile device mailbox policy. Valid input for this parameter is `$true` or `$false`. The default value for the built-in mobile device mailbox policy named `Default` is `$true`. The default value for new mobile device mailbox policies that you create is `$false`. There can be only one default policy. If another policy is currently set as the default, and you set

			this parameter to `$true`, this policy becomes the default policy. The value of this parameter on the other policy is automatically changed to `$false`, and that policy is no longer the default policy.
MaxAttachmentSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxAttachmentSize* parameter specifies the maximum size of attachments that can be downloaded to the mobile device. Valid input for this parameter is a size value between 0 and 2147482624 bytes (approximately 2 GB), or the value `Unlimited`. The default value is `Unlimited`. Unqualified values are treated as bytes. You can qualify the value with `KB` (kilobytes), `MB` (megabytes) or `GB` (gigabytes). For example, to set the limit to 4 kilobytes, enter the value `4096` or `4KB`. The maximum value is 1024 bytes (one kilobyte)

			less than two gigabytes (2*1024^3), so these are the maximum qualified values you can use with this parameter. • `2097151KB` • `2047.999024MB` • `1.999999047GB`
MaxCalendarAgeFilter	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.CalendarAgeFilterType	The *MaxCalendarAgeFilter* parameter specifies the maximum range of calendar days that can be synchronized to the mobile device. Valid values for this parameter are: • `All` • `TwoWeeks` • `OneMonth` • `ThreeMonths` • `SixMonths` The default value is `All`.
MaxEmailAgeFilter	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.EmailAgeFilterType	The *MaxEmailAgeFilter* parameter specifies the maximum number of days of email items to synchronize to the mobile device. Valid values for this parameter are: • `All` • `OneDay` • `ThreeDays` • `OneWeek` • `TwoWeeks` • `OneMonth` The default value is `All`.

MaxEmailBodyTruncationSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxEmailBodyTruncationSize* parameter specifies the maximum size at which email messages are truncated when synchronized to the mobile device. Valid input for this parameter is an integer between 0 and 2147483647 (`Int32`) or the value `Unlimited`. The default value is `Unlimited`. Unqualified values are treated as bytes. You can qualify the value with `KB` (kilobytes), `MB` (megabytes) or `GB` (gigabytes). For example, to set the limit to 4 kilobytes, enter the value `4KB` or `4096`.
MaxEmailHTMLBodyTruncationSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxEmailHTMLBodyTruncationSize* parameter specifies the maximum size at which HTML-formatted email messages are truncated when synchronized to the mobile device. Valid input for this parameter is an

			integer between 0 and 2147483647 (`Int32`) or the value `Unlimited`. The default value is `Unlimited`. Unqualified values are treated as bytes. You can qualify the value with `KB` (kilobytes), `MB` (megabytes) or `GB` (gigabytes). For example, to set the limit to 4 kilobytes, enter the value `4KB` or `4096`.
MaxInactivityTimeLock	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxInactivityTimeLock* parameter specifies the length of time that the mobile device can be inactive before the password is required to reactivate it. This parameter accepts the following values: • Timespan *hh:mm:ss*, where *hh* = hours, *mm* = minutes and *ss*= seconds. The valid input range is `00:01:00` to `01:00:00` (one minute to one hour). • The value `Unlimited`. The default value is

			`Unlimited`.
MaxPasswordFailedAttempts	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxPasswordFailedAttempts* parameter specifies the number of attempts a user can make to enter the correct password for the mobile device. You can enter any number from 4 through 16 or the value `Unlimited`. The default value is `Unlimited`.
MinPasswordComplexCharacters	Optional	System.Int32	The *MinPasswordComplexCharacters* parameter specifies the minimum number of complex characters required in a mobile device password. A complex character isn't a letter. You can enter any number from 1 through 4. The default value is 1.
MinPasswordLength	Optional	System.Int32	The *MinPasswordLength* parameter specifies the minimum number of characters in the mobile device password. You can enter any number from 1 through 16 or the

			value `$null`. The default value is blank. The maximum password length is 16 characters.
MobileOTAUpdateMode	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MobileOTAUpdateModeType	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
PasswordEnabled	Optional	System.Boolean	The *PasswordEnabled* parameter specifies whether a password is required on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When set to `$true`, this parameter requires the user to set a password on the mobile device.
PasswordExpiration	Optional	Microsoft.Exchange.Data.Unlimited	The *PasswordExpiration* parameter specifies how long a password can be used on a mobile device before the user is forced to change the password. This parameter accepts the following values:

			• Timespan *ddd.hh:mm:ss*, where *ddd* = days, *hh* = hours, *mm* = minutes and *ss*= seconds. The valid input range is `1.00:00:00` to `730.00:00:00` (one day to two years). • The value `Unlimited`. The default value is `Unlimited`.
PasswordHistory	Optional	System.Int32	The *PasswordHistory* parameter specifies the number of unique new passwords that need to be created on the mobile device before an old password can be reused. You can enter any number from 0 through 50. The default value is 0.
PasswordRecoveryEnabled	Optional	System.Boolean	The *PasswordRecoveryEnabled* parameter specifies whether the recovery password for the mobile device is stored in Exchange. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.

			When set to `$true`, this parameter enables you to store the recovery password for the mobile device in Exchange. The recovery password can be viewed from Microsoft Outlook Web App or the Exchange admin center.
RequireDeviceEncryption	Optional	System.Boolean	The *RequireDeviceEncryption* parameter specifies whether encryption is required on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
RequireEncryptedSMIMEMessages	Optional	System.Boolean	The *RequireEncryptedSMIMEMessages* parameter specifies whether the mobile device must send encrypted S/MIME messages. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
RequireEncryptionSMIMEAlgorithm	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.Encryption	The *RequireEncryptionSMIMEAlgorithm* parameter specifies the algorithm

		SMIMEAlgorithmType	that's required to encrypt S/MIME messages on a mobile device. The valid values for this parameter are: • `DES` • `TripleDES` • `RC240bit` • `RC264bit` • `RC2128bit` The default value is `TripleDES`.
RequireManualSyncWhenRoaming	Optional	System.Boolean	The *RequireManualSyncWhenRoaming* parameter specifies whether the mobile device must synchronize manually while roaming. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
RequireSignedSMIMEAlgorithm	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.SignedSMIMEAlgorithmType	The *RequireSignedSMIMEAlgorithm* parameter specifies the algorithm that's used to sign S/MIME messages on the mobile device. Valid values for this parameter are `SHA1` or `MD5`. The default value is `SHA1`.
RequireSignedSMIME	Optional	System.Boolean	The

Messages			*RequireSignedSMIMEMessages* parameter specifies whether the mobile device must send signed S/MIME messages. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
RequireStorageCardEncryption	Optional	System.Boolean	The *RequireStorageCardEncryption* parameter specifies whether storage card encryption is required on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. Setting this parameter to `$true` also sets the *DeviceEncryptionEnabled* parameter to `$true`.
UnapprovedInROMApplicationList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UnapprovedInROMApplicationList* parameter specifies a list of applications that can't be run in ROM on the mobile device.
UNCAccessEnabled	Optional	System.Boolean	The *UNCAccessEnabled* parameter specifies whether access to

			Microsoft Windows file shares is enabled from the mobile device. In on-premises Exchange 2013 organizations, access to specific shares is configured on the Exchange ActiveSync virtual directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WSSAccessEnabled	Optional	System.Boolean	The *WSSAccessEnabled* parameter specifies whether access to Microsoft Windows SharePoint Services is enabled from the mobile device. In on-premises Exchange 2013 organizations, access to specific shares is configured on the

			Exchange ActiveSync virtual directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MobileDeviceMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MobileDeviceMailboxPolicy** cmdlet to remove a specific Microsoft Mobile Device mailbox policy from a computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MobileDeviceMailboxPolicy -Identity <MailboxPolicyIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Force
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the Mobile Device mailbox policy SalesPolicy.

```
Remove-MobileDeviceMailboxPolicy -Identity "SalesPolicy"
```

EXAMPLE 2

This example removes the Mobile Device mailbox policy Default after confirmation is given.

```
Remove-MobileDeviceMailboxPolicy -Identity "Default" -
Confirm $true
```

EXAMPLE 3

This example removes the Mobile Device mailbox policy Management and bypasses any confirmation prompts.

```
Remove-MobileDeviceMailboxPolicy -Identity "Management" -
Force $true
```

Detailed Description

A Mobile Device mailbox policy is a group of settings that specifies how mobile phones connect to Exchange. Exchange supports multiple Mobile Device mailbox policies. The **Remove-MobileDeviceMailboxPolicy** cmdlet removes a specific Mobile Device mailbox policy. If any users are assigned to the policy when you remove it, the **Remove-MobileDeviceMailboxPolicy** cmdlet fails.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile Device mailbox policy settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the policy name.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't

			have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies that the command should run immediately and bypass confirmation prompts.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MobileDeviceMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MobileDeviceMailboxPolicy** cmdlet to modify mobile device mailbox policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MobileDeviceMailboxPolicy -Identity <MailboxPolicyIdParameter> [-
AllowApplePushNotifications <$true | $false>] [-AllowBluetooth <Disable |
HandsfreeOnly | Allow>] [-AllowBrowser <$true | $false>] [-AllowCamera
<$true | $false>] [-AllowConsumerEmail <$true | $false>] [-
AllowDesktopSync <$true | $false>] [-AllowExternalDeviceManagement <$true
| $false>] [-AllowGooglePushNotifications <$true | $false>] [-
AllowHTMLEmail <$true | $false>] [-AllowInternetSharing <$true | $false>]
[-AllowIrDA <$true | $false>] [-AllowMicrosoftPushNotifications <$true |
$false>] [-AllowMobileOTAUpdate <$true | $false>] [-
AllowNonProvisionableDevices <$true | $false>] [-AllowPOPIMAPEmail <$true
| $false>] [-AllowRemoteDesktop <$true | $false>] [-AllowSimplePassword
<$true | $false>] [-AllowSMIMEEncryptionAlgorithmNegotiation
<BlockNegotiation | OnlyStrongAlgorithmNegotiation |
AllowAnyAlgorithmNegotiation>] [-AllowSMIMESoftCerts <$true | $false>] [-
AllowStorageCard <$true | $false>] [-AllowTextMessaging <$true | $false>]
[-AllowUnsignedApplications <$true | $false>] [-
AllowUnsignedInstallationPackages <$true | $false>] [-AllowWiFi <$true |
$false>] [-AlphanumericPasswordRequired <$true | $false>] [-
ApprovedApplicationList <ApprovedApplicationCollection>] [-
AttachmentsEnabled <$true | $false>] [-Confirm [<SwitchParameter>]] [-
DeviceEncryptionEnabled <$true | $false>] [-DevicePolicyRefreshInterval
<Unlimited>] [-DomainController <Fqdn>] [-IrmEnabled <$true | $false>] [-
IsDefault <$true | $false>] [-MaxAttachmentSize <Unlimited>] [-
```

```
MaxCalendarAgeFilter <All | TwoWeeks | OneMonth | ThreeMonths |
SixMonths>] [-MaxEmailAgeFilter <All | OneDay | ThreeDays | OneWeek |
TwoWeeks | OneMonth>] [-MaxEmailBodyTruncationSize <Unlimited>] [-
MaxEmailHTMLBodyTruncationSize <Unlimited>] [-MaxInactivityTimeLock
<Unlimited>] [-MaxPasswordFailedAttempts <Unlimited>] [-
MinPasswordComplexCharacters <Int32>] [-MinPasswordLength <Int32>] [-
MobileOTAUpdateMode <MajorVersionUpdates | MinorVersionUpdates |
BetaVersionUpdates>] [-Name <String>] [-PasswordEnabled <$true | $false>]
[-PasswordExpiration <Unlimited>] [-PasswordHistory <Int32>] [-
PasswordRecoveryEnabled <$true | $false>] [-RequireDeviceEncryption <$true
| $false>] [-RequireEncryptedSMIMEMessages <$true | $false>] [-
RequireEncryptionSMIMEAlgorithm <TripleDES | DES | RC2128bit | RC264bit |
RC240bit>] [-RequireManualSyncWhenRoaming <$true | $false>] [-
RequireSignedSMIMEAlgorithm <SHA1 | MD5>] [-RequireSignedSMIMEMessages
<$true | $false>] [-RequireStorageCardEncryption <$true | $false>] [-
UnapprovedInROMApplicationList <MultiValuedProperty>] [-UNCAccessEnabled
<$true | $false>] [-WhatIf [<SwitchParameter>]] [-WSSAccessEnabled <$true
| $false>]
```

Examples

EXAMPLE 1

This example sets several policy settings for the mobile device mailbox policy Sales Policy.

```
Set-MobileDeviceMailboxPolicy -Identity "Sales Policy" -
PasswordEnabled $true -AlphanumericPasswordRequired $true -
PasswordRecoveryEnabled $true -AttachmentsEnabled $true -
MaxInactivityTimeLock 15:00 -IsDefault $false
```

EXAMPLE 2

This example sets several policy settings for the mobile device mailbox policy Management.

```
Set-MobileDeviceMailboxPolicy -Identity Management -
PasswordEnabled $true -AlphanumericPasswordRequired $true -
PasswordRecoveryEnabled $true -AllowCamera $true -AllowWiFi
$false -AllowStorageCard $true -AllowPOPIMAPEmail $false
```

EXAMPLE 3

This example sets several policy settings for the mobile device mailbox policy Default and requires confirmation before applying the settings.

```
Set-MobileDeviceMailboxPolicy -Identity Default -
PasswordEnabled $true -AlphanumericPasswordRequired $true -
PasswordRecoveryEnabled $true -AllowWiFi $false -
AllowStorageCard $true -AllowPOPIMAPEmail $false -IsDefault
$true -AllowTextMessaging $true -Confirm $true
```

Detailed Description

Mobile device mailbox policies define settings for mobile devices that are used to access mailboxes in your organization. The default mobile device mailbox policy is applied to all new mailboxes that you create. You can assign a mobile device mailbox policy to existing mailboxes by using the **Set-CASMailbox** cmdlet, or by editing the mailbox properties in the Exchange admin center (EAC).

Note:

Some mobile device mailbox policy settings require the mobile device to have specific built-in features that enforce these security and device management settings. If your organization allows all devices, you must set the *AllowNonProvisionableDevices* parameter to `$true`. This applies to devices that can't enforce all policy settings.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile device mailbox policy settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the mobile device mailbox policy. You can use any value that uniquely identifies the policy. For example: • Name • Distinguished name (DN) • GUID The name of the built-in mobile device mailbox policy is `Default`.
AllowApplePushNotifications	Optional	System.Boolean	This parameter is available only in the

			cloud-based service. The *AllowApplePushNotifications* parameter specifies whether push notifications are allowed for Apple mobile devices. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowBluetooth	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.BluetoothType	The *AllowBluetooth* parameter specifies whether the Bluetooth capabilities are allowed on the mobile device. The available options are `Disable`, `HandsfreeOnly`, and `Allow`. The default value is `Allow`.
AllowBrowser	Optional	System.Boolean	The *AllowBrowser* parameter specifies whether Microsoft Pocket Internet Explorer is allowed on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. This parameter doesn't affect third-party browsers.

AllowCamera	Optional	System.Boolean	The *AllowCamera* parameter specifies whether the mobile device's camera is allowed. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowConsumerEmail	Optional	System.Boolean	The *AllowConsumerEmail* parameter specifies whether the user can configure a personal email account on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. This parameter doesn't control access to email accounts using third-party mobile device email programs.
AllowDesktopSync	Optional	System.Boolean	The *AllowDesktopSync* parameter specifies whether the mobile device can synchronize with a desktop computer through a cable. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowExternalDeviceM	Optional	System.Boolean	The

anagement			*AllowExternalDeviceManagement* parameter specifies whether an external device management program is allowed to manage the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
AllowGooglePushNotifications	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *AllowGooglePushNotifications* parameter controls whether the user can receive push notifications from Google for OWA for Devices. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowHTMLEmail	Optional	System.Boolean	The *AllowHTMLEmail* parameter specifies whether HTML-formatted email is enabled on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. If set to `$false`, all email is

			converted to plain text before synchronization occurs.
AllowInternetSharing	Optional	System.Boolean	The *AllowInternetSharing* parameter specifies whether the mobile device can be used as a modem to connect a computer to the Internet. This process is also known as *tethering*. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowIrDA	Optional	System.Boolean	The *AllowIrDA* parameter specifies whether infrared connections are allowed to the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowMicrosoftPushNotifications	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *AllowMicrosoftPushNotifications* parameter specifies whether push notifications are enabled on the mobile device. Valid input for this

			parameter is `$true` or `$false`. The default value is `$true`.
AllowMobileOTAUpdate	Optional	System.Boolean	The *AllowMobileOTAUpdate* parameter specifies whether the policy can be sent to the mobile device over a cellular data connection. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowNonProvisionableDevices	Optional	System.Boolean	The *AllowNonProvisionableDevices* parameter specifies whether all mobile devices can synchronize with Exchange. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. When set to `$true`, this parameter enables all mobile devices to synchronize with Exchange, regardless of whether the device can enforce all settings that are defined by the policy. This also includes mobile devices managed by a

			separate device management system. When set to `$false`, this parameter blocks mobile devices that aren't provisioned from synchronizing with Exchange.
AllowPOPIMAPEmail	Optional	System.Boolean	The *AllowPOPIMAPEmail* parameter specifies whether the user can configure a POP3 or IMAP4 email account on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. This parameter doesn't control access by third-party email programs.
AllowRemoteDesktop	Optional	System.Boolean	The *AllowRemoteDesktop* parameter specifies whether the mobile device can initiate a remote desktop connection. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowSimplePassword	Optional	System.Boolean	The *AllowSimplePassword* parameter specifies whether a simple

			password is allowed on the mobile device. A simple password is a password that has a specific pattern, such as 1111 or 1234. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowSMIMEEncryptionAlgorithmNegotiation	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.SMIMEEncryptionAlgorithmNegotiationType	The *AllowSMIMEEncryptionAlgorithmNegotiation* parameter specifies whether the messaging application on the mobile device can negotiate the encryption algorithm if a recipient's certificate doesn't support the specified encryption algorithm. Valid values for this parameter are: • `AllowAnyAlgorithmNegotiation` • `BlockNegotiation` • `OnlyStrongAlgorithmNegotiation` The default value is `AllowAnyAlgorithmNegotiation`.
AllowSMIMESoftCerts	Optional	System.Boolean	The *AllowSMIMESoftCerts* parameter specifies whether S/MIME software certificates are allowed on

			the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowStorageCard	Optional	System.Boolean	The *AllowStorageCard* parameter specifies whether the mobile device can access information stored on a storage card. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowTextMessaging	Optional	System.Boolean	The *AllowTextMessaging* parameter specifies whether text messaging is allowed from the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowUnsignedApplications	Optional	System.Boolean	The *AllowUnsignedApplications* parameter specifies whether unsigned applications can be installed on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.

AllowUnsignedInstallationPackages	Optional	System.Boolean	The *AllowUnsignedInstallationPackages* parameter specifies whether unsigned installation packages can be executed on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowWiFi	Optional	System.Boolean	The *AllowWiFi* parameter specifies whether wireless Internet access is allowed on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AlphanumericPasswordRequired	Optional	System.Boolean	The *AlphanumericPasswordRequired* parameter specifies whether the password for the mobile device must be alphanumeric. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
ApprovedApplicationList	Optional	Microsoft.Exchange.Data.Directory.SystemCo	The *ApprovedApplicationList*

		nfiguration.Approved ApplicationCollection	parameter specifies a configured list of approved applications for the device.
AttachmentsEnabled	Optional	System.Boolean	The *AttachmentsEnabled* parameter specifies whether attachments can be downloaded on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. When set to `$false`, this parameter blocks the user from downloading attachments on the mobile device.
Confirm	Optional	System.Management. Automation.SwitchPar ameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DeviceEncryptionEnab led	Optional	System.Boolean	The *DeviceEncryptionEnabled* parameter specifies whether encryption is

			enabled on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When this parameter is set to `$true`, device encryption is enabled on the mobile device.
DevicePolicyRefreshInterval	Optional	Microsoft.Exchange.Data.Unlimited	The *DevicePolicyRefreshInterval* parameter specifies how often the policy is sent to the mobile device. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The default value is `Unlimited`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory.
IrmEnabled	Optional	System.Boolean	The *IrmEnabled* parameter specifies whether Information Rights Management (IRM) is enabled for the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
IsDefault	Optional	System.Boolean	The *IsDefault* parameter specifies whether this policy is the default mobile device mailbox policy. Valid input for this parameter is `$true` or `$false`. The default value for the built-in mobile device mailbox policy named `Default` is `$true`. The default value for new mobile device mailbox policies that you create is `$false`. There can be only one default policy. If another policy is currently set as the default, and you set this parameter to `$true`, this policy becomes the default policy. The value of this parameter on the

			other policy is automatically changed to `$false`, and that policy is no longer the default policy.
MaxAttachmentSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxAttachmentSize* parameter specifies the maximum size of attachments that can be downloaded to the mobile device. Valid input for this parameter is a size value between 0 and 2147482624 bytes (approximately 2 GB), or the value `Unlimited`. The default value is `Unlimited`. Unqualified values are treated as bytes. You can qualify the value with `KB` (kilobytes), `MB` (megabytes) or `GB` (gigabytes). For example, to set the limit to 4 kilobytes, enter the value `4096` or `4KB`. The maximum value is 1024 bytes (one kilobyte) less than two gigabytes (2*1024^3), so these are the maximum qualified values you can use with

			this parameter. • 2097151KB • 2047.999024MB • 1.999999047GB
MaxCalendarAgeFilter	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.CalendarAgeFilterType	The *MaxCalendarAgeFilter* parameter specifies the maximum range of calendar days that can be synchronized to the mobile device. Valid values for this parameter are: • All • TwoWeeks • OneMonth • ThreeMonths • SixMonths The default value is All.
MaxEmailAgeFilter	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.EmailAgeFilterType	The *MaxEmailAgeFilter* parameter specifies the maximum number of days of email items to synchronize to the mobile device. Valid values for this parameter are: • All • OneDay • ThreeDays • OneWeek • TwoWeeks • OneMonth The default value is All.
MaxEmailBodyTruncationSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxEmailBodyTruncationSize* parameter specifies the maximum size at

			which email messages are truncated when synchronized to the mobile device. Valid input for this parameter is an integer between 0 and 2147483647 (`Int32`) or the value `Unlimited`. The default value is `Unlimited`. Unqualified values are treated as bytes. You can qualify the value with `KB` (kilobytes), `MB` (megabytes) or `GB` (gigabytes). For example, to set the limit to 4 kilobytes, enter the value `4KB` or `4096`.
MaxEmailHTMLBodyTruncationSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxEmailHTMLBodyTruncationSize* parameter specifies the maximum size at which HTML-formatted email messages are truncated when synchronized to the mobile device. Valid input for this parameter is an integer between 0 and 2147483647 (`Int32`) or the value `Unlimited`. The default value is `Unlimited`.

			Unqualified values are treated as bytes. You can qualify the value with KB (kilobytes), MB (megabytes) or GB (gigabytes). For example, to set the limit to 4 kilobytes, enter the value 4KB or 4096.
MaxInactivityTimeLock	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxInactivityTimeLock* parameter specifies the length of time that the mobile device can be inactive before the password is required to reactivate it. This parameter accepts the following values: • Timespan *hh:mm:ss*, where *hh* = hours, *mm* = minutes and *ss*= seconds. The valid input range is 00:01:00 to 01:00:00 (one minute to one hour). • The value Unlimited. The default value is Unlimited.
MaxPasswordFailedAttempts	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxPasswordFailedAttempts* parameter specifies

			the number of attempts a user can make to enter the correct password for the mobile device. You can enter any number from 4 through 16 or the value `Unlimited`. The default value is `Unlimited`.
MinPasswordComplexCharacters	Optional	System.Int32	The *MinPasswordComplexCharacters* parameter specifies the minimum number of complex characters required in a mobile device password. A complex character isn't a letter. You can enter any number from 1 through 4. The default value is 1.
MinPasswordLength	Optional	System.Int32	The *MinPasswordLength* parameter specifies the minimum number of characters in the mobile device password. You can enter any number from 1 through 16 or the value `$null`. The default value is blank. The maximum password length is 16 characters.

MobileOTAUpdateMode	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MobileOTAUpdateModeType	This parameter is reserved for internal Microsoft use.
Name	Optional	System.String	The *Name* parameter specifies the friendly name of the mobile device mailbox policy.
PasswordEnabled	Optional	System.Boolean	The *PasswordEnabled* parameter specifies whether a password is required on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When set to `$true`, this parameter requires the user to set a password on the mobile device.
PasswordExpiration	Optional	Microsoft.Exchange.Data.Unlimited	The *PasswordExpiration* parameter specifies how long a password can be used on a mobile device before the user is forced to change the password. This parameter accepts the following values: • Timespan *ddd.hh:mm:ss*, where *ddd* = days, *hh* = hours,

			mm = minutes and *ss*= seconds. The valid input range is `1.00:00:00` to `730.00:00:00` (one day to two years). • The value `Unlimited`. The default value is `Unlimited`.
PasswordHistory	Optional	System.Int32	The *PasswordHistory* parameter specifies the number of unique new passwords that need to be created on the mobile device before an old password can be reused. You can enter any number from 0 through 50. The default value is 0.
PasswordRecoveryEnabled	Optional	System.Boolean	The *PasswordRecoveryEnabled* parameter specifies whether the recovery password for the mobile device is stored in Exchange. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When set to `$true`, this parameter enables you to store the recovery

			password for the mobile device in Exchange. The recovery password can be viewed from Microsoft Outlook Web App or the Exchange admin center.
RequireDeviceEncryption	Optional	System.Boolean	The *RequireDeviceEncryption* parameter specifies whether encryption is required on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
RequireEncryptedSMIMEMessages	Optional	System.Boolean	The *RequireEncryptedSMIMEMessages* parameter specifies whether the mobile device must send encrypted S/MIME messages. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
RequireEncryptionSMIMEAlgorithm	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.EncryptionSMIMEAlgorithmType	The *RequireEncryptionSMIMEAlgorithm* parameter specifies the algorithm that's required to encrypt S/MIME messages on a mobile device. The valid

			values for this parameter are: • DES • TripleDES • RC240bit • RC264bit • RC2128bit The default value is TripleDES.
RequireManualSyncWhenRoaming	Optional	System.Boolean	The *RequireManualSyncWhenRoaming* parameter specifies whether the mobile device must synchronize manually while roaming. Valid input for this parameter is $true or $false. The default value is $false.
RequireSignedSMIMEAlgorithm	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.SignedSMIMEAlgorithmType	The *RequireSignedSMIMEAlgorithm* parameter specifies the algorithm that's used to sign S/MIME messages on the mobile device. Valid values for this parameter are SHA1 or MD5. The default value is SHA1.
RequireSignedSMIMEMessages	Optional	System.Boolean	The *RequireSignedSMIMEMessages* parameter specifies whether the mobile device

			must send signed S/MIME messages. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
RequireStorageCardEncryption	Optional	System.Boolean	The *RequireStorageCardEncryption* parameter specifies whether storage card encryption is required on the mobile device. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. Setting this parameter to `$true` also sets the *DeviceEncryptionEnabled* parameter to `$true`.
UnapprovedInROMApplicationList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UnapprovedInROMApplicationList* parameter specifies a list of applications that can't be run in ROM on the mobile device.
UNCAccessEnabled	Optional	System.Boolean	The *UNCAccessEnabled* parameter specifies whether access to Microsoft Windows file shares is enabled from the mobile device. In on-

			premises Exchange 2013 organizations, access to specific shares is configured on the Exchange ActiveSync virtual directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WSSAccessEnabled	Optional	System.Boolean	The *WSSAccessEnabled* parameter specifies whether access to Microsoft Windows SharePoint Services is enabled from the mobile device. In on-premises Exchange 2013 organizations, access to specific shares is configured on the Exchange ActiveSync virtual directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MobileDeviceStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MobileDeviceStatistics** cmdlet to retrieve the list of mobile devices configured to synchronize with a specified user's mailbox and return a list of statistics about the mobile devices.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MobileDeviceStatistics -Identity <MobileDeviceIdParameter> <COMMON
PARAMETERS>


Get-MobileDeviceStatistics -Mailbox <MailboxIdParameter> <COMMON
PARAMETERS>


COMMON PARAMETERS: [-ActiveSync <SwitchParameter>] [-DomainController
<Fqdn>] [-GetMailboxLog <SwitchParameter>] [-NotificationEmailAddresses
<MultiValuedProperty>] [-OWAforDevices <SwitchParameter>] [-
ShowRecoveryPassword <SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves the statistics for the mobile phone configured to synchronize with the mailbox that belongs to the user Tony Smith.

```
Get-MobileDeviceStatistics -Identity TonySmith
```

EXAMPLE 2

This example uses the **Get-CASMailbox** cmdlet to determine who in the organization has a Microsoft Exchange ActiveSync mobile device. For each mobile device, the Exchange ActiveSync device statistics are retrieved.

```
$UserList = Get-CASMailbox -Filter
{hasactivesyncdevicepartnership -eq $true -and -not
displayname -like "CAS_{*"} | Get-Mailbox
$UserList | foreach { Get-MobileDeviceStatistics -Mailbox
$_}
```

EXAMPLE 3

This example retrieves the statistics for the mobile phone configured to synchronize with the mailbox that belongs to the user Tony Smith. It also outputs the Exchange ActiveSync log file and sends it to the System Administrator at admin@contoso.com.

```
Get-MobileDeviceStatistics -Mailbox TonySmith -
GetMailboxLog $true -NotificationEmailAddresses
"admin@contoso.com"
```

Detailed Description

The **Get-MobileDeviceStatistics** cmdlet returns a list of statistics about each mobile device. Additionally, it allows you to retrieve logs and send those logs to a recipient for troubleshooting purposes.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mobile Device user settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MobileDeviceIdParameter	The *Identity* parameter specifies the user's device ID. If the *Mailbox* parameter is specified, the *Identity* parameter

			is disabled.
Mailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the user mailbox for which you want to retrieve the mobile phone statistics.
ActiveSync	Optional	System.Management.Automation.SwitchParameter	The *ActiveSync* switch specifies whether to return statistics for Microsoft Exchange ActiveSync or other mobile device synchronization.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
GetMailboxLog	Optional	System.Management.Automation.SwitchParameter	The *GetMailboxLog* parameter specifies whether to send the mailbox logs via email to the administrator running the task. If the parameter is set to

			`$true`, the command sends the mailbox logs via email to the administrator running the task. The default value of this parameter is `$false`.
NotificationEmailAddresses	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *NotificationEmailAddresses* parameter specifies an optional list of comma-separated aliases or email addresses where the mailbox logs are sent. If the *GetMailboxLog* parameter is set to `$false`, this parameter is ignored.
OWAforDevices	Optional	System.Management.Automation.SwitchParameter	The *OWAforDevices* parameter specifies whether Outlook Web App for Mobile Devices is enabled for the mobile device.
ShowRecoveryPassword	Optional	System.Management.Automation.SwitchParameter	The *ShowRecoveryPassword* parameter specifies whether to return the recovery password for

			the mobile phone as one of the displayed statistics. If this parameter is set to `$true`, the command returns the recovery password for the mobile phone as one of the displayed statistics.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-OutlookAnywhere

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-OutlookAnywhere** cmdlet to retrieve all Outlook Anywhere settings on a computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-OutlookAnywhere -Server <ServerIdParameter> <COMMON PARAMETERS>

Get-OutlookAnywhere [-Identity <VirtualDirectoryIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-ADPropertiesOnly <SwitchParameter>] [-
DomainController <Fqdn>] [-ShowMailboxVirtualDirectories
```

```
<SwitchParameter>]
```

Examples

EXAMPLE 1

This example uses the *Server* and *Identity* parameters to retrieve all Outlook Anywhere settings on the server CAS01.

```
Get-OutlookAnywhere -Server CAS01
```

EXAMPLE 2

This example displays all Outlook Anywhere settings for the server EXCH01.

```
Get-OutlookAnywhere -Identity "EXCH01\rpc (Default Web
Site)"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Anywhere configuration (enable, disable, change, view)" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name or GUID of the server for which you want to display the settings. This parameter is required if you aren't running the **Get-OutlookAnywhere** cmdlet on a server that

			has the Client Access server role installed.
ADPropertiesOnly	Optional	System.Management.Automation.SwitchParameter	The *ADPropertiesOnly* switch specifies whether to return only the properties about the virtual directory stored in Active Directory. The properties stored in the Internet Information Services (IIS) metabase aren't returned.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name or GUID of the virtual directory. It's represented as: *ServerName\VirtualDirectoryName (WebsiteName)*.
ShowMailboxVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowMailboxVirtualDirectories* switch returns

			virtual directories on Client Access servers when used in a query. If you don't use this parameter, only virtual directories on Mailbox servers are returned.
ShowBackendVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowBackendVirtualDirectories* switch specifies whether to list the virtual directories located on the Mailbox servers within the organization.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-OutlookAnywhere

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-26*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-OutlookAnywhere** cmdlet to modify the properties on a computer running Microsoft

Exchange Server 2013 enabled for Microsoft Outlook Anywhere.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-OutlookAnywhere -Identity <VirtualDirectoryIdParameter> [-Confirm
[<SwitchParameter>]] [-DefaultAuthenticationMethod <Basic | Digest | Ntlm
| Fba | WindowsIntegrated | LiveIdFba | LiveIdBasic | WSSecurity |
Certificate | NegoEx | OAuth | Adfs | Kerberos | Negotiate |
LiveIdNegotiate | Misconfigured>] [-DomainController <Fqdn>] [-
ExtendedProtectionFlags <MultiValuedProperty>] [-ExtendedProtectionSPNList
<MultiValuedProperty>] [-ExtendedProtectionTokenChecking <None | Allow |
Require>] [-ExternalClientAuthenticationMethod <Basic | Digest | Ntlm |
Fba | WindowsIntegrated | LiveIdFba | LiveIdBasic | WSSecurity |
Certificate | NegoEx | OAuth | Adfs | Kerberos | Negotiate |
LiveIdNegotiate | Misconfigured>] [-ExternalClientsRequireSsl <$true |
$false>] [-ExternalHostname <String>] [-IISAuthenticationMethods
<MultiValuedProperty>] [-InternalClientAuthenticationMethod <Basic |
Digest | Ntlm | Fba | WindowsIntegrated | LiveIdFba | LiveIdBasic |
WSSecurity | Certificate | NegoEx | OAuth | Adfs | Kerberos | Negotiate |
LiveIdNegotiate | Misconfigured>] [-InternalClientsRequireSsl <$true |
$false>] [-InternalHostname <String>] [-Name <String>] [-SSLOffloading
<$true | $false>] [-WhatIf [<SwitchParameter>]] [-XropUrl <Uri>]
```

Examples

EXAMPLE 1

This example sets the client authentication method to NTLM for the /rpc virtual directory on the Client Access server CAS01.

```
Set-OutlookAnywhere -Identity: "CAS01\rpc (Default Web
Site)" -ExternalClientAuthenticationMethod NTLM
```

EXAMPLE 2

This example sets SSL offloading for the /rpc virtual directory to false, which means that SSL isn't used for securing client connections to the Client Access server EXCH1.

```
Set-OutlookAnywhere -Identity "EXCH1\rpc (Default Web
Site)" -SSLOffloading $false
```

EXAMPLE 3

This example sets the authentication method for the /rpc virtual directory setting in IIS to NTLM.

```
Set-OutlookAnywhere -Identity "EXCH1\rpc (Default Web
Site)" -IISAuthenticationMethods NTLM
```

EXAMPLE 4

This example sets the available authentication methods for the /rpc virtual directory setting in IIS to use both Basic and NTLM authentication. After you set this value, you can use the IIS virtual directory to handle authentication for multiple applications that require different authentication

methods.

```
Set-OutlookAnywhere -Identity "EXCH1\rpc (Default Web
Site)" -IISAuthenticationMethods Basic,NTLM
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Anywhere configuration (enable, disable, change, view)" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name or GUID of the virtual directory. It's represented as: *ServerName**VirtualDirectoryName* (*WebsiteName*).
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DefaultAuthenticationMethod	Optional	Microsoft.Exchange.Data.Directory.SystemCo	The *DefaultAuthenticationMet*

		nfiguration.Authentica tionMethod	*hod* parameter can be specified to set the *ExternalClientAuthenticati onMethod*, *InternalClientAuthenticati onMethod*, and *IISAuthenticationMethods* parameters to the same authentication value. This authentication method can be set to one of these values: • `Basic` • `Ntlm` • `Negotiate` **Note:** If the *DefaultAuthenticationMet hod* parameter is specified, *InternalClientAuthenticati onMethod*, *ExternalClientAuthenticati onMethod* and *IISAuthenticationMethods* parameters cannot be used.
DomainController	Optional	Microsoft.Exchange.Da ta.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFla	Optional	Microsoft.Exchange.Da	The *ExtendedProtectionFlags*

gs		ta.MultiValuedProperty	parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain name (FQDN) format, for example ContosoMail. You

			specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory.

			The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN value`. You specify the domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionTokenChecking* parameter defines how you want to use Extended Protection for Authentication on the specified Exchange virtual

			directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the client and server will work, but may not be as secure as a connection

			using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter. **Note:** If you have a proxy server between the client and

			the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalClientAuthenticationMethod	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthenticationMethod	The *ExternalClientAuthenticationMethod* parameter specifies the authentication method used for external client authentication. Possible values include: • `Basic` • `Ntlm` • `Negotiate`
ExternalClientsRequireSsl	Optional	System.Boolean	The *ExternalClientsRequireSsl* parameter specifies whether clients connecting via Outlook Anywhere from outside the network must use Secure Sockets Layer (SSL). The default value is `$true`.
ExternalHostname	Optional	System.String	The *ExternalHostname* parameter specifies the external host name to use in the Microsoft Outlook profiles for users enabled for Outlook Anywhere.

IISAuthenticationMethods	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *IISAuthenticationMethods* parameter specifies the authentication method enabled on the /rpc virtual directory in Internet Information Services (IIS). You can set the virtual directory to allow Basic authentication or NTLM authentication. Alternatively, you can also set the virtual directory to allow both Basic and NTLM authentication. All other authentication methods are disabled. You may want to enable both Basic and NTLM authentication if you're using the IIS virtual directory with multiple applications that require different authentication methods. **Note:** When you configure this setting using the IIS interface, you can enable as many authentication methods as you want. For more information about configuring this parameter with multiple

			values, see the example later in this topic.
InternalClientAuthenticationMethod	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthenticationMethod	The *InternalClientAuthenticationMethod* parameter specifies the authentication method used for internal client authentication. Possible values include: • `Basic` • `Ntlm` • `Negotiate`
InternalClientsRequireSsl	Optional	System.Boolean	The *InternalClientsRequireSsl* parameter specifies whether clients connecting via Outlook Anywhere from inside the network require SSL. The default value is `$true`.
InternalHostname	Optional	System.String	The *InternalHostname* parameter specifies the internal hostname for the Outlook Anywhere virtual directory.
Name	Optional	System.String	The *Name* parameter specifies the name of the server being modified.
SSLOffloading	Optional	System.Boolean	The *SSLOffloading* parameter specifies whether the Client Access

			server requires SSL. This value should be set only to `$true` when an SSL hardware solution is running in front of the Client Access server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
XropUrl	Optional	System.Uri	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-OutlookConnectivity

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-OutlookConnectivity** cmdlet to test end-to-end Microsoft Outlook client connectivity in the Microsoft Exchange Server 2013 organization. This includes testing for Outlook Anywhere (RPC/HTTP) connections.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-OutlookConnectivity -ProbeIdentity <String> [-Credential
<PSCredential>] [-Hostname <String>] [-MailboxId <MailboxIdParameter>] [-
RunFromServerId <ServerIdParameter>] [-TimeOutSeconds <String>]
```

Examples

EXAMPLE 1

This example runs a protocol test from the Mailbox server.

```
Test-OutlookConnectivity -ProbeIdentity
"OutlookSelfTestProbe"
```

EXAMPLE 2

This example runs a logon test from a Client Access server for the mailbox administrator@contoso.com.

```
Test-OutlookConnectivity -ProbeIdentity
"OutlookRpcCTPProbe" -MailboxID administrator@contoso.com
```

Detailed Description

Running the **Test-OutlookConnectivity** cmdlet validates a user's Outlook connection. The command has been simplified in Exchange Server 2013 to only support ad hoc validation of a single mailbox. If the cmdlet fails, the output notes the step that failed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Test Outlook connectivity" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
ProbeIdentity	Required	System.String	The *ProbeIdentity* parameter specifies the type of probe to call.
Credential	Optional	System.Management.Automation.PSCredential	The Credential parameter specifies the credential used by the probe. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
Hostname	Optional	System.String	The *Hostname* parameter specifies the protocol endpoint target of the probe.
MailboxId	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *MailboxID* parameter specifies the mailbox that is the target of the probe.
RunFromServerId	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *RunFromServerID* parameter specifies the server on which the probe should be

			executed.
TimeOutSeconds	Optional	System.String	The *TimeOutSeconds* parameter specifies the timeout period in seconds before the probe is abandoned.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-OutlookProvider

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-OutlookProvider** cmdlet to obtain the global settings from the **AutoDiscoverConfig** object under the **Global Settings** object in Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-OutlookProvider [-Identity <OutlookProviderIdParameter>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example obtains the protocol settings for the web protocol named WEB and pipes the output to display each setting on a separate line.

```
Get-OutlookProvider -Identity WEB | Format-List
```

Detailed Description

The **Get-OutlookProvider** cmdlet gets the global settings from the **AutoDiscoverConfig** object in Active Directory and returns an **OutlookProvider** object to be managed in the Exchange Administration Center.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Autodiscover service configuration settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OutlookProviderIdParameter	The *Identity* parameter specifies the `ADIDParameter` value of the MAPI protocol for which you want to obtain global settings.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-OutlookProvider

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-OutlookProvider** cmdlet to create the **AutoDiscoverConfig** object, and then populate the object with relevant settings.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-OutlookProvider -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the **AutoDiscoverConfig** object MyOABUrl.

```
New-OutlookProvider -Name MyOABUrl
```

EXAMPLE 2

This example creates the **AutoDiscoverConfig** object Autodiscover1, and the specified domain controller writes the change to Active Directory.

```
New-OutlookProvider -DomainController DC1 -Name
Autodiscover1
```

Detailed Description

The **New-OutlookProvider** cmdlet creates the **AutoDiscoverConfig** object under the **Global Settings** object in Active Directory and sets the attributes specified.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Autodiscover service configuration settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter provides the common name of the **AutoDiscoverConfig** object. This can be a user-friendly name for identification.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-OutlookProvider

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-OutlookProvider** cmdlet to delete the **AutoDiscoverConfig** object from Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-OutlookProvider -Identity <OutlookProviderIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the **AutoDiscoverConfig** object named Test Object from Active Directory.

```
Remove-OutlookProvider -Identity "Test Object"
```

Detailed Description

The **Remove-OutlookProvider** cmdlet deletes the **AutoDiscoverConfig** object under the **Global Settings** object in Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Autodiscover service configuration settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.OutlookProviderIdParameter	The *Identity* parameter specifies the **AutoDiscoverConfig**

			object to remove from Active Directory.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-OutlookProvider

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-OutlookProvider** cmdlet to set specific global settings using the **msExchOutlookProvider** attribute on the **msExchAutoDiscoverConfig** object in Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-OutlookProvider -Identity <OutlookProviderIdParameter> [-
CertPrincipalName <String>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Name <String>] [-OutlookProviderFlags <None |
ServerExclusiveConnect>] [-RequiredClientVersions <String[]>] [-Server
<String>] [-TTL <Int32>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the duration that the Autodiscover service settings are valid for the Microsoft Outlook provider **msExchAutoDiscoverConfig**.

```
Set-OutlookProvider -Identity msExchAutoDiscoverConfig -TTL
```

Detailed Description

The **Set-OutlookProvider** cmdlet creates the global settings for the Autodiscover service. It sets the **AutoDiscoverConfig** object under the **Global Settings** object in Active Directory and sets the attributes specified in the parameters listed in the Parameters section.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Autodiscover service configuration settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.OutlookProviderIdParameter	The *Identity* parameter specifies the ADIDParameter value of the MAPI protocol for which you want to set global settings.
CertPrincipalName	Optional	System.String	The *CertPrincipalName* parameter specifies the Secure Sockets Layer (SSL) certificate principal name required for connecting to Exchange from an external location. This parameter is only used for Outlook Anywhere clients.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Name	Optional	System.String	The *Name* parameter specifies a common name for the **Outlook Provider Configuration** object. This can be a user-friendly name for identification.
OutlookProviderFlags	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.OutlookProviderFlags	The *OutlookProviderFlags* parameter specifies how Outlook clients should connect to the Exchange server. The value can be set to `ServerExclusiveConnect`, `ExternalClientsRequireSSL`, `InternalClientsRequireSSL`, or to `None` to clear the flags. The recommended value is `None`, which is also the default setting.
RequiredClientVersions	Optional	System.String[]	The *RequiredClientVersions* parameter specifies the minimum version of Microsoft Outlook that's allowed to connect to the Exchange server. This information is in the Autodiscover response to the client connection request. Valid input for this parameter is "<*MinimumVersion*>, <*ExpirationDate*>".

			<*MinimumVersion*> is the version of Outlook in the format *xx.x.xxxx.xxxx*. For example, to specify Outlook 2010 Service Pack 2 (SP2), use the value `14.0.7012.1000`. <*ExpirationDate*> is the UTC date-time when connections by older versions of Outlook will be blocked. The UTC date-time is represented in the ISO 8601 date-time format: *yyyy-mm-dd*T*hh:mm:ss.fff*Z, where *yyyy* = year, *mm* = month, *dd* = day, T indicates the beginning of the time component, *hh* = hour, *mm* = minute, *ss* = second, *fff* = fractions of a second, and Z signifies Zulu, which is another way to denote UTC. An example of a valid value for this parameter is `"14.0.7012.1000, 2014-01-01T12:00:00Z"`.
Server	Optional	System.String	The *Server* parameter specifies the Mailbox server to use for Outlook Anywhere clients.
TTL	Optional	System.Int32	The *TTL* parameter specifies the duration (in hours) that the specified settings are valid. If a value is specified, the settings are rediscovered via the Autodiscover service after the duration specified with this parameter. A value of 0 indicates that no rediscovery is

			required. The default value is 1 hour.	
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.	

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-OwaMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-OwaMailboxPolicy** cmdlet to retrieve all Microsoft Office Outlook Web App mailbox policies in a Microsoft Exchange Server 2013 organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-OwaMailboxPolicy [-Identity <MailboxPolicyIdParameter>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves the properties of all mailbox policies for the organization Contoso.

```
Get-OwaMailboxPolicy -Organization Contoso
```

EXAMPLE 2

This example retrieves the properties of the Outlook Web App mailbox policy Executives in the organization Fabrikam.

```
Get-OwaMailboxPolicy -Identity Fabrikam\Executives
```

EXAMPLE 3

This example retrieves the information for the Outlook Web App mailbox policy Corporate for the tenant Contoso in the organization Proseware.

```
Get-OwaMailboxPolicy -Identity Proseware\Contoso\Corporate
```

Detailed Description

The **Get-OwaMailboxPolicy** cmdlet retrieves information about existing Outlook Web App mailbox policies.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Web App mailbox policies" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the name of the policy or path to the policy including the name, for example, *<Organization>**<ResellerA>**<Reseller...n>**<Tenant>**<Object Name>*.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-OwaMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-OwaMailboxPolicy** cmdlet to create Microsoft Outlook Web App mailbox policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-OwaMailboxPolicy -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-IsDefault <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the Outlook Web App mailbox policy Corporate for the default organization.

```
New-OwaMailboxPolicy -Name Corporate
```

Detailed Description

Use the Set-OwaMailboxPolicy cmdlet to configure the new policy.

Note:

Changes to Outlook Web App mailbox polices may take up to 60 minutes to take effect. In on-premises Exchange 2013, you can force an update by running the command `IISRESET /noforce`.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Web App mailbox policies" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies a name for the new policy.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IsDefault	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *IsDefault*

			parameter specifies whether this policy is the default policy. The default value is `$false`. If another policy is currently set as the default, setting this parameter replaces the old default policy with this policy.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-OwaMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-OwaMailboxPolicy** cmdlet to remove Microsoft Outlook Web App mailbox policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-OwaMailboxPolicy -Identity <MailboxPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the mailbox policy Executives.

```
Remove-OwaMailboxPolicy -Identity Executives
```

EXAMPLE 2

This example removes the mailbox policy Employees for the organization Fabrikam.

```
Remove-OwaMailboxPolicy -Identity Fabrikam\Employees
```

EXAMPLE 3

This example removes the mailbox policy Corporate for the tenant Contoso in the organization Litware.

```
Remove-OwaMailboxPolicy -Identity Litware\Contoso\Corporate
```

Detailed Description

The **Remove-OwaMailboxPolicy** cmdlet removes an existing Outlook Web App mailbox policy.

Note:

Changes to Outlook Web App mailbox polices may take up to 60 minutes to take effect. In on-premises Exchange 2013, you can force an update by running the command `IISRESET /noforce`.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Web App mailbox policies" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the name of the policy or the path to the policy including the name, for example, *<Organization>**<ResellerA>**<Reseller...n>**<Tenant>**<Object Name>*.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* parameter specifies whether to suppress the warning or confirmation messages that appear during specific configuration changes.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-OwaMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-17*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-OwaMailboxPolicy** cmdlet to configure existing Microsoft Outlook Web App mailbox policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-OwaMailboxPolicy -Identity <MailboxPolicyIdParameter> [-
ActionForUnknownFileAndMIMETypes <Allow | ForceSave | Block>] [-
ActiveSyncIntegrationEnabled <$true | $false>] [-AllAddressListsEnabled
<$true | $false>] [-AllowCopyContactsToDeviceAddressBook <$true | $false>]
[-AllowedFileTypes <MultiValuedProperty>] [-AllowedMimeTypes
<MultiValuedProperty>] [-AllowOfflineOn <PrivateComputersOnly |
NoComputers | AllComputers>] [-BlockedFileTypes <MultiValuedProperty>] [-
BlockedMimeTypes <MultiValuedProperty>] [-CalendarEnabled <$true |
$false>] [-ChangePasswordEnabled <$true | $false>] [-Confirm
[<SwitchParameter>]] [-ContactsEnabled <$true | $false>] [-
DefaultClientLanguage <Int32>] [-DefaultTheme <String>] [-
DelegateAccessEnabled <$true | $false>] [-
DirectFileAccessOnPrivateComputersEnabled <$true | $false>] [-
DirectFileAccessOnPublicComputersEnabled <$true | $false>] [-
DisableFacebook <SwitchParameter>] [-DisplayPhotosEnabled <$true |
$false>] [-DomainController <Fqdn>] [-ExplicitLogonEnabled <$true |
$false>] [-FacebookEnabled <$true | $false>] [-
ForceSaveAttachmentFilteringEnabled <$true | $false>] [-ForceSaveFileTypes
<MultiValuedProperty>] [-ForceSaveMimeTypes <MultiValuedProperty>] [-
ForceWacViewingFirstOnPrivateComputers <$true | $false>] [-
ForceWacViewingFirstOnPublicComputers <$true | $false>] [-
ForceWebReadyDocumentViewingFirstOnPrivateComputers <$true | $false>] [-
ForceWebReadyDocumentViewingFirstOnPublicComputers <$true | $false>] [-
GlobalAddressListEnabled <$true | $false>] [-GroupCreationEnabled <$true |
$false>] [-InstantMessagingEnabled <$true | $false>] [-
InstantMessagingType <None | Ocs | Msn>] [-IRMEnabled <$true | $false>] [-
IsDefault <SwitchParameter>] [-JournalEnabled <$true | $false>] [-
JunkEmailEnabled <$true | $false>] [-LinkedInEnabled <$true | $false>] [-
LogonAndErrorLanguage <Int32>] [-Name <String>] [-NotesEnabled <$true |
```

```
$false>] [-OrganizationEnabled <$true | $false>] [-OutboundCharset
<AlwaysUTF8 | AutoDetect | UserLanguageChoice>] [-OWALightEnabled <$true |
$false>] [-PhoneticSupportEnabled <$true | $false>] [-PlacesEnabled <$true
| $false>] [-PredictedActionsEnabled <$true | $false>] [-
PremiumClientEnabled <$true | $false>] [-PublicFoldersEnabled <$true |
$false>] [-RecoverDeletedItemsEnabled <$true | $false>] [-
RemindersAndNotificationsEnabled <$true | $false>] [-
ReportJunkEmailEnabled <$true | $false>] [-RulesEnabled <$true | $false>]
[-SearchFoldersEnabled <$true | $false>] [-SetPhotoEnabled <$true |
$false>] [-SetPhotoURL <String>] [-SignaturesEnabled <$true | $false>] [-
SilverlightEnabled <$true | $false>] [-
SkipCreateUnifiedGroupCustomSharepointClassification <$true | $false>] [-
SMimeEnabled <$true | $false>] [-SpellCheckerEnabled <$true | $false>] [-
TasksEnabled <$true | $false>] [-TextMessagingEnabled <$true | $false>] [-
ThemeSelectionEnabled <$true | $false>] [-UMIntegrationEnabled <$true |
$false>] [-UNCAccessOnPrivateComputersEnabled <$true | $false>] [-
UNCAccessOnPublicComputersEnabled <$true | $false>] [-UseGB18030 <$true |
$false>] [-UseISO885915 <$true | $false>] [-UserDiagnosticEnabled <$true |
$false>] [-WacExternalServicesEnabled <$true | $false>] [-WacOMEXEnabled
<$true | $false>] [-WacViewingOnPrivateComputersEnabled <$true | $false>]
[-WacViewingOnPublicComputersEnabled <$true | $false>] [-WeatherEnabled
<$true | $false>] [-WebPartsFrameOptionsType <Deny | AllowFrom | None |
SameOrigin>] [-WebReadyDocumentViewingForAllSupportedTypes <$true |
$false>] [-WebReadyDocumentViewingOnPrivateComputersEnabled <$true |
$false>] [-WebReadyDocumentViewingOnPublicComputersEnabled <$true |
$false>] [-WebReadyDocumentViewingSupportedFileTypes
<MultiValuedProperty>] [-WebReadyDocumentViewingSupportedMimeTypes
<MultiValuedProperty>] [-WebReadyFileTypes <MultiValuedProperty>] [-
WebReadyMimeTypes <MultiValuedProperty>] [-WhatIf [<SwitchParameter>]] [-
WSSAccessOnPrivateComputersEnabled <$true | $false>] [-
WSSAccessOnPublicComputersEnabled <$true | $false>]
```

Examples

EXAMPLE 1

This example disables access to the calendar for the mailbox policy `Corporate` for the tenant Contoso in the organization EMEA.

```
Set-OwaMailboxPolicy -Identity EMEA\Contoso\Corporate -
CalendarEnabled $false
```

EXAMPLE 2

This example disables access to the Tasks folder for the default mailbox policy.

```
Set-OwaMailboxPolicy -Identity Default -TasksEnabled $false
```

EXAMPLE 3

This example sets the allowed file type extensions to .doc and .pdf for the default mailbox policy, allowing users to save files with those extensions locally or view them from a web browser.

```
Set-OwaMailboxPolicy -Identity Default -AllowedFileTypes
'.doc', '.pdf'
```

Detailed Description

By default, an Outlook Web App mailbox policy named Default is created in the organization. The **Set-OwaMailboxPolicy** cmdlet configures an existing Outlook Web App mailbox policy.

Note:

Changes to Outlook Web App mailbox polices may take up to 60 minutes to take effect. In on-premises Exchange 2013, you can force an update by running the command `IISRESET /noforce`.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Web App mailbox policies" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the name of the policy or the path to the policy including the name, for example, *<Organization>**<ResellerA>**<Reseller...n>**<Tenant>**<Object Name>*.
ActionForUnknownFileAndMIMETypes	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AttachmentBlockingActions	The *ActionForUnknownFileAndMIMETypes* parameter specifies how to handle files that aren't included in other File Access Management lists. The following values are valid for this parameter: • `Allow` • `ForceSave` • `Block`

ActiveSyncIntegrationEnabled	Optional	System.Boolean	The *ActiveSyncIntegrationEnabled* parameter specifies whether to disable Microsoft Exchange ActiveSync on the Outlook Web App **Options** page.
AllAddressListsEnabled	Optional	System.Boolean	The *AllAddressListsEnabled* parameter specifies which address lists are available to the user. You can use either `$true` or `$false` as follows: • If set to `$true`, users can view all address lists. • If set to `$false`, users can view only the global address list.
AllowCopyContactsToDeviceAddressBook	Optional	System.Boolean	The *AllowCopyContactsToDeviceAddressBook* parameter specifies if users can copy the contents of their Contacts folder to a mobile device's native address book when using OWA for Devices.
AllowedFileTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AllowedFileTypes* parameter specifies the extensions of file types

			that the user can save locally and view from a web browser. If the same extensions are in multiple settings lists, the most secure setting overrides the less secure settings.
AllowedMimeTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AllowedMimeTypes* parameter creates extensions of MIME attachments that users can save locally and view from a web browser. If the same extensions are in multiple settings lists, the most secure setting overrides the less secure settings.
AllowOfflineOn	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AllowOfflineOnEnum	The *AllowOfflineOn* parameter specifies which computers can use Outlook Web App in offline mode. The possible values are `PrivateComputersOnly`, `NoComputers`, or `AllComputers`. The value is set to `AllComputers` by default. If you set the value to `PrivateComputersOnly`, only users who log into

			Outlook Web App using the **Private** option are able to use Outlook Web App in offline mode. If the feature is enabled, and if users are using a supported browser, they can use Outlook Web App in offline mode. Users can turn the feature on or off in Outlook Web App. To turn the feature on, users click the gear icon, and then select **Stop using offline**. To turn the feature off, users click the gear icon, and then select **Use mail offline**. The supported browsers for this feature in Microsoft Exchange Server 2013 are Microsoft Internet Explorer 10, Safari 4, or Chrome 16. For more information, see Using Outlook Web App offline.
BlockedFileTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BlockedFileTypes* parameter specifies a list of extensions of attachments that are blocked. Attachments that contain these blocked

			extensions can't be saved locally or viewed from a web browser.
BlockedMimeTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BlockedMimeTypes* parameter specifies a list of MIME extensions of attachments that are blocked. Attachments that contain these blocked MIME extensions can't be saved locally or viewed from a web browser.
CalendarEnabled	Optional	System.Boolean	The *CalendarEnabled* parameter specifies whether to enable or disable the calendar for users.
ChangePasswordEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *ChangePasswordEnabled* parameter specifies whether users can change their passwords from inside Outlook Web App.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ContactsEnabled	Optional	System.Boolean	The *ContactsEnabled* parameter specifies whether Contacts are enabled for users.
DefaultClientLanguage	Optional	System.Int32	This parameter has been deprecated and is no longer used.
DefaultTheme	Optional	System.String	The *DefaultTheme* parameter specifies the default theme used by Outlook Web App when the user hasn't selected a theme.
DelegateAccessEnabled	Optional	System.Boolean	The *DelegateAccessEnabled* parameter specifies whether delegates can use Outlook Web App to open folders they have delegate access to through this virtual directory.
DirectFileAccessOnPrivateComputersEnabled	Optional	System.Boolean	The *DirectFileAccessOnPrivateComputersEnabled* parameter specifies the

			left-click options on attachments when the user has chosen to log on using the **Private** option. If this parameter is set to `$true`, **Open** is an available option. If it's set to `$false`, the **Open** option is disabled.
DirectFileAccessOnPublicComputersEnabled	Optional	System.Boolean	The *DirectFileAccessOnPublicComputersEnabled* parameter specifies the left-click options on attachments when the user has chosen to log on using the **Public** option. If this parameter is set to `$true`, **Open** is an available option. If it's set to `$false`, the **Open** option is disabled.
DisableFacebook	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *DisableFacebook* switch disables Facebook integration with Outlook Web App.
DisplayPhotosEnabled	Optional	System.Boolean	The *DisplayPhotosEnabled* parameter specifies

			whether users see sender photos in Outlook Web App. The possible values for this parameter are `$true` or `$false`. The default value is `$false`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExplicitLogonEnabled	Optional	System.Boolean	The *ExplicitLogonEnabled* parameter specifies whether to allow a user to open someone else's mailbox in Outlook Web App. If this parameter is set to `$true`, it allows a user to open someone else's mailbox in Outlook Web App.
FacebookEnabled	Optional	System.Boolean	The *FacebookEnabled* parameter specifies whether users can synchronize their

			Facebook contacts to their Contacts folder. The possible values for this parameter are `$true` or `$false`. The default value is `$true`.
ForceSaveAttachmentFilteringEnabled	Optional	System.Boolean	The *ForceSaveAttachmentFilteringEnabled* parameter specifies whether files which are included in the list of extensions created by the *ForceSaveFileTypes* parameter are filtered before the user can save them.
ForceSaveFileTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ForceSaveFileTypes* parameter creates a list of extensions of attachments that can be opened only after the file is saved locally on the user's computer.
ForceSaveMimeTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ForceSaveMimeTypes* parameter specifies the MIME types of attachments that can be opened only after the file is saved locally on the user's computer.
ForceWacViewingFirst	Optional	System.Boolean	The

OnPrivateComputers			*ForceWacViewingFirstOnPrivateComputers* parameter specifies whether a user who has logged on using the **Private** option can open an Office file directly without first viewing it as a web page.
ForceWacViewingFirstOnPublicComputers	Optional	System.Boolean	The *ForceWacViewingFirstOnPublicComputers* parameter specifies whether a user who has logged on using the **Public** option can open an Office file directly without first viewing it as a web page.
ForceWebReadyDocumentViewingFirstOnPrivateComputers	Optional	System.Boolean	The *ForceWebReadyDocumentViewingFirstOnPrivateComputers* parameter specifies whether a user who has logged on using the **Private** option can open a document directly without first viewing it as a web page.
ForceWebReadyDocumentViewingFirstOnP	Optional	System.Boolean	The *ForceWebReadyDocumen*

ublicComputers			*tViewingFirstOnPublicComputers* parameter specifies whether a user who has logged on using the **Public** option can open a document directly without first viewing it as a web page.
GlobalAddressListEnabled	Optional	System.Boolean	The *GlobalAddressListEnabled* parameter specifies whether to show the global address list in Outlook Web App.
GroupCreationEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
InstantMessagingEnabled	Optional	System.Boolean	The *InstantMessagingEnabled* parameter specifies whether to enable instant messaging in Outlook Web App.
InstantMessagingType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.InstantMessagingTypeOptions	The *InstantMessagingType* parameter specifies the type of instant messaging provider to be used. Set this parameter to `None` for no provider and `Ocs` for Microsoft Office Communication Server. The `Msn` value is no longer

			used and will be deprecated.
IRMEnabled	Optional	System.Boolean	The *IRMEnabled* parameter specifies whether the Information Rights Management (IRM) feature is enabled.
IsDefault	Optional	System.Management.Automation.SwitchParameter	The *IsDefault* parameter specifies whether this policy is the default policy. The default value is `$false`. If another policy is currently set as the default, setting this parameter replaces the old default policy with this policy.
JournalEnabled	Optional	System.Boolean	The *JournalEnabled* parameter specifies whether the Journal folder is visible.
JunkEmailEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *JunkEmailEnabled* parameter specifies whether the Junk Email management tools are enabled.
LinkedInEnabled	Optional	System.Boolean	This parameter is

			available only in the cloud-based service. The *LinkedInEnabled* parameter specifies whether users can synchronize their LinkedIn contacts to their Contacts folder. The possible values for this parameter are `$true` or `$false`. The default value is `$true`.
LogonAndErrorLanguage	Optional	System.Int32	The *LogonAndErrorLanguage* parameter specifies which language Outlook Web App uses for forms-based authentication and for error messages that occur when a user's current language setting can't be read. When this parameter has a value of 0, the language selection is undefined.
Name	Optional	System.String	The *Name* parameter specifies a name for the policy.
NotesEnabled	Optional	System.Boolean	The *NotesEnabled* parameter specifies whether the Notes folder is visible in Outlook Web

			App.
OrganizationEnabled	Optional	System.Boolean	When the *OrganizationEnabled* parameter is set to `$false`, the **Automatic Reply** option doesn't include external and internal options, the address book doesn't show the organization hierarchy, and the **Resources** tab in Calendar forms is disabled.
OutboundCharset	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.OutboundCharsetOptions	The *OutboundCharset* parameter specifies the character set used for messages sent by users on a specific Outlook Web App virtual directory.
OWALightEnabled	Optional	System.Boolean	The *OWALightEnabled* parameter, when set to `$false`, removes the option to use the light version of Outlook Web App from the logon page and removes the **Accessibility** check box from the **General Options** tab. This parameter doesn't apply

			to Outlook Web App.
PhoneticSupportEnabled	Optional	System.Boolean	The *PhoneticSupportEnabled* parameter specifies phonetically spelled entries in the address book. This parameter is available for use in Japan.
PlacesEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
PredictedActionsEnabled	Optional	System.Boolean	The *PredictedActionsEnabled* parameter specifies whether you want Outlook Web App to customize the user experience by making predictions about the action that a user should take on a particular item. This value is set to `$false` by default. If this value is set to `$true`, Outlook Web App tries to make suggestions for the user. The following are examples: • The user interface may change the order of items in an options list based on the context the user is in.

			• Outlook Web App may highlight an icon or other user interface element that's the next logical step in the completion of a task. • Upon initiating a move email item operation, the user interface may suggest the folder that you want to move email items based on previous actions of the user.
PremiumClientEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
PublicFoldersEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
RecoverDeletedItemsEnabled	Optional	System.Boolean	The *RecoverDeletedItemsEnabled* parameter specifies whether a user can use Outlook Web App to view, recover, or delete permanently items that have been deleted from the Deleted Items folder. By default, the *RecoverDeletedItemsEnabled* parameter is set to `$true`. If the

			RecoverDeletedItemsEnabled parameter is set to `$false`, the items deleted from the Deleted Items folder are retained. However, users can't view, recover, or permanently delete them using Outlook Web App.
RemindersAndNotificationsEnabled	Optional	System.Boolean	The *RemindersAndNotificationsEnabled* parameter specifies whether notifications and reminders are enabled in Outlook Web App. This parameter doesn't apply to the light version of Outlook Web App.
ReportJunkEmailEnabled	Optional	System.Boolean	The *ReportJunkEmailEnabled* parameter specifies whether users can report messages as junk to Microsoft in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. If you set this parameter to `$false`, the **Report mail as junk** window

			doesn't appear after users select **Mark as junk**. This parameter is meaningful only when the *JunkEmailEnabled* parameter is set to `$true`.
RulesEnabled	Optional	System.Boolean	The *RulesEnabled* parameter specifies whether a user can view, create, or modify server-side rules using Outlook Web App. By default, the *RulesEnabled* parameter is set to `$true`. If the *RulesEnabled* parameter is set to `$false`, users must use Microsoft Outlook to view, create, and modify server-side rules.
SearchFoldersEnabled	Optional	System.Boolean	The *SearchFoldersEnabled* parameter specifies whether Search Folders are available in Outlook Web App.
SetPhotoEnabled	Optional	System.Boolean	The *SetPhotoEnabled* parameter specifies whether users can add, change, and remove their sender photo in Outlook Web App. The possible values for this parameter

			are `$true` or `$false`. The default value is `$false`. When this value is set to `$true`, users can manage their sender photo using two methods. They can click their name in the upper-right corner of Outlook Web App, click **Change**, and then browse to the photo they want to use. Alternatively, users can manage their photo by clicking the gear icon in the upper-right corner of Outlook Web App, and then clicking **Options** > **Account** > **My account** > **Edit** > **Change**.
SetPhotoURL	Optional	System.String	The *SetPhotoURL* parameter specifies the location of the user photos. This value isn't set by default.
SignaturesEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
SilverlightEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
SkipCreateUnifiedGro	Optional	System.Boolean	The

upCustomSharepointClassification			*SkipCreateUnifiedGroupCustomSharepointClassification* parameter causes the custom Sharepoint page during unified group creation to be skipped.
SMimeEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used. To configure the S/MIME settings in Outlook Web App, use the **Get-SmimeConfig** and **Set-SmimeConfig** cmdlets. For more information, see S/MIME for message signing and encryption.
SpellCheckerEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
TasksEnabled	Optional	System.Boolean	The *TasksEnabled* parameter specifies whether a user can use the Tasks feature in Outlook Web App. This parameter doesn't apply to the light version of Outlook Web App.
TextMessagingEnabled	Optional	System.Boolean	The *TextMessagingEnabled* parameter specifies

			whether users can send and receive text messages. This parameter doesn't apply to the light version of Outlook Web App.
ThemeSelectionEnabled	Optional	System.Boolean	The *ThemeSelectionEnabled* parameter specifies whether users can select a theme in Outlook Web App.
UMIntegrationEnabled	Optional	System.Boolean	The *UMIntegrationEnabled* parameter specifies whether Unified Messaging is enabled in Outlook Web App. This setting applies only if Unified Messaging has been enabled for a user using the **Enable-UMMailbox** cmdlet. This parameter doesn't apply to the light version of Outlook Web App.
UNCAccessOnPrivateComputersEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
UNCAccessOnPublicComputersEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.

UseGB18030	Optional	System.Boolean	The *UseGB18030* parameter specifies when to use the character set GB18030. This parameter is a character-handling registry key that works in coordination with the *OutboundCharset* registry key. When the *UseGB18030* parameter is set to `$true`, the character set GB18030 is used wherever GB2312 would have been used.
UseISO885915	Optional	System.Boolean	The *UseISO885915* parameter specifies when to use the character set ISO8859-15. This parameter is a character-handling registry key that works in coordination with the *OutboundCharset* registry key. When the *UseISO885915* parameter is set to `$true`, the character set ISO8859-15 is used wherever ISO8859-1 would have been used.
UserDiagnosticEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.

WacExternalServicesEnabled	Optional	System.Boolean	The *WacExternalServicesEnabled* parameter enables or disables external services that may be used by Web Access Companion (WAC). The default value is `$true`.
WacOMEXEnabled	Optional	System.Boolean	The *WacOMEXEnabled* parameter enables or disables apps for Outlook. The default value is `$false`.
WacViewingOnPrivateComputersEnabled	Optional	System.Boolean	The *WacViewingOnPrivateComputersEnabled* parameter specifies whether a user who has logged into Outlook Web App using the **Private** option can view supported Office files using Outlook Web App.
WacViewingOnPublicComputersEnabled	Optional	System.Boolean	The *WacViewingOnPublicComputersEnabled* parameter specifies whether a user who has logged into Outlook Web App using the **Public** option can view supported Office files using Outlook Web

			App.
WeatherEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
WebPartsFrameOptionsType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.WebPartsFrameOptions	The *WebPartsFrameOptionsType* parameter specifies what sources can access Outlook Web App web parts in IFRAME or FRAME elements. This parameter can have the following values: • `None` This indicates that there are no restrictions on displaying Outlook Web App content in a frame. • `SameOrigin` This is the default value and the recommended value. This allows display of Outlook Web App content only in a frame that has the same origin as the content. • `Deny` This blocks display of Outlook Web App content in a frame regardless of the origin of the site attempting to access it. • `AllowFrom` This isn't yet available. It will be implemented in a later release.
WebReadyDocumentViewingForAllSupport	Optional	System.Boolean	The *WebReadyDocumentView*

edTypes			*ingForAllSupportedTypes* parameter enables WebReady Document Viewing for all supported file and MIME types. If this parameter is set to `$false`, use the *WebReadyFileTypes* and *WebReadyMimeTypes* parameters to set which file and MIME types to convert.
WebReadyDocumentViewingOnPrivateComputersEnabled	Optional	System.Boolean	The *WebReadyDocumentViewingOnPrivateComputersEnabled* parameter specifies whether WebReady Document Viewing is enabled when the user selects the **This is a private computer** option on the Outlook Web App logon page.
WebReadyDocumentViewingOnPublicComputersEnabled	Optional	System.Boolean	The *WebReadyDocumentViewingOnPublicComputersEnabled* parameter specifies whether WebReady Document Viewing is enabled when the user selects the **This is a public or shared**

			computer option on the Outlook Web App logon page.
WebReadyDocumentViewingSupportedFileTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The read-only *WebReadyDocumentViewingSupportedFileTypes* parameter lists the file types supported by the conversion engine.
WebReadyDocumentViewingSupportedMimeTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The read-only *WebReadyDocumentViewingSupportedMimeTypes* parameter lists the MIME types supported by the conversion engine
WebReadyFileTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *WebReadyFileTypes* parameter creates a list of file types on which WebReady Document Viewing is performed.
WebReadyMimeTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *WebReadyMimeTypes* parameter creates a list of MIME types on which WebReady Document Viewing is performed.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WSSAccessOnPrivateComputersEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
WSSAccessOnPublicComputersEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-OwaVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-OwaVirtualDirectory** cmdlet to retrieve all Microsoft Office Outlook Web App virtual directories on a computer running Microsoft Exchange Server 2013 that has the Client Access server role installed.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-OwaVirtualDirectory -Server <ServerIdParameter> <COMMON PARAMETERS>

Get-OwaVirtualDirectory [-Identity <VirtualDirectoryIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-ADPropertiesOnly <SwitchParameter>] [-
DomainController <Fqdn>] [-ShowBackEndVirtualDirectories
<SwitchParameter>] [-ShowMailboxVirtualDirectories <SwitchParameter>]
```

Examples

EXAMPLE 1

This example uses the *Server* parameter to retrieve all Outlook Web App virtual directories on the server Contoso.

```
Get-OwaVirtualDirectory -Server Contoso
```

EXAMPLE 2

This example uses the *Identity* parameter to retrieve the configuration settings on the Outlook Web App virtual directory owa (default Web site) on the server Contoso.

```
Get-OwaVirtualDirectory -Identity "Contoso\owa (default Web
site)"
```

Detailed Description

The **Get-OwaVirtualDirectory** cmdlet retrieves and displays the configuration settings currently set on Outlook Web App virtual directories on Exchange. The **Get-OwaVirtualDirectory** cmdlet can also retrieve and display the configuration settings on a specific Outlook Web App virtual directory.

The **Get-OwaVirtualDirectory** cmdlet can be run on a local server or run remotely if the server name is specified in the *Identity* or *Server* parameters. This cmdlet can also be run without parameters to retrieve the configuration settings from all Outlook Web App virtual directories on all Internet Information Services (IIS) websites located on the Client Access servers in the organization.

The **Get-OwaVirtualDirectory** cmdlet can be run on any server that has the Exchange Server administration tools installed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Web App virtual directories" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name or GUID of the server that hosts the Outlook Web App virtual directories that you want to display. It's required if you aren't running the command on a server that has the Client Access server role installed.
ADPropertiesOnly	Optional	System.Management.Automation.SwitchParameter	The *ADPropertiesOnly* switch specifies whether to return only the properties about the virtual directory stored in Active Directory. The properties stored in the Internet Information Services (IIS) metabase aren't returned.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from

			Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name or GUID of an Outlook Web App virtual directory. It's represented as: *ServerName\VirtualDirectoryName (WebsiteName)*.
ShowBackEndVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowBackEndVirtualDirectories* switch is no longer used and will be deprecated.
ShowMailboxVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowMailboxVirtualDirectories* switch specifies whether the virtual directories on the Mailbox server role within the organization are shown.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-OwaVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-OwaVirtualDirectory** cmdlet to create a Microsoft Office Outlook Web App virtual directory in an existing Internet Information Services (IIS) website on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-OwaVirtualDirectory [-ApplicationRoot <String>] [-AppPoolId <String>]
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
ExtendedProtectionFlags <MultiValuedProperty>] [-ExtendedProtectionSPNList
<MultiValuedProperty>] [-ExtendedProtectionTokenChecking <None | Allow |
Require>] [-ExternalAuthenticationMethods <MultiValuedProperty>] [-
ExternalUrl <Uri>] [-InternalUrl <Uri>] [-Path <String>] [-Role
<ClientAccess | Mailbox>] [-Server <ServerIdParameter>] [-WebSiteName
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the Outlook Web App virtual directory in an IIS website on the local Exchange server, which isn't a default website.

```
New-OwaVirtualDirectory -WebSiteName "Contoso.com"
```

Detailed Description

By default, when Exchange is installed, the Outlook Web App virtual directory owa is created in the default IIS website on the local server running Exchange. The **New-OwaVirtualDirectory** cmdlet must be run on the Exchange server hosting the Client Access server role on which you want to host the new virtual directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Web App virtual directories" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
ApplicationRoot	Optional	System.String	The *ApplicationRoot* parameter sets the path of the virtual directory in the metabase. **Note:** This parameter hasn't been implemented.
AppPoolId	Optional	System.String	The *AppPoolId* parameter sets the IIS application pool in which Outlook Web App runs. We recommend that you leave this parameter at its default setting.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain

			name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're

			using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN value`. You specify the domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenChecki	The *ExtendedProtectionTokenChecking* parameter defines how you want to

		ngMode	use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the

			client and server will work, but may not be as secure as a connection using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter.

			Note: If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalAuthenticationMethods	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalAuthenticationMethods* parameter, a Services Discovery property setting, specifies the authentication methods supported on the Exchange server from outside the firewall. **Note:** This parameter hasn't been implemented; however, it can be set by using the **Set-OwaVirtualDirectory** cmdlet.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the host name to be used to connect to the Exchange server from outside the firewall. This setting is important when Secure Sockets Layer (SSL) is used. **Note:**

			You can only configure this setting on Exchange virtual directories. The default Exchange virtual directory is /owa.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the host name to be used to connect to the Exchange server from inside the firewall. This setting is important when SSL is used. **Note:** You can only configure this setting on Exchange virtual directories. The default Exchange virtual directory is /owa.
Path	Optional	System.String	The *Path* parameter sets the file system path of the virtual directory. This parameter should be used with care and only when you must use a different file system path than the default.
Role	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.VirtualDirectoryRole	The *Role* parameter specifies the configuration that should be used when the virtual directory is created. The following are the values that can be used with this parameter: • `ClientAccess` Configures the virtual

			directory for use on a Client Access server. • `Mailbox` Configures the virtual directory for use on a Mailbox server.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to create the virtual directory. You can use any value that uniquely identifies the server, for example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the virtual directory will be created on the server where the Remote PowerShell session is established This will always be a Mailbox server. If you want to create the virtual directory on a Client Access server or another Mailbox server, you must use the *Server* parameter.
WebSiteName	Optional	System.String	The *WebSiteName* parameter specifies the

			name of the IIS website under which the Outlook Web App virtual directory is created.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-OwaVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-OwaVirtualDirectory** cmdlet to remove Microsoft Office Outlook Web App virtual directories located in the Internet Information Services (IIS) website on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-OwaVirtualDirectory -Identity <VirtualDirectoryIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the specified Outlook Web App virtual directory owa located on the default IIS website on the Exchange server Contoso.

```
Remove-OwaVirtualDirectory -Identity "Contoso\owa (default
Web site)"
```

Detailed Description

To run the **Remove-OwaVirtualDirectory** cmdlet, you must specify the name of the Outlook Web App virtual directory that you want to remove. You might be required to use the **Remove-OwaVirtualDirectory** cmdlet in the following situations:

- The **Get-OwaVirtualDirectory** cmdlet detects an Outlook Web App virtual directory deleted in IIS but not deleted in Active Directory. These abandoned Outlook Web App virtual directory objects are known as Active Directory orphans. We recommend that you remove this kind of Outlook Web App virtual directory in Active Directory by using the **Remove-OwaVirtualDirectory** cmdlet.
- You are troubleshooting an Outlook Web App configuration issue that requires you to delete the existing Outlook Web App virtual directory and then re-create the Outlook Web App virtual directory.

Caution:

The **Remove-OwaVirtualDirectory** cmdlet permanently removes an Outlook Web App virtual directory or directories. When you use the **Remove-OwaVirtualDirectory** cmdlet, make sure that you don't accidentally delete the default Outlook Web App virtual directory. There are no default values for the **Remove-OwaVirtualDirectory** cmdlet.

Note:

The **Remove-OwaVirtualDirectory** cmdlet supports the **Confirm** flag.

The **Remove-OwaVirtualDirectory** cmdlet can be run on any server that has the Exchange Server administration tools installed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Web App virtual directories" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name or GUID of an Outlook Web App virtual directory. It's represented as: *ServerName**VirtualDirectoryName (WebsiteName)*.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-OwaVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-09*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-OwaVirtualDirectory** cmdlet to modify the properties of Microsoft Outlook Web App virtual directories on a server running Microsoft Exchange Server 2013 that has the Client Access server role installed.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-OwaVirtualDirectory -Identity <VirtualDirectoryIdParameter> [-
ActionForUnknownFileAndMIMETypes <Allow | ForceSave | Block>] [-
ActiveSyncIntegrationEnabled <$true | $false>] [-AdfsAuthentication <$true
| $false>] [-AllAddressListsEnabled <$true | $false>] [-
AllowCopyContactsToDeviceAddressBook <$true | $false>] [-AllowedFileTypes
<MultiValuedProperty>] [-AllowedMimeTypes <MultiValuedProperty>] [-
AllowOfflineOn <PrivateComputersOnly | NoComputers | AllComputers>] [-
AnonymousFeaturesEnabled <$true | $false>] [-BasicAuthentication <$true |
$false>] [-BlockedFileTypes <MultiValuedProperty>] [-BlockedMimeTypes
<MultiValuedProperty>] [-CalendarEnabled <$true | $false>] [-
ChangePasswordEnabled <$true | $false>] [-ClientAuthCleanupLevel <High |
Low>] [-Confirm [<SwitchParameter>]] [-ContactsEnabled <$true | $false>]
[-DefaultClientLanguage <Int32>] [-DefaultDomain <String>] [-DefaultTheme
<String>] [-DelegateAccessEnabled <$true | $false>] [-DigestAuthentication
<$true | $false>] [-DirectFileAccessOnPrivateComputersEnabled <$true |
$false>] [-DirectFileAccessOnPublicComputersEnabled <$true | $false>] [-
DisplayPhotosEnabled <$true | $false>] [-DomainController <Fqdn>] [-
Exchange2003Url <Uri>] [-ExchwebProxyDestination <NotSpecified |
MailboxServer | PublicFolderServer>] [-ExplicitLogonEnabled <$true |
$false>] [-ExtendedProtectionFlags <MultiValuedProperty>] [-
ExtendedProtectionSPNList <MultiValuedProperty>] [-
ExtendedProtectionTokenChecking <None | Allow | Require>] [-
ExternalAuthenticationMethods <MultiValuedProperty>] [-ExternalUrl <Uri>]
[-FailbackUrl <Uri>] [-FilterWebBeaconsAndHtmlForms <UserFilterChoice |
ForceFilter | DisableFilter>] [-ForceSaveAttachmentFilteringEnabled <$true
| $false>] [-ForceSaveFileTypes <MultiValuedProperty>] [-
ForceSaveMimeTypes <MultiValuedProperty>] [-
ForceWacViewingFirstOnPrivateComputers <$true | $false>] [-
ForceWacViewingFirstOnPublicComputers <$true | $false>] [-
ForceWebReadyDocumentViewingFirstOnPrivateComputers <$true | $false>] [-
ForceWebReadyDocumentViewingFirstOnPublicComputers <$true | $false>] [-
FormsAuthentication <$true | $false>] [-GlobalAddressListEnabled <$true |
$false>] [-GzipLevel <Off | Low | High | Error>] [-
InstantMessagingCertificateThumbprint <String>] [-InstantMessagingEnabled
<$true | $false>] [-InstantMessagingServerName <String>] [-
InstantMessagingType <None | Ocs | Msn>] [-IntegratedFeaturesEnabled
<$true | $false>] [-InternalUrl <Uri>] [-IRMEnabled <$true | $false>] [-
JournalEnabled <$true | $false>] [-JunkEmailEnabled <$true | $false>] [-
LiveIdAuthentication <$true | $false>] [-LogonAndErrorLanguage <Int32>] [-
LogonFormat <FullDomain | PrincipalName | UserName>] [-
LogonPageLightSelectionEnabled <$true | $false>] [-
LogonPagePublicPrivateSelectionEnabled <$true | $false>] [-NotesEnabled
<$true | $false>] [-NotificationInterval <Int32>] [-OAuthAuthentication
<$true | $false>] [-OrganizationEnabled <$true | $false>] [-
OutboundCharset <AlwaysUTF8 | AutoDetect | UserLanguageChoice>] [-
OWALightEnabled <$true | $false>] [-PlacesEnabled <$true | $false>] [-
PredictedActionsEnabled <$true | $false>] [-PremiumClientEnabled <$true |
$false>] [-PublicFoldersEnabled <$true | $false>] [-
RecoverDeletedItemsEnabled <$true | $false>] [-RedirectToOptimalOWAServer
<$true | $false>] [-RemindersAndNotificationsEnabled <$true | $false>] [-
RemoteDocumentsActionForUnknownServers <Allow | Block>] [-
RemoteDocumentsAllowedServers <MultiValuedProperty>] [-
RemoteDocumentsBlockedServers <MultiValuedProperty>] [-
RemoteDocumentsInternalDomainSuffixList <MultiValuedProperty>] [-
ReportJunkEmailEnabled <$true | $false>] [-RulesEnabled <$true | $false>]
[-SearchFoldersEnabled <$true | $false>] [-SetPhotoEnabled <$true |
$false>] [-SetPhotoURL <String>] [-SignaturesEnabled <$true | $false>] [-
SilverlightEnabled <$true | $false>] [-SMimeEnabled <$true | $false>] [-
SpellCheckerEnabled <$true | $false>] [-TasksEnabled <$true | $false>] [-
TextMessagingEnabled <$true | $false>] [-ThemeSelectionEnabled <$true |
$false>] [-UMIntegrationEnabled <$true | $false>] [-
UNCAccessOnPrivateComputersEnabled <$true | $false>] [-
UNCAccessOnPublicComputersEnabled <$true | $false>] [-UseGB18030 <$true |
$false>] [-UseISO885915 <$true | $false>] [-UserContextTimeout <Int32>] [-
UserDiagnosticEnabled <$true | $false>] [-VirtualDirectoryType
<NotSpecified | Mailboxes | PublicFolders | Exchweb | Exadmin>] [-
WacViewingOnPrivateComputersEnabled <$true | $false>] [-
WacViewingOnPublicComputersEnabled <$true | $false>] [-WeatherEnabled
```

```
<$true | $false>] [-WebPartsFrameOptionsType <Deny | AllowFrom | None |
SameOrigin>] [-WebReadyDocumentViewingForAllSupportedTypes <$true |
$false>] [-WebReadyDocumentViewingOnPrivateComputersEnabled <$true |
$false>] [-WebReadyDocumentViewingOnPublicComputersEnabled <$true |
$false>] [-WebReadyDocumentViewingSupportedFileTypes
<MultiValuedProperty>] [-WebReadyDocumentViewingSupportedMimeTypes
<MultiValuedProperty>] [-WebReadyFileTypes <MultiValuedProperty>] [-
WebReadyMimeTypes <MultiValuedProperty>] [-WhatIf [<SwitchParameter>]] [-
WindowsAuthentication <$true | $false>] [-
WSSAccessOnPrivateComputersEnabled <$true | $false>] [-
WSSAccessOnPublicComputersEnabled <$true | $false>]
```

Examples

EXAMPLE 1

This example sets the *DirectFileAccessOnPublicComputersEnabled* parameter to `$false` on the Outlook Web App virtual directory owa on the default Internet Information Services (IIS) website on the Exchange server Contoso.

```
Set-OwaVirtualDirectory -Identity "Contoso\owa (default Web
site)" -DirectFileAccessOnPublicComputersEnabled $false
```

EXAMPLE 2

This example sets the *ActionForUnknownFileAndMIMETypes* parameter to `Block` on the default Outlook Web App virtual directory owa on the server Contoso.

```
Set-OwaVirtualDirectory -Identity "Contoso\owa (default Web
site)" -ActionForUnknownFileAndMIMETypes Block
```

Detailed Description

Before you run the **Set-OwaVirtualDirectory** cmdlet, consider the following items:

- You must have Write access to virtual directory objects in Active Directory. If you don't have the necessary permissions and you try to run the **Set-OwaVirtualDirectory** cmdlet on the Active Directory virtual directory object, the cmdlet fails.
- You must have Write access to virtual directory objects in the metabase for some properties, such as **Authentication** and **GZip**. If you don't have the necessary permissions to run the **Set-OwaVirtualDirectory** cmdlet on a metabase virtual directory object or on a parameter that writes to the metabase, the cmdlet fails.
- Verify that the data source can be read. Depending on the properties that you want to set on an Outlook Web App virtual directory, you may want to run the cmdlet in a test environment on the Outlook Web App virtual directory object in Active Directory, the metabase, or both.
- You can run the **Set-OwaVirtualDirectory** cmdlet on any server that has the Exchange Server administration tools installed.
- Several parameters for the **Set-OwaVirtualDirectory** cmdlet can contain more than one value. These are known as multivalued properties. Make sure that you modify multivalued properties

correctly. For information, see Modifying multivalued properties.

- Many of the Outlook Web App virtual directory settings require you to run the **IISReset /noforce** command before the change takes effect. For example, when you enable or disable forms-based authentication, or when you enable or disable the **Private computer** option on the sign-in page.
- To switch from forms-based authentication to Basic authentication, you must first disable forms-based authentication, and then as a separate task, enable Basic authentication. You can't disable forms-based authentication and enable Basic authentication in a single task.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Outlook Web App virtual directories" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name or GUID of an Outlook Web App virtual directory. The *Identity* parameter is represented as: *ServerName**VirtualDirectoryName (WebsiteName)*.
ActionForUnknownFileAndMIMETypes	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AttachmentBlockingActions	The *ActionForUnknownFileAndMIMETypes* parameter specifies how to handle files that aren't included in other File Access Management lists. The following values are valid for this parameter: • `Allow` • `ForceSave` • `Block`

ActiveSyncIntegrationEnabled	Optional	System.Boolean	The *ActiveSyncIntegrationEnabled* parameter disables Microsoft Exchange ActiveSync on the Outlook Web App **Options** page. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AdfsAuthentication	Optional	System.Boolean	The *AdfsAuthentication* parameter enables or disables Active Directory Federation Services (ADFS) authentication on the Outlook Web App virtual directory. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. The ADFS authentication settings for **Set-OwaVirtualDirectory** and **Set-EcpVirtualDirectory** are related. You need to set the *AdfsAuthentication* parameter on **Set-EcpVirtualDirectory** to `$true` before you can set the *AdfsAuthentication* parameter on **Set-**

			OwaVirtualDirectory to `$true`. Likewise, you need to set the *AdfsAuthentication* parameter on **Set-OwaVirtualDirectory** to `$false` before you can set the *AdfsAuthentication* parameter on **Set-EcpVirtualDirectory** to `$false`.
AllAddressListsEnabled	Optional	System.Boolean	The *AllAddressListsEnabled* parameter specifies which address lists are available to the user. You can use one of the following values: • If set to `$true`, users can view all address lists. • If set to `$false`, users can view only the global address list (GAL). The default value is `$true`.
AllowCopyContactsToDeviceAddressBook	Optional	System.Boolean	The *AllowCopyContactsToDeviceAddressBook* parameter specifies if users can copy the contents of their Contacts folder to a mobile device's native address book when using

			OWA for Devices. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowedFileTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AllowedFileTypes* parameter specifies the extensions of file types that the user can save locally and view from a web browser. If the same extensions are in multiple settings lists, the most secure setting overrides the less secure settings. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax:

			`@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
AllowedMimeTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AllowedMimeTypes* parameter specifies extensions of MIME attachments that users can save locally and view from a web browser. If the same extensions are in multiple settings lists, the most secure setting overrides the less secure settings. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<va`

			`lue2>"...}`.
AllowOfflineOn	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AllowOfflineOnEnum	The *AllowOfflineOn* parameter specifies which computers can use Outlook Web App in offline mode. The possible values are: • `NoComputers` • `AllComputers` • `PrivateComputersOnly` The default value is `AllComputers`. By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. In order to show the **Private computer** option at sign-in, you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`. If the feature is enabled and users are using a supported browser, they can use Outlook Web App in offline mode. Users can turn the feature on or off in Outlook Web App. To turn the feature on, click the gear icon and then

			select **Stop using offline**. To turn the feature off, click the gear icon and then select **Use mail offline**. The supported browsers are Microsoft Internet Explorer 10, Safari 4, or Chrome 16. For more information, see Using Outlook Web App offline.
AnonymousFeaturesEnabled	Optional	System.Boolean	The *AnonymousFeaturesEnabled* parameter specifies whether you want to allow Outlook Web App users that are logged on anonymously to access specific features. For example, if this value is set to `$true`, users logged on using anonymous authentication can view and change meeting content. This parameter is set to `$true` by default.
BasicAuthentication	Optional	System.Boolean	The *BasicAuthentication* parameter enables or disables Basic authentication on the Outlook Web App virtual directory. This parameter

			can be used with the *FormsAuthentication* parameter or with the *DigestAuthentication* and *WindowsAuthentication* parameters. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
BlockedFileTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BlockedFileTypes* parameter specifies a list of extensions of attachments that are blocked. Attachments that contain these blocked extensions can't be saved locally or viewed from a web browser. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or

			more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
BlockedMimeTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BlockedMimeTypes* parameter specifies a list of MIME extensions of attachments that are blocked. Attachments that contain these blocked MIME extensions can't be saved locally or viewed from a web browser. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax:

			`@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
CalendarEnabled	Optional	System.Boolean	The *CalendarEnabled* parameter specifies whether to enable the calendar for users. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
ChangePasswordEnabled	Optional	System.Boolean	The *ChangePasswordEnabled* parameter specifies whether users can change their passwords from inside Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
ClientAuthCleanupLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ClientAuthCleanupLevels	The *ClientAuthCleanupLevel* parameter specifies how much of the cache is cleared when the user logs off in Outlook Web App. This parameter doesn't apply to the light version of Outlook Web App.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ContactsEnabled	Optional	System.Boolean	The *ContactsEnabled* parameter specifies whether Contacts are enabled for users. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
DefaultClientLanguage	Optional	System.Int32	This parameter has been deprecated and is no longer used.
DefaultDomain	Optional	System.String	The *DefaultDomain* parameter specifies which domain to use when the *LogonFormat* parameter is set to `UserName`.
DefaultTheme	Optional	System.String	The *DefaultTheme* parameter specifies the default theme used by Outlook Web App when the user hasn't selected a

			theme.
DelegateAccessEnabled	Optional	System.Boolean	The *DelegateAccessEnabled* parameter specifies whether delegates can use Outlook Web App to open folders, which they have delegate access to, through this virtual directory. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
DigestAuthentication	Optional	System.Boolean	The *DigestAuthentication* parameter enables or disables Digest authentication on the virtual directory. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. This parameter can't be used with the *FormsAuthentication* parameter.
DirectFileAccessOnPrivateComputersEnabled	Optional	System.Boolean	The *DirectFileAccessOnPrivateComputersEnabled* parameter specifies the

			left-click options on attachments. If this parameter is set to `$true`, **Open** is an available option. If it's set to `$false`, the **Open** option is disabled. The default value is `$true`. By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. In order to show the **Private computer** option at sign-in, you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`.
DirectFileAccessOnPublicComputersEnabled	Optional	System.Boolean	The *DirectFileAccessOnPublicComputersEnabled* parameter specifies the left-click options on attachments when a user logs on after unchecking the **Private computer** option. If this parameter is set to `$true`, **Open** is an available option. If it's set to `$false`, the **Open**

			option is disabled. The default value is `$true`. By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. Therefore, this parameter is meaningful only if you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`. This allows you to uncheck the **Private computer** option at sign-in, which indicates you are connecting from a public computer.
DisplayPhotosEnabled	Optional	System.Boolean	The *DisplayPhotosEnabled* parameter specifies whether users see sender photos in Outlook Web App. The possible values for this parameter are `$true` or `$false`. The default value is `$true`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration

			change to Active Directory.
Exchange2003Url	Optional	System.Uri	This parameter has been deprecated and is no longer used.
ExchwebProxyDestination	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExchwebProxyDestinations	This parameter has been deprecated and is no longer used.
ExplicitLogonEnabled	Optional	System.Boolean	The *ExplicitLogonEnabled* parameter specifies whether to allow a user to open someone else's mailbox in Outlook Web App. If this parameter is set to `$true`, it allows a user to open someone else's mailbox in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are:

			• **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless

			certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing

			entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN` value. You specify the

			domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/mail.contoso.com. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionTokenChecking* parameter defines how you want to use Extended Protection for Authentication on the specified Exchange virtual

			directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the client and server will work, but may not be as secure as a connection

			using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter. **Note:** If you have a proxy server between the client and

			the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalAuthenticationMethods	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalAuthenticationMethods* parameter specifies the authentication methods supported on the Exchange server from outside the firewall.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the host name used to connect to the Exchange server from outside the firewall. This setting is also important when Secure Sockets Layer (SSL) is used. **Note:** You can only configure this option on Exchange 2013 virtual directories. The default Exchange virtual directory is /owa.
FailbackUrl	Optional	System.Uri	The *FailbackUrl* parameter specifies the host name Outlook Web App uses to connect to the Client Access server after

			failback in a site resilience process and requires a separate Domain Name System (DNS) entry pointing to the original Client Access server's IP address. The *FailbackUrl* parameter must be different from the *ExternalUrl* parameter.
FilterWebBeaconsAndHtmlForms	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.WebBeaconFilterLevels	The *FilterWebBeaconsAndHtmlForms* parameter specifies how web beacons are handled. The possible values are `UserFilterChoice`, `ForceFilter`, and `DisableFilter`.
ForceSaveAttachmentFilteringEnabled	Optional	System.Boolean	The *ForceSaveAttachmentFilteringEnabled* parameter specifies whether files included in the list of extensions created by the *ForceSaveFileTypes* parameter are filtered before the user can save them. Valid input for this parameter is `$true` or `$false`. The default value

			is `$false`.
ForceSaveFileTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ForceSaveFileTypes* parameter specifies a list of extensions of attachments that can be opened only after the file is saved locally on the user's computer. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
ForceSaveMimeTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ForceSaveMimeTypes* parameter specifies the MIME types of attachments that can be

			opened only after the file is saved locally on the user's computer.
ForceWacViewingFirstOnPrivateComputers	Optional	System.Boolean	The *ForceWacViewingFirstOnPrivateComputers* parameter specifies whether a user can open an Office file directly without first viewing it as a web page. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. In order to show the **Private computer** option at sign-in, you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`.
ForceWacViewingFirstOnPublicComputers	Optional	System.Boolean	The *ForceWacViewingFirstOnPublicComputers* parameter specifies whether a user who logs

			on after unchecking the **Private computer** option can open an Office file directly without first viewing it as a web page. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. Therefore, this parameter is meaningful only if you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`. This allows you to uncheck the **Private computer** option at sign-in, which indicates you are connecting from a public computer.
ForceWebReadyDocumentViewingFirstOnPrivateComputers	Optional	System.Boolean	The *ForceWebReadyDocumentViewingFirstOnPrivateComputers* parameter specifies whether a user can open a document directly without first

			viewing it as a web page. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. In order to show the **Private computer** option at sign-in, you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`.
ForceWebReadyDocumentViewingFirstOnPublicComputers	Optional	System.Boolean	The *ForceWebReadyDocumentViewingFirstOnPublicComputers* parameter specifies whether a user who logs on after unchecking the **Private computer** option can open a document directly without first viewing it as a web page. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.

			By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. Therefore, this parameter is meaningful only if you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`. This allows you to uncheck the **Private computer** option at sign-in, which indicates you are connecting from a public computer.
FormsAuthentication	Optional	System.Boolean	The *FormsAuthentication* parameter enables or disables forms-based authentication on the Outlook Web App virtual directory. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. If the *FormsAuthentication* parameter is set to `$true`, the *BasicAuthentication* parameter is set to `$true`, and the

			DigestAuthentication and *WindowsAuthentication* parameters are set to `$false`.
GlobalAddressListEnabled	Optional	System.Boolean	The *GlobalAddressListEnabled* parameter specifies whether to show the global address list in Outlook Web App.
GzipLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.GzipLevel	The *GzipLevel* parameter sets Gzip configuration information for the Outlook Web App virtual directory.
InstantMessagingCertificateThumbprint	Optional	System.String	The *InstantMessagingCertificateThumbprint* parameter specifies the trusted certificate used to communicate between the instant messaging server and the Client Access server. Use the Get-ExchangeCertificate cmdlet to find the thumbprint of the certificate.
InstantMessagingEnabled	Optional	System.Boolean	The *InstantMessagingEnabled* parameter specifies

			whether to enable instant messaging in Outlook Web App.
InstantMessagingServerName	Optional	System.String	The *InstantMessagingServerName* parameter specifies the fully qualified domain name (FQDN) of the instant messaging server or set of servers behind a load balancing device.
InstantMessagingType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.InstantMessagingTypeOptions	The *InstantMessagingType* parameter specifies the type of instant messaging provider to be used. Set this parameter to `None` for no provider and `Ocs` for Microsoft Office Communications Server. The `Msn` value is no longer used and will be deprecated.
IntegratedFeaturesEnabled	Optional	System.Boolean	The *IntegratedFeaturesEnabled* parameter specifies whether to allow Outlook Web App users who are logged on using Integrated Windows authentication to access specific features. For

			example, if this value is set to `$true`, users logged on using Integrated Windows authentication can view and change meeting content. This is set to `$true` by default.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the host name of the Exchange server for connection from inside the firewall. This setting is also important when SSL is used. **Note:** You can only configure this option on Exchange 2013 virtual directories. The default Exchange virtual directory is /owa.
IRMEnabled	Optional	System.Boolean	The *IRMEnabled* parameter specifies whether the Information Rights Management (IRM) feature is enabled.
JournalEnabled	Optional	System.Boolean	The *JournalEnabled* parameter specifies whether the Journal folder is visible.
JunkEmailEnabled	Optional	System.Boolean	The *JunkEmailEnabled* parameter specifies whether the Junk Email

			management tools are enabled.
LiveIdAuthentication	Optional	System.Boolean	The *LiveIdAuthentication* parameter specifies whether to configure Outlook Web App to use logon via a Microsoft account (formerly known as Windows Live ID). Set this parameter to `$true` to configure Outlook Web App to use logon via a Microsoft account.
LogonAndErrorLanguage	Optional	System.Int32	The *LogonAndErrorLanguage* parameter specifies which language Outlook Web App uses for forms-based authentication and for error messages that occur when a user's current language setting can't be read. When this parameter has a value of `0`, the language selection is undefined.
LogonFormat	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.LogonFormats	The *LogonFormat* parameter specifies the type of logon format forms-based authentication must use on the Outlook Web App

			sign-in page. Possible attributes are **FullDomain**, **UserName**, or **PrincipalName**. If you specify the **FullDomain** attribute, the **User name** field on the Outlook Web App sign-in page requires the user name to be entered in the format *domain**user name*. If you specify the **UserName** attribute, you must also specify the *DefaultDomain* parameter. If you specify the **PrincipalName** attribute, the **User name** field on the Outlook Web App sign-in page requires a user principal name (UPN) address. This sign-in method works only for users whose UPN name is the same as their email address.
LogonPageLightSelectionEnabled	Optional	System.Boolean	The *LogonPageLightSelectionEnabled* parameter specifies whether the

			Outlook Web App sign-in page includes the option to sign in to the light version of Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
LogonPagePublicPrivateSelectionEnabled	Optional	System.Boolean	The *LogonPagePublicPrivateSelectionEnabled* parameter specifies whether the Outlook Web App sign-in page includes the **Private computer** option at sign-in. By default, Valid input for this parameter is `$true` or `$false`. The default value is `$false`. By default, Exchange 2013 assumes you are connecting from a private computer. To make the **Private computer** option available to check or uncheck, set the value of this parameter to `$true`, and then restart IIS.

NotesEnabled	Optional	System.Boolean	The *NotesEnabled* parameter specifies whether the Notes folder is visible in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
NotificationInterval	Optional	System.Int32	This parameter has been deprecated and is no longer used.
OAuthAuthentication	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
OrganizationEnabled	Optional	System.Boolean	When the *OrganizationEnabled* parameter is set to `$false`, the **Automatic Reply** option doesn't include external and internal options, the address book doesn't show the organization hierarchy, and the **Resources** tab in Calendar forms is disabled. The default value is `$true`.
OutboundCharset	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.Outbound	The *OutboundCharset* parameter specifies the character set used for

		CharsetOptions	messages sent by users on a specific Outlook Web App virtual directory.
OWALightEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
PlacesEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
PredictedActionsEnabled	Optional	System.Boolean	The *PredictedActionsEnabled* parameter specifies whether you want Outlook Web App to customize the user experience by making predictions about the action that a user should take on a particular item. This value is set to `$false` by default. If this value is set to `$true`, Outlook Web App tries to make suggestions for the user. For example: • The user interface may change the order of items in an options list based on the context the user is in. • Outlook Web App may highlight an icon or other user interface

			element that is the next logical step in the completion of a task. • Upon initiating a move email item operation, the user interface may suggest the folder that you want to move email items based on previous actions of the user.
PremiumClientEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
PublicFoldersEnabled	Optional	System.Boolean	The *PublicFoldersEnabled* parameter specifies whether a user can browse or read items in public folders using Outlook Web App. By default, the *PublicFoldersEnabled* parameter is set to `$true`. If the *PublicFoldersEnabled* parameter is set to `$false`, users can only access their private mailboxes in Outlook Web App.
RecoverDeletedItemsEnabled	Optional	System.Boolean	The *RecoverDeletedItemsEnabled* parameter specifies

			whether a user can use Outlook Web App to view, recover, or delete permanently items that have been deleted from the Deleted Items folder. By default, the *RecoverDeletedItemsEnabled* parameter is set to `$true`. If the *RecoverDeletedItemsEnabled* parameter is set to `$false`, the items deleted from the Deleted Items folder are retained. However, users can't view, recover, or permanently delete them using Outlook Web App.
RedirectToOptimalOWAServer	Optional	System.Boolean	The *RedirectToOptimalOWAServer* parameter, when set to `$true`, causes Outlook Web App to use service discovery to find the best Client Access server to use after a user authenticates. If redirection is disabled, Outlook Web App doesn't redirect clients to the most optimal Client Access server.

			Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
RemindersAndNotificationsEnabled	Optional	System.Boolean	The *RemindersAndNotificationsEnabled* parameter specifies whether notifications and reminders are enabled in Outlook Web App. This parameter doesn't apply to the light version of Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
RemoteDocumentsActionForUnknownServers	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.RemoteDocumentsActions	This parameter has been deprecated and is no longer used.
RemoteDocumentsAllowedServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter has been deprecated and is no longer used.
RemoteDocumentsBlockedServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter has been deprecated and is no longer used.
RemoteDocumentsInternalDomainSuffixList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter has been deprecated and is no

		y	longer used.
ReportJunkEmailEnabled	Optional	System.Boolean	The *ReportJunkEmailEnabled* parameter specifies whether users can report messages as junk to Microsoft in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. If you set this parameter to `$false`, the **Report mail as junk** window doesn't appear after users select **Mark as junk**. This parameter is meaningful only when the *JunkEmailEnabled* parameter is set to `$true`.
RulesEnabled	Optional	System.Boolean	The *RulesEnabled* parameter specifies whether a user can view, create, or modify server-side rules using Outlook Web App. By default, the *RulesEnabled* parameter is set to `$true`. If the *RulesEnabled* parameter is set to `$false`, users must use Microsoft Outlook to view, create, and modify

			server-side rules.
SearchFoldersEnabled	Optional	System.Boolean	The *SearchFoldersEnabled* parameter specifies whether Search Folders are available in Outlook Web App.
SetPhotoEnabled	Optional	System.Boolean	The *SetPhotoEnabled* parameter specifies whether users can add, change, and remove their sender photo in Outlook Web App. The possible values for this parameter are `$true` or `$false`. The default value is `$false`. When this value is set to `$true`, users can manage their sender photo using two methods. They can click their name in the upper-right corner of Outlook Web App, click **change**, and then browse to the photo they want to use. Alternatively, users can manage their photo by clicking the gear icon in the upper-right corner of Outlook Web App, and then clicking **Options** > **Account** > **My account** > **Edit** > **Change**.

SetPhotoURL	Optional	System.String	The *SetPhotoURL* parameter specifies the location of the user photos. This value isn't set by default.
SignaturesEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
SilverlightEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
SMimeEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used. To configure the S/MIME settings in Outlook Web App, use the **Get-SmimeConfig** and **Set-SmimeConfig** cmdlets. For more information, see S/MIME for message signing and encryption.
SpellCheckerEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
TasksEnabled	Optional	System.Boolean	The *TasksEnabled* parameter specifies whether a user can use the Tasks feature in Outlook Web App. This

			parameter doesn't apply to the light version of Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
TextMessagingEnabled	Optional	System.Boolean	The *TextMessagingEnabled* parameter specifies whether users can send and receive text messages. This parameter doesn't apply to the light version of Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
ThemeSelectionEnabled	Optional	System.Boolean	The *ThemeSelectionEnabled* parameter specifies whether users can select a theme in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
UMIntegrationEnabled	Optional	System.Boolean	The *UMIntegrationEnabled*

			parameter specifies whether Unified Messaging is enabled on Outlook Web App. This setting applies only if Unified Messaging has been enabled for a user using the **Enable-UMMailbox** cmdlet. This parameter doesn't apply to the light version of Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
UNCAccessOnPrivateComputersEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
UNCAccessOnPublicComputersEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
UseGB18030	Optional	System.Boolean	The *UseGB18030* parameter specifies when to use the character set GB18030. This parameter is a character-handling registry key that works in coordination with the *OutboundCharset* registry key. When the

			UseGB18030 parameter is set to `$true`, the character set GB18030 is used wherever GB2312 would have been used. The default value is `$false`.
UseISO885915	Optional	System.Boolean	The *UseISO885915* parameter specifies when to use the character set ISO8859-15. This parameter is a character-handling registry key that works in coordination with the *OutboundCharset* registry key. When the *UseISO885915* parameter is set to `$true`, the character set ISO8859-15 is used wherever ISO8859-1 would have been used. The default value is `$false`.
UserContextTimeout	Optional	System.Int32	The *UserContextTimeout* parameter specifies the time-out setting, in minutes, for a user context object. This parameter doesn't limit public and private forms-based authentication time-out settings.

UserDiagnosticEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
VirtualDirectoryType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.VirtualDirectoryTypes	This parameter has been deprecated and is no longer used.
WacViewingOnPrivateComputersEnabled	Optional	System.Boolean	The *WacViewingOnPrivateComputersEnabled* parameter specifies whether a user can view supported Office files using Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. In order to show the **Private computer** option at sign-in, you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`.
WacViewingOnPublicComputersEnabled	Optional	System.Boolean	The *WacViewingOnPublicCom*

			putersEnabled parameter specifies whether a user who logs on after unchecking the **Private computer** option can view supported Office files using Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. Therefore, this parameter is meaningful only if you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`. This allows you to uncheck the **Private computer** option at sign-in, which indicates you are connecting from a public computer.
WeatherEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
WebPartsFrameOptio	Optional	Microsoft.Exchange.Da	The *WebPartsFrameOptionsTy*

nsType		ta.Directory.SystemConfiguration.WebPartsFrameOptions	*pe* parameter specifies what sources can access Outlook Web App web parts in IFRAME or FRAME elements. This parameter can have the following values: • `None` This indicates that there are no restrictions on displaying Outlook Web App content in a frame. • `SameOrigin` This is the default value and the recommended value. This allows display of Outlook Web App content only in a frame that has the same origin as the content. • `Deny` This blocks display of Outlook Web App content in a frame regardless of the origin of the site attempting to access it. • `AllowFrom` This isn't yet available. It will be implemented in a later release.
WebReadyDocumentViewingForAllSupportedTypes	Optional	System.Boolean	The *WebReadyDocumentViewingForAllSupportedTypes* parameter enables WebReady Document Viewing for all supported file and MIME types. If this parameter is set to

			`$false`, use the *WebReadyFileTypes* and *WebReadyMimeTypes* parameters to set which file and MIME types to convert. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
WebReadyDocumentViewingOnPrivateComputersEnabled	Optional	System.Boolean	The *WebReadyDocumentViewingOnPrivateComputersEnabled* parameter enables or disables WebReady Document Viewing. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. In order to show the **Private computer** option at sign-in, you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`.

WebReadyDocumentViewingOnPublicComputersEnabled	Optional	System.Boolean	The *WebReadyDocumentViewingOnPublicComputersEnabled* parameter enables or disables WebReady Document Viewing for a user who logs on after unchecking the **Private computer** option. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, Outlook Web App in Exchange 2013 assumes you are connecting from a private computer. Therefore, this parameter is meaningful only if you set the *LogonPagePublicPrivateSelectionEnabled* parameter to the value `$true`. This allows you to uncheck the **Private computer** option at sign-in, which indicates you are connecting from a public computer.
WebReadyDocumentViewingSupportedFileTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The read-only *WebReadyDocumentViewingSupportedFileTypes*

			parameter lists the file types supported by the conversion engine. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
WebReadyDocumentViewingSupportedMimeTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The read-only *WebReadyDocumentViewingSupportedMimeTypes* parameter lists the MIME types supported by the conversion engine. To enter multiple values and overwrite any existing entries, use the following syntax:

			`<value1>,<value2>...` If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
WebReadyFileTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *WebReadyFileTypes* parameter creates a list of file types on which WebReady Document Viewing is performed. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...` If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...`

			To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
WebReadyMimeTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *WebReadyMimeTypes* parameter creates a list of MIME types on which WebReady Document Viewing is performed. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsAuthentication	Optional	System.Boolean	The *WindowsAuthentication* parameter enables or disables Integrated Windows authentication on the Outlook Web App virtual directory. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
WSSAccessOnPrivateComputersEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.
WSSAccessOnPublicComputersEnabled	Optional	System.Boolean	This parameter has been deprecated and is no longer used.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-PopConnectivity

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-PopConnectivity** cmdlet to verify that the POP3 service is running as expected. The **Test-PopConnectivity** cmdlet can be used to test the POP3 functionality for a specified Client Access server for all mailboxes on servers running Microsoft Exchange Server 2013 in the same Active Directory site.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-PopConnectivity [-ClientAccessServer <ServerIdParameter>] [-
ConnectionType <Plaintext | Tls | Ssl>] [-MailboxCredential
<PSCredential>] [-PerConnectionTimeout <Int32>] [-PortClientAccessServer
<Int32>] [-TrustAnySSLCertificate <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-LightMode
<SwitchParameter>] [-MailboxServer <ServerIdParameter>] [-
MonitoringContext <SwitchParameter>] [-ResetTestAccountCredentials
<SwitchParameter>] [-Timeout <UInt32>] [-UserType <LEGACY | EDU | BPOS>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the POP3 connectivity for the Client Access server Contoso12 by using the credentials for the user contoso\kweku.

```
Test-PopConnectivity -ClientAccessServer:Contoso12 -
MailboxCredential:(Get-Credential contoso\kweku)
```

EXAMPLE 2

This example tests the POP3 connectivity of the specific Client Access server Contoso12 and tests all

Exchange mailboxes.

```
Test-PopConnectivity -ClientAccessServer:Contoso12
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Test POP3 settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
ClientAccessServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *ClientAccessServer* parameter specifies the name of the Client Access server to test.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ConnectionType	Optional	Microsoft.Exchange.Monitoring.ProtocolConnectionType	The *ConnectionType* parameter specifies the type of connection used to connect to the Client Access server.

			This setting can be set to `Plaintext`, `Tls`, or `ssl`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
LightMode	Optional	System.Management.Automation.SwitchParameter	The *LightMode* parameter instructs the command to perform only a test logon to the server using the POP3 protocol. If you don't use this parameter, the test also tests the sending and receiving of a message using the POP3 protocol.
MailboxCredential	Optional	System.Management.Automation.PSCredential	The *MailboxCredential* parameter specifies the mailbox credential for a single mailbox test.
MailboxServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *MailboxServer* parameter specifies the name of the Mailbox server to test. If not specified, all Mailbox servers in the local

			Active Directory site are tested.
MonitoringContext	Optional	System.Management.Automation.SwitchParameter	The *MonitoringContext* switch includes the associated monitoring events and performance counters in the results. You don't need to specify a value with this switch. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
PerConnectionTimeout	Optional	System.Int32	The *PerConnectionTimeout* parameter specifies the amount of time, in seconds, to wait per connection for the test operation to finish. The default value for the *PerConnectionTimeout* parameter is 120 seconds. You must

			specify a time-out value greater than 0 seconds and less than 120 seconds. We recommend that you configure this parameter with a value of 5 seconds or more.
PortClientAccessServer	Optional	System.Int32	The *PortClientAccessServer* parameter specifies the port to use to connect to the Client Access server. The default port is 110 for POP3. The valid range is from 0 through 65,535.
ResetTestAccountCredentials	Optional	System.Management.Automation.SwitchParameter	The *ResetTestAccountCredentials* switch resets the password for the test account that's used to run this command. The password for the test account is typically reset every seven days. Use this switch to force a password reset any time it's required for security reasons.
Timeout	Optional	System.UInt32	The *Timeout* parameter

			specifies the amount of time, in seconds, to wait for the test operation to finish. The default value for the *Timeout* parameter is 180 seconds. You must specify a time-out value greater than 0 seconds and less than 1 hour (3,600 seconds). We recommend that you configure this parameter with a value of 5 seconds or more.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	The *TrustAnySSLCertificate* parameter specifies whether Secure Sockets Layer (SSL) certificate validation failures are reported. This parameter instructs the command to check the POP3 service without generating an error when the certificate doesn't match the URL of the Client Access server.
UserType	Optional	Microsoft.Exchange.Mo	This parameter is

		nitoring.DatacenterUserType	reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-POPSettings

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-POPSettings** cmdlet to display the properties of a single server running Microsoft

Exchange Server 2013 that has the Client Access server role installed and is running the POP3 service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PopSettings [-DomainController <Fqdn>] [-Server <ServerIdParameter>]
```

Examples

EXAMPLE 1

This example displays the parameters and values for the Client Access server CAS01 running the POP3 service.

```
Get-POPSettings -Server CAS01
```

Detailed Description

With the **Get-POPSettings** cmdlet, you can view the settings for the POP3 service running on an Exchange Client Access server. Information can only be returned about servers located in the Exchange organization from which the cmdlet is being run.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "POP3 settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies whether the command returns the

			properties of an individual Client Access server in your organization for which you're viewing POP3 settings.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-PopSettings

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-PopSettings** cmdlet to set specific POP3 settings for the server running Microsoft Exchange Server 2013 that has the Client Access server role installed and that's running the Microsoft Exchange POP3 service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-PopSettings [-AuthenticatedConnectionTimeout <EnhancedTimeSpan>] [-
Banner <String>] [-CalendarItemRetrievalOption <iCalendar | intranetUrl |
InternetUrl | Custom>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-EnableExactRFC822Size <$true | $false>] [-
EnableGSSAPIAndNTLMAuth <$true | $false>] [-EnforceCertificateErrors
<$true | $false>] [-ExtendedProtectionPolicy <None | Allow | Require>] [-
ExternalConnectionSettings <MultiValuedProperty>] [-
InternalConnectionSettings <MultiValuedProperty>] [-
LiveIdBasicAuthReplacement <$true | $false>] [-LogFileLocation <String>]
[-LogFileRollOverSettings <Hourly | Daily | Weekly | Monthly>] [-LoginType
<PlainTextLogin | PlainTextAuthentication | SecureLogin>] [-
LogPerFileSizeQuota <Unlimited>] [-MaxCommandSize <Int32>] [-
MaxConnectionFromSingleIP <Int32>] [-MaxConnections <Int32>] [-
MaxConnectionsPerUser <Int32>] [-MessageRetrievalMimeFormat <TextOnly |
```

```
HtmlOnly | HtmlAndTextAlternative | TextEnrichedOnly |
TextEnrichedAndTextAlternative | BestBodyFormat | Tnef>] [-
MessageRetrievalSortOrder <Ascending | Descending>] [-OwaServerUrl <Uri>]
[-PreAuthenticatedConnectionTimeout <EnhancedTimeSpan>] [-
ProtocolLogEnabled <$true | $false>] [-ProxyTargetPort <Int32>] [-Server
<ServerIdParameter>] [-SSLBindings <MultiValuedProperty>] [-
SuppressReadReceipt <$true | $false>] [-UnencryptedOrTLSBindings
<MultiValuedProperty>] [-WhatIf [<SwitchParameter>]] [-X509CertificateName
<String>]
```

Examples

EXAMPLE 1

This example sets the plain text or TLS connection to the Client Access server CAS01. In this example, the connection uses an IP address of 10.0.0.0 and a port number of 993.

```
Set-PopSettings -Server "CAS01" -UnencryptedOrTLSBindings
10.0.0.0:993
```

EXAMPLE 2

This example turns on POP3 protocol logging. It also changes the POP3 protocol logging directory to C:\Pop3Logging.

```
Set-PopSettings -ProtocolLogEnabled $true -LogFileLocation
"C:\Pop3Logging"
```

EXAMPLE 3

This example changes the POP3 protocol logging to create a new log file when a log file reaches 2 megabytes (MB).

```
Set-PopSettings -LogPerFileSizeQuota 2000000
```

EXAMPLE 4

This example changes the POP3 protocol logging to create a new log file every hour.

```
Set-PopSettings -LogPerFileSizeQuota 0 -
LogFileRollOverSettings Hourly
```

Detailed Description

You can run the **Set-PopSettings** cmdlet for a single Client Access server that has the Microsoft Exchange POP3 service installed, or for all Exchange Client Access servers that have the Microsoft Exchange POP3 service installed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "POP3 settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
AuthenticatedConnectionTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *AuthenticatedConnectionTimeout* parameter specifies the time to wait before closing an idle authenticated connection. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Valid input for this parameter is `00:00:30` to `1:00:00`. The default setting is `00:30:00` or 30 minutes.
Banner	Optional	System.String	The *Banner* parameter specifies the banner string displayed after a connection to a Client Access server has been established.
CalendarItemRetrievalOption	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.CalendarItemRetrievalOptions	The *CalendarItemRetrievalOption* parameter specifies the type of calendar item

			returned when the calendar is accessed by using POP3. The default value is `iCalendar`. You can specify the value for this parameter by using a numerical value or text string. The following values are available: • `0` or `iCalendar` • `1` or `intranetUrl` • `2` or `InternetUrl` • `3` or `Custom` If you're using `3` or `Custom`, you must specify the *OwaServerUrl* parameter setting.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration

			change to Active Directory.
EnableExactRFC822Size	Optional	System.Boolean	The *EnableExactRFC822Size* parameter calculates the exact size of each MIME message that can be retrieved from the server. When you set this parameter to `$true`, the exact size of MIME messages stored on the Exchange server is available to POP3 or IMAP4 client programs that rely on knowing the exact size of each MIME message. **Note:** This parameter is set to `$false` by default. If you don't set this option to `$true`, the size of each MIME message that the Exchange server returns to POP3 and IMAP4 client programs may be slightly different than the exact size of the message. Because setting this option to `$true` can negatively affect performance, you should only use this option if many of your users are using a client that requires knowing the exact size of MIME

			messages.
EnableGSSAPIAndNTLMAuth	Optional	System.Boolean	The *EnableGSSAPIAndNTLMAuth* parameter specifies whether connections can use Integrated Windows authentication (NTLM) by using the Generic Security Services application programming interface (GSSAPI). This setting applies to connections where Transport Layer Security (TLS) is disabled. By default, this parameter is set to `$true`. You can disable NTLM for POP3 connections by setting the value to `$false`. NTLM authentication isn't supported for POP3 connections in Microsoft Exchange Server 2010 release to manufacturing (RTM). Support for NTLM authentication for POP3 connections was brought back in Exchange 2010 Service Pack 1 (SP1).
EnforceCertificateErrors	Optional	System.Boolean	The *EnforceCertificateErrors* parameter specifies

			whether to enforce valid Secure Sockets Layer (SSL) certificates. To use this parameter, specify the destination Client Access server for which you want to enforce valid SSL certificates. If the *EnforceCertificateErrors* parameter is set to `$true` and the proxy's target certificate isn't valid, the proxy logon attempt fails. The default setting is `$false`.
ExtendedProtectionPolicy	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionPolicy* parameter specifies how you want to use Extended Protection for Authentication for POP3 connections. By default, this parameter is set to `None`. The possible values are: • `None` Extended Protection for Authentication won't be used. • `Allow` Extended Protection for Authentication will be used only if the connecting POP3 connection supports it.

			Otherwise, the connections will be established without Extended Protection for Authentication. • `Require` Extended Protection for Authentication will be required for all POP3 connections. If the connecting host doesn't support Extended Protection for Authentication, the connection will be rejected. Extended Protection for Authentication enhances the protection and handling of credentials when authenticating network connections using Integrated Windows authentication. Integrated Windows authentication is also known as NTLM. We strongly recommend that you use Extended Protection for Authentication if you're using Integrated Windows authentication. To use Extended Protection for Authentication, the client and server computers must meet the specific requirements. These

			include operating system requirements and security update requirements. In addition, the POP3 client program must support the use of Extended Protection for Authentication.
ExternalConnectionSettings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalConnectionSettings* parameter specifies the host name, port, and encryption type that Exchange uses when POP3 clients connect to their email from outside your corporate network. Enter a value using the format: <*HostName*>:<*Port*>:<*Encryption Type*>. The <*Encryption Type*> part of the multivalued value is optional. Valid values for <*Encryption Type*> are either TLS or SSL.
InternalConnectionSettings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *InternalConnectionSettings* parameter specifies the host name, port, and encryption type used when POP3 clients

			connect to their email from inside your corporate network. This setting is also used to specify the host name, port, and encryption type used when a user connection is forwarded to another Client Access server. Enter a value using the format: <*HostName*>:<*Port*>:<*Encryption Type*>. The <*Encryption Type*> part of the multivalued value is optional. Valid values for <*Encryption Type*> are either TLS or SSL. If your server name is Server01 and your domain is Contoso.com, the default value is `Server01.Contoso.com:995:SSL, Server01.Contoso.com:110:TLS`.
LiveIdBasicAuthReplacement	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LogFileLocation	Optional	System.String	The *LogFileLocation* parameter specifies the location for the POP3 protocol log files. By

			default, POP3 protocol log files are located in the C:\Program Files\Microsoft\Exchange Server\V15\Logging\Pop3 directory.
LogFileRollOverSettings	Optional	Microsoft.Exchange.Diagnostics.LogFileRollOver	The *LogFileRollOverSettings* parameter defines how frequently POP3 protocol logging creates a new log file. By default, a new log file is created daily. You can specify the value for this parameter by using a numerical value or text string. The possible values are: • `1` or `Hourly` • `2` or `Daily` • `3` or `Weekly` • `4` or `Monthly` This setting only applies when the value for the *LogPerFileSizeQuota* parameter is set to `0`.
LoginType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.LoginOptions	The *LoginType* parameter specifies the authentication setting used for the Client Access server running the Microsoft Exchange POP3 service. You can specify the value for this

			parameter by using a numerical value or text string. The possible values are: • `1` or `PlainTextLogin` • `2` or `PlainTextAuthentication` • `3` or `SecureLogin` The default value is `SecureLogin`.
LogPerFileSizeQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *LogPerFileSizeQuota* parameter defines the maximum size of a POP3 protocol log file in bytes. By default, this value is set to `0`. When this value is set to `0`, a new protocol log file is created at the frequency specified by the *LogFileRollOverSettings* parameter.
MaxCommandSize	Optional	System.Int32	The *MaxCommandSize* parameter specifies the maximum size of a single command. The default size is `512` bytes. The possible values are from 40 through 1024 bytes.
MaxConnectionFromSingleIP	Optional	System.Int32	The *MaxConnectionFromSingleIP* parameter specifies

			the number of connections that the specified server accepts from a single IP address. The default value is 2147483647. The possible values are from 1 through 2147483647.
MaxConnections	Optional	System.Int32	The *MaxConnections* parameter specifies the total number of connections that the specified server accepts. This includes authenticated and unauthenticated connections. The default value is 2147483647. The possible values are from 1 through 2147483647.
MaxConnectionsPerUser	Optional	System.Int32	The *MaxConnectionsPerUser* parameter specifies the maximum number of connections that the Client Access server accepts from a particular user. The default value is 16. The possible values are from 1 through 2147483647.

MessageRetrievalMimeFormat	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MimeTextFormat	The *MessageRetrievalMimeFormat* parameter specifies the format of the messages retrieved from the server. The default value is `BestBodyFormat`. You can specify the value for this parameter by using a numerical value or text string. The possible values are: • 0 or `TextOnly` • 1 or `HtmlOnly` • 2 or `HtmlAndTextAlternative` • 3 or `TextEnrichedOnly` • 4 or `TextEnrichedAndTextAlternative` • 5 or `BestBodyFormat` • 6 or `Tnef`
MessageRetrievalSortOrder	Optional	Microsoft.Exchange.Data.SortOrder	The *MessageRetrievalSortOrder* parameter specifies the order in which the retrieved messages are sorted. This value can be either 0 (Ascending) or 1 (Descending). The default value is `Ascending`.
OwaServerUrl	Optional	System.Uri	The *OwaServerUrl* parameter specifies the Client Access server from which to retrieve calendar

			information for instances of custom Microsoft Office Outlook Web App calendar items.
PreAuthenticatedConnectionTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *PreAuthenticatedConnectionTimeout* parameter specifies the time to wait before closing an idle connection that isn't authenticated. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Valid input for this parameter is `00:00:30` to `1:00:00`.The default value is `00:1:00` or one minute.
ProtocolLogEnabled	Optional	System.Boolean	The *ProtocolLogEnabled* parameter specifies whether to enable protocol logging. The default setting is `$false`. For more information, see Configure protocol logging for POP3 and IMAP4.
ProxyTargetPort	Optional	System.Int32	The *ProxyTargetPort* parameter specifies the

			port on the Exchange Server 2003 back-end server to which the Microsoft Exchange POP3 service on a Client Access server relays commands. The default port is `9955`.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies an individual Client Access server in your organization for which you're specifying POP3 settings.
SSLBindings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SSLBindings* parameter specifies whether the command sets the IP address and port number to use for an SSL session. This is a multivalued property. Enter a value using the format: <*IP address*>:<*Port number*>. The default value is `0.0.0.0:995`.
SuppressReadReceipt	Optional	System.Boolean	The *SuppressReadReceipt* parameter specifies whether to stop duplicate read receipts from being sent to POP3 senders that are using the **Send read**

			receipts for messages I send option in their POP3 email program. By default, this option is set to `$false`. By default, POP3 senders that use the **Send read receipts for messages I send** option receive a read receipt in both of the following circumstances: • When messages they send are downloaded by the recipient. • When the recipient opens the message. The valid values and descriptions for this parameter are: • `$false` POP3 users are sent a read receipt each time a recipient downloads a message. POP3 users are also sent a read receipt when the user opens the message. • `$true` POP3 users that use the **Send read receipts for messages I send** option in their email client programs only receive a read receipt when the recipient opens the message.

UnencryptedOrTLSBindings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UnencryptedOrTLSBindings* parameter specifies the IP address and port number for communication over the TLS–encrypted connection or the connection that isn't encrypted. This is a multivalued property. Enter a value using the format: <*IP address*>:<*Port number*>. The default value is `0.0.0.0:110`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
X509CertificateName	Optional	System.String	The *X509CertificateName* parameter specifies the host name in the SSL certificate from the **Associated Subject** field.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-PowerShellConnectivity

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-28*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-PowerShellConnectivity** cmdlet to test whether Windows PowerShell remoting on the target Client Access server is functioning correctly.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-PowerShellConnectivity -ConnectionUri <Uri> -TestCredential
<PSCredential> <COMMON PARAMETERS>


Test-PowerShellConnectivity [-ClientAccessServer <ServerIdParameter>] [-
TestType <Internal | External>] [-VirtualDirectoryName <String>] <COMMON
PARAMETERS>


COMMON PARAMETERS: [-Authentication <Default | Basic | Negotiate |
NegotiateWithImplicitCredential | Credssp | Digest | Kerberos>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-MailboxServer
<ServerIdParameter>] [-MonitoringContext <SwitchParameter>] [-
ResetTestAccountCredentials <SwitchParameter>] [-TrustAnySSLCertificate
<SwitchParameter>] [-UserType <LEGACY | EDU | BPOS>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the PowerShell (Default Web Site) virtual directory on the CAS2 server. The *TrustAnySSLCertificate* switch is used to skip the certificate check during connection. You might need to use the New-TestCasConnectivityUser.ps1 script to create a test user that the command can use

to test the Windows PowerShell virtual directory.

```
Test-PowerShellConnectivity -ClientAccessServer CAS2 -
VirtualDirectoryName "PowerShell (Default Web Site)" -
TrustAnySSLCertificate
```

EXAMPLE 2

This example tests the remote Windows PowerShell virtual directory at the URI https://contoso.com/powershell. Because the SSL certificate should be valid, the *TrustAnySSLCertificate* switch isn't used. The remote server uses Basic authentication so the *Authentication* parameter is used with a value of `Basic`.

Before running the test, the credentials to be used to connect to the remote Windows PowerShell virtual directory need to be retrieved. The following command retrieves the credentials from the person running the test using the **Get-Credential** cmdlet and stores them in the *$UserCredentials* variable.

```
$UserCredentials = Get-Credential
```

The test is then run using the **Test-PowerShellConnectivity** cmdlet with the options previously described.

```
Test-PowerShellConnectivity -ConnectionUri https://
contoso.com/powershell -TestCredential $UserCredentials -
Authentication Basic
```

Detailed Description

The **Test-PowerShellConnectivity** cmdlet connects to a Client Access server to test whether Windows PowerShell remoting on that server is working correctly and whether the Client Access server can perform commands against a remote Mailbox server.

When you run the **Test-PowerShellConnectivity** cmdlet, you must specify either the fully qualified domain name (FQDN) of the Client Access server to connect to by using the *ClientAccessServer* parameter, or the Uniform Resource Identifier (URI) of a Client Access server by using the *ConnectionUri* parameter. You can't use both the *ClientAccessServer* and *ConnectionUri* parameters in the same command.

The first time you use the **Test-PowerShellConnectivity** cmdlet with the *ClientAccessServer* parameter, you might be required to create a test user. To create a test user, use the New-TestCasConnectivityUser.ps1 script.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "PowerShell virtual directories" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
ConnectionUri	Required	System.Uri	The *ConnectionUri* parameter specifies the URI of the Client Access server to test, for example, https://contoso.com/powershell. If the *ClientAccessServer* parameter is specified, you can't use the *ConnectionUri* parameter.
TestCredential	Required	System.Management.Automation.PSCredential	The *TestCredential* parameter specifies the credentials to use when connecting to the Client Access server. This parameter can only be used when the *ConnectionUri* parameter is specified. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.

Authentication	Optional	System.Management.Automation.Runspaces.AuthenticationMechanism	The *Authentication* parameter specifies the type of authentication to use when establishing a connection. You can use one of the following values: • `Default` • `Basic` • `Negotiate` • `NegotiateWithImplicitCredential` • `Credssp` • `Digest` • `Kerberos` **Note:** The `NegotiateWithImplicitCredential` value has been deprecated and is no longer used.
ClientAccessServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *ClientAccessServer* parameter specifies the name of the Client Access server to test. If the *ConnectionUri* parameter is specified, you can't use the *ClientAccessServer* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a

			value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
MailboxServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *MailboxServer* parameter specifies the name of the Mailbox server to test. If not specified, all Mailbox servers in the local Active Directory site are tested.
MonitoringContext	Optional	System.Management.Automation.SwitchParameter	The *MonitoringContext* switch includes the associated monitoring events and performance counters in the results. You don't need to specify a value with this switch. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 -

			Operations Manager.
ResetTestAccountCredentials	Optional	System.Management.Automation.SwitchParameter	The *ResetTestAccountCredentials* switch resets the password for the test account that's used to run this command. The password for the test account is typically reset every seven days. Use this switch to force a password reset any time it's required for security reasons.
TestType	Optional	Microsoft.Exchange.Monitoring.OwaConnectivityTestType	The *TestType* parameter specifies whether the command should test the internal or external URL of a virtual directory. This parameter can only be used with the *ClientAccessServer* parameter. The valid values are `Internal` and `External`. The default value is `Internal`.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	The *TrustAnySSLCertificate* parameter specifies whether Secure Sockets Layer (SSL) certificate

			validation failures are reported. This parameter instructs the command to check Windows PowerShell connectivity without generating an error when the certificate doesn't match the URL of the Client Access server.
UserType	Optional	Microsoft.Exchange.Monitoring.DatacenterUserType	This parameter is reserved for internal Microsoft use.
VirtualDirectoryName	Optional	System.String	The *VirtualDirectoryName* parameter specifies the virtual directory on the Client Access server to test. The *VirtualDirectoryName* parameter can only be used with the *ClientAccessServer* parameter. If the *VirtualDirectoryName* parameter isn't specified, all virtual directories on the Client Access server are tested.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PowerShellVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-PowerShellVirtualDirectory** cmdlet to view an existing virtual directory in Internet Information Services (IIS).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PowerShellVirtualDirectory -Server <ServerIdParameter> <COMMON
PARAMETERS>


Get-PowerShellVirtualDirectory [-Identity <VirtualDirectoryIdParameter>]
<COMMON PARAMETERS>


COMMON PARAMETERS: [-ADPropertiesOnly <SwitchParameter>] [-
DomainController <Fqdn>] [-ShowBackEndVirtualDirectories
<SwitchParameter>] [-ShowMailboxVirtualDirectories <SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves a list of all the Windows PowerShell virtual directories and uses the **Format-Table** cmdlet to display only the *Name* property and any properties that begin with `Internal`. For more information about the **Format-Table** cmdlet, see Working with command output.

```
Get-PowerShellVirtualDirectory | Format-Table Name,
Internal* -Wrap
```

EXAMPLE 2

This example retrieves a list of all the Windows PowerShell virtual directories that exist on the server Server01.

```
Get-PowerShellVirtualDirectory -Server Server01
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "PowerShell virtual directories" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies which server to query when returning a list of Windows PowerShell virtual directories. You can't use the *Server* parameter with the *Identity* parameter.
ADPropertiesOnly	Optional	System.Management.A	The *ADPropertiesOnly*

		utomation.SwitchParameter	switch specifies whether to return only the properties about the virtual directory stored in Active Directory. The properties stored in the Internet Information Services (IIS) metabase aren't returned.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name of the Windows PowerShell virtual directory to retrieve. If the virtual directory contains spaces, enclose the name in quotation marks ("). You can't use the *Identity* parameter with the *Server* parameter.
ShowBackEndVirtualDirectories	Optional	System.Management.Automation.SwitchPara	This parameter is reserved for internal

		meter	Microsoft use.
ShowMailboxVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowMailboxVirtualDirectories* switch instructs the command to return the Windows PowerShell virtual directories that are located on servers running the Mailbox server role. This switch should only be used with the direction of Microsoft Support.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-PowerShellVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-PowerShellVirtualDirectory** cmdlet to create a Windows PowerShell virtual directory in Internet Information Services (IIS).

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-PowerShellVirtualDirectory -Name <String> [-AppPoolId <String>] [-
BasicAuthentication <$true | $false>] [-CertificateAuthentication <$true |
$false>] [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
ExternalUrl <Uri>] [-InternalUrl <Uri>] [-LimitMaximumMemory <$true |
$false>] [-Path <String>] [-RequireSSL <$true | $false>] [-Role
<ClientAccess | Mailbox>] [-Server <ServerIdParameter>] [-WebSiteName
<String>] [-WhatIf [<SwitchParameter>]] [-WindowsAuthentication <$true |
$false>]
```

Examples

EXAMPLE 1

This example creates a Windows PowerShell virtual directory and configures it to accept only certificate authentication.

```
New-PowerShellVirtualDirectory -Name "Contoso Certificates
Required" -BasicAuthentication $false -
WindowsAuthentication $false -CertificateAuthentication
$true
```

Detailed Description

Although it's possible to create a Windows PowerShell virtual directory, we recommend that you only do so at the request of Microsoft Customer Service and Support.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "PowerShell virtual directories" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name of the new Windows PowerShell virtual directory. The name you provide will have the name of the website it's created under

			appended to it. If the name you provide contains spaces, enclose the name in quotation marks (").
AppPoolId	Optional	System.String	The *AppPoolId* parameter specifies the pool of programs that can be used with the Windows PowerShell virtual directory.
BasicAuthentication	Optional	System.Boolean	The *BasicAuthentication* parameter specifies whether Basic authentication is enabled on the Windows PowerShell virtual directory. The valid values are `$true` and `$false`. The default value is `$true`.
CertificateAuthentication	Optional	System.Boolean	The *CertificateAuthentication* parameter specifies whether certificate authentication is enabled on the Windows PowerShell virtual directory. The valid values are `$true` and `$false`. The default value is `$false`.
Confirm	Optional	System.Management.	The *Confirm* switch

		Automation.SwitchParameter	causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the external URL that the Windows PowerShell virtual directory points to.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the internal URL that the Windows PowerShell virtual directory points to.
LimitMaximumMemory	Optional	System.Boolean	The *LimitMaximumMemory* parameter specifies whether to limit the

			amount of memory used by the application pool in the Windows PowerShell virtual directory. If this parameter is set to `$true`, the amount of memory made available to the application pool is limited to 70 percent of the total physical memory in the server. If this parameter is set to `$false`, the amount of memory isn't limited. The default value is `$false`.
Path	Optional	System.String	The *Path* parameter specifies the directory that contains the system files for the Windows PowerShell virtual directory.
RequireSSL	Optional	System.Boolean	The *RequireSSL* parameter specifies whether the Windows PowerShell virtual directory should require that the client connection be made using Secure Sockets Layer (SSL). The valid values are `$true` and `$false`. The default value is `$true`.

Role	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.VirtualDirectoryRole	The *Role* parameter species the configuration that should be used when the virtual directory is created. The following are the values that can be used with this parameter: • `FrontEnd` Configure the virtual directory for use on a Client Access server. • `BackEnd` Configure the virtual directory for use on a Mailbox server.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server name on which the new Windows PowerShell virtual directory should be created.
WebSiteName	Optional	System.String	The *WebSiteName* parameter specifies the name of the IIS website under which the Windows PowerShell virtual directory is created.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsAuthentication	Optional	System.Boolean	The *WindowsAuthentication* parameter specifies whether Integrated Windows authentication is enabled on the Windows PowerShell virtual directory. The valid values are `$true` and `$false`. The default value is `$true`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-PowerShellVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-PowerShellVirtualDirectory** cmdlet to remove an existing Windows PowerShell

virtual directory from Internet Information Services (IIS).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-PowerShellVirtualDirectory -Identity <VirtualDirectoryIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes a Windows PowerShell virtual directory without confirmation.

Caution:
Be careful when using the **Remove-PowerShellVirtualDirectory** cmdlet without confirmation. You won't be prompted before the virtual directory is deleted.

```
Remove-PowerShellVirtualDirectory "Internal (Default Web
Site)" -Confirm:$False
```

Detailed Description

Although it's possible to remove a Windows PowerShell virtual directory, we recommend that you only do so at the request of Microsoft Customer Service and Support.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "PowerShell virtual directories" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the Windows PowerShell virtual directory to remove. If the name contains spaces, enclose the name in quotation marks (").

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-PowerShellVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-PowerShellVirtualDirectory** cmdlet to change an existing Windows PowerShell virtual directory in Internet Information Services (IIS).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-PowerShellVirtualDirectory -Identity <VirtualDirectoryIdParameter> [-
BasicAuthentication <$true | $false>] [-CertificateAuthentication <$true |
$false>] [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
EnableCertificateHeaderAuthModule <$true | $false>] [-
EnableDelegatedAuthModule <$true | $false>] [-
EnableSessionKeyRedirectionModule <$true | $false>] [-ExternalUrl <Uri>]
[-InternalUrl <Uri>] [-LiveIdBasicAuthentication <$true | $false>] [-
LiveIdNegotiateAuthentication <$true | $false>] [-RequireSSL <$true |
$false>] [-WhatIf [<SwitchParameter>]] [-WindowsAuthentication <$true |
$false>]
```

Examples

EXAMPLE 1

This example modifies the external URL of the Contoso Windows PowerShell virtual directory.

```
Set-PowerShellVirtualDirectory "Contoso (default Web site)"
-ExternalUrl "http://www.contoso.com/powershell"
```

Detailed Description

Although it's possible to modify a Windows PowerShell virtual directory, we recommend that you only do so at the request of Microsoft Customer Service and Support.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "PowerShell virtual directories" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name of the Windows PowerShell virtual directory that you want to modify.
BasicAuthentication	Optional	System.Boolean	The *BasicAuthentication* parameter specifies whether Basic authentication is enabled on the Windows PowerShell virtual directory. The valid values are `$true` and `$false`. The default value is `$true`.
CertificateAuthentication	Optional	System.Boolean	The *CertificateAuthentication* parameter specifies whether certificate authentication is enabled on the

			Windows PowerShell virtual directory. The valid values are `$true` and `$false`. The default value is `$false`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EnableCertificateHeaderAuthModule	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
EnableDelegatedAuthModule	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.

EnableSessionKeyRedirectionModule	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the external URL that the Windows PowerShell virtual directory points to.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the internal URL that the Windows PowerShell virtual directory points to.
LiveIdBasicAuthentication	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LiveIdNegotiateAuthentication	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RequireSSL	Optional	System.Boolean	The *RequireSSL* parameter specifies whether the Windows PowerShell virtual directory should require that the client connection be made using Secure Sockets Layer (SSL). The valid values are `$true` and

			`$false`. The default value is `$true`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsAuthentication	Optional	System.Boolean	The *WindowsAuthentication* parameter specifies whether Integrated Windows authentication is enabled on the Windows PowerShell virtual directory. The valid values are `$true` and `$false`. The default value is `$true`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-PushNotificationProxy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-PushNotificationProxy** cmdlet to disable the push notification proxy that's configured between an on-premises Microsoft Exchange organization and a Microsoft Office 365 organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-PushNotificationProxy [-BackOffTimeInSeconds <Int32>] [-Confirm
[<SwitchParameter>]] [-DisplayName <String>] [-DomainController <Fqdn>] [-
Enabled <$true | $false>] [-ExchangeMaximumVersion <Version>] [-
ExchangeMinimumVersion <Version>] [-NumberOfChannels <Int32>] [-QueueSize
<Int32>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example disables the push notification proxy in the on-premises Exchange organization.

```
Disable-PushNotificationProxy
```

Detailed Description

The push notification proxy relays event notifications (for example, new email or calendar updates) for on-premises mailboxes through Office 365 to OWA for Devices on the user's device.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Push notification proxy settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
BackOffTimeInSeconds	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DisplayName	Optional	System.String	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge

			Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ExchangeMaximumVersion	Optional	System.Version	This parameter is reserved for internal Microsoft use.
ExchangeMinimumVersion	Optional	System.Version	This parameter is reserved for internal Microsoft use.
NumberOfChannels	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
QueueSize	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-PushNotificationProxy

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-PushNotificationProxy** cmdlet to enable a push notification proxy between an on-premises Microsoft Exchange organization and a Microsoft Office 365 organization.

Important:
In order for event notifications to be successfully delivered, you also need to configure OAuth authentication between your on-premises Exchange organization and your Office 365 organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-PushNotificationProxy [-BackOffTimeInSeconds <Int32>] [-Confirm
[<SwitchParameter>]] [-DisplayName <String>] [-DomainController <Fqdn>] [-
Enabled <$true | $false>] [-ExchangeMaximumVersion <Version>] [-
ExchangeMinimumVersion <Version>] [-NumberOfChannels <Int32>] [-
Organization <String>] [-QueueSize <Int32>] [-Uri <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example displays the status of the push notification proxy in the on-premises Exchange organization.

```
Enable-PushNotificationProxy -WhatIf
```

Example 2

This example enables the push notification proxy in the on-premises Exchange organization by using the Office 365 organization contoso.com.

```
Enable-PushNotificationProxy -Organization contoso.com
```

Detailed Description

The push notification proxy relays event notifications (for example, new email or calendar updates) for on-premises mailboxes through Office 365 to OWA for Devices on the user's device.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Push notification proxy settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
BackOffTimeInSeconds	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing

			continues. You don't have to specify a value with the *Confirm* switch.
DisplayName	Optional	System.String	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ExchangeMaximumVersion	Optional	System.Version	This parameter is reserved for internal Microsoft use.

ExchangeMinimumVersion	Optional	System.Version	This parameter is reserved for internal Microsoft use.
NumberOfChannels	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
Organization	Optional	System.String	The *Organization* parameter specifies the domain name of the Office 365 organization. For example, `contoso.com`.
QueueSize	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
Uri	Optional	System.String	The *Uri* parameter specifies the push notification service endpoint in Office 365. The default value is `https://outlook.office365.com/PushNotifications`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-RpcClientAccess

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-RpcClientAccess** cmdlet to display the settings for the Exchange RPC Client Access service that's running on a Microsoft Exchange server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-RpcClientAccess [-DomainController <Fqdn>] [-Server
<ServerIdParameter>]
```

Examples

EXAMPLE 1

This example displays the parameters and values for the Exchange server ENT01 that's running the Exchange RPC Client Access service.

```
Get-RpcClientAccess -Server ENT01
```

Detailed Description

The **Get-RpcClientAccess** cmdlet returns the Exchange RPC Client Access service configuration for Exchange servers. Information can be returned only about servers located in the Exchange organization from which the cmdlet is run.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "RPC Client Access settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-RpcClientAccess

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-RpcClientAccess** cmdlet to manage the settings for the Exchange RPC Client Access service that's running on a Microsoft Exchange Server 2010 Client Access server.

```
Set-RpcClientAccess -Server <ServerIdParameter> [-BlockedClientVersions
<String>] [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
EncryptionRequired <$true | $false>] [-MaximumConnections <Int32>] [-Name
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example restricts clients that aren't running Office Outlook 2007 from connecting to the Client Access server CAS01.

Important:
The values used with the *BlockedClientVersions* parameter are examples. You can determine the correct client software versions by parsing the RPC Client Access log files located at `%ExchangeInstallPath%Logging\RPC Client Access`.

```
Set-RpcClientAccess -Server CAS01 -BlockedClientVersions
"0.0.0-5.6535.6535;7.0.0;8.02.4-11.6535.6535"
```

Detailed Description

You can run the **Set-RpcClientAccess** cmdlet for a single Client Access server that has the Exchange RPC Client Access service installed or for all Exchange Client Access servers that have the Exchange RPC Client Access service installed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "RPC Client Access settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Client Access server.
BlockedClientVersions	Optional	System.String	The *BlockedClientVersions* parameter specifies which versions of Microsoft Outlook are restricted from connecting. The Exchange RPC Client Access service rejects Outlook connections if versions are in the range specified. This setting affects MAPI and Outlook Anywhere client connections. The value must be less than 256 characters in length. Versions should be single numbers in the format *X.Y.Z* where *X* is a major version number, *Y* is the minor revision number, and *Z* specifies the build, and ranges should be delimited by semicolons (for example, 0.0.0-5.9.9;

			7.0.0-65535.65535.65535). For more information, see Configure Outlook client blocking.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EncryptionRequired	Optional	System.Boolean	The *EncryptionRequired* parameter specifies whether to require Outlook connections to be encrypted. The Exchange RPC Client Access service rejects unencrypted Outlook

			connections if this parameter is set to `$true`.
MaximumConnections	Optional	System.Int32	The *MaximumConnections* parameter specifies the maximum number of concurrent connections allowed. The Exchange RPC Client Access service reads and limits connections based on this property. This parameter has a range from 1 through 65535. **Note:** Although you can configure a non-default value for this parameter, changes to this setting aren't enforced in this version of Exchange.
Name	Optional	System.String	The *Name* parameter specifies the name of the configuration object in Active Directory. By default, this parameter is set to RpcClientAccess.
WhatIf	Optional	System.Management.Automation.SwitchPara	The *WhatIf* switch instructs the command

		meter	to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SmimeConfig

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-SmimeConfig** cmdlet to view the S/MIME configuration for Microsoft Outlook Web App.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-SmimeConfig [-Identity <OrganizationIdParameter>] [-DomainController
<Fqdn>]
```

Examples

Example 1

This example shows the S/MIME configuration that used with Outlook Web App.

```
Get-SmimeConfig
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "S/MIME configuration" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance

			of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-SmimeConfig

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-SmimeConfig** cmdlet to modify the S/MIME configuration for Microsoft Outlook Web App.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-SmimeConfig [-Identity <OrganizationIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Name <String>] [-
OWAAllowUserChoiceOfSigningCertificate <$true | $false>] [-
OWAAlwaysEncrypt <$true | $false>] [-OWAAlwaysSign <$true | $false>] [-
OWABCCEncryptedEmailForking <UInt32>] [-OWACheckCRLOnSend <$true |
$false>] [-OWAClearSign <$true | $false>] [-OWACopyRecipientHeaders <$true
| $false>] [-OWACRLConnectionTimeout <UInt32>] [-OWACRLRetrievalTimeout
<UInt32>] [-OWADisableCRLCheck <$true | $false>] [-OWADLExpansionTimeout
<UInt32>] [-OWAEncryptionAlgorithms <String>] [-OWAEncryptTemporaryBuffers
<$true | $false>] [-OWAForceSMIMEClientUpgrade <$true | $false>] [-
```

```
OWAIncludeCertificateChainAndRootCertificate <$true | $false>] [-
OWAIncludeCertificateChainWithoutRootCertificate <$true | $false>] [-
OWAIncludeSMIMECapabilitiesInMessage <$true | $false>] [-
OWAOnlyUseSmartCard <$true | $false>] [-
OWASenderCertificateAttributesToDisplay <String>] [-
OWASignedEmailCertificateInclusion <$true | $false>] [-
OWASigningAlgorithms <String>] [-OWATripleWrapSignedEncryptedMail <$true |
$false>] [-OWAUseKeyIdentifier <$true | $false>] [-
OWAUseSecondaryProxiesWhenFindingCertificates <$true | $false>] [-
SMIMECertificateIssuingCA <Byte[]>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example sets the S/MIME configuration to allow users the choice of signing the message, limits the Certificate Revocation List (CRL) retrieval time-out to 10 seconds, and specifies the 128 bit RC2 encryption algorithm.

```
Set-SmimeConfig -OWAAllowUserChoiceOfSigningCertificate
$true -OWACRLRetrievalTimeout 10000 -
OWAEncryptionAlgorithms 6602:128
```

Detailed Description

⚠Warning:

The **Set-SmimeConfig** cmdlet can change several important parameters than can reduce the overall level of message security. Review your organization's security policy before you make any changes.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "S/MIME configuration" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is reserved for internal Microsoft use.

Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
Name	Optional	System.String	This parameter is reserved for internal Microsoft use.
OWAAllowUserChoiceOfSigningCertificate	Optional	System.Boolean	The *OWAAllowUserChoiceOfSigningCertificate* parameter specifies whether to allow users to select the certificate to use when they digitally sign email messages in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
OWAAlwaysEncrypt	Optional	System.Boolean	The *OWAAlwaysEncrypt* parameter specifies whether all outgoing messages are automatically encrypted in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default

			value is `$false`.
OWAAlwaysSign	Optional	System.Boolean	The *OWAAlwaysSign* parameter specifies whether all outgoing messages are automatically signed in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
OWABCCEncryptedEmailForking	Optional	System.UInt32	The *OWABCCEncryptedEmailForking* parameter specifies how Bcc messages are encrypted in Outlook Web App. This parameter uses the following values: • 0 = One encrypted message per Bcc recipient. • 1 = One single encrypted message for all Bcc recipients. • 2 = One encrypted message without Bcc forking. The default value is 0. **Note:** This setting affects the

			security and privacy of Outlook Web App. Consult your organization's security policy before you change this setting.
OWACheckCRLOnSend	Optional	System.Boolean	The *OWACheckCRLOnSend* parameter specifies how the certificate revocation list (CRL) check is enforced when an email message is sent in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When this parameter is set to `$false` and the CRL distribution point is inaccessible, Outlook Web App allows signed or encrypted messages to be sent. When this parameter is set to `$true`, Outlook Web App displays a warning dialog box and prevents signed or encrypted messages from being sent.

OWAClearSign	Optional	System.Boolean	The *OWAClearSign* parameter specifies how email messages are signed in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default is `$true`. When this parameter is set to `$true`, digitally signed messages are clear-signed. When this parameter is set to `$false`, digitally signed messages are opaque-signed. Clear-signed messages are larger than opaque-signed messages, but clear-signed messages can be read in most email clients, including clients that don't support S/MIME.
OWACopyRecipientHeaders	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
OWACRLConnectionTimeout	Optional	System.UInt32	The *OWACRLConnectionTimeout* parameter specifies the time in

			milliseconds that Outlook Web App waits while connecting to retrieve a single CRL as part of a certificate validation operation. Valid input for this parameter is an integer between 0 and 4294967295 (`UInt32`). The default value is 60000 (60 seconds). When multiple CRLs in a certificate chain must be retrieved, the time limit that's specified by this parameter applies to each connection. For example, if a certificate requires the retrieval of three CRLs, and this parameter is set to 60000 (60 seconds), each individual CRL retrieval operation has a time limit of 60 seconds. If any one of the CRLs isn't retrieved before the time limit expires, the entire operation fails. The total time limit for all

			the retrievals is controlled by the *OWACRLRetrievalTimeout* parameter.
OWACRLRetrievalTimeout	Optional	System.UInt32	The *OWACRLRetrievalTimeout* parameter specifies the time in milliseconds that Outlook Web App waits to retrieve all CRLs when validating a certificate. Valid input for this parameter is an integer between 0 and 4294967295 (`UInt32`). The default value is 10000 (10 seconds). If all the required CRLs are not retrieved before the time limit expires, the operation fails. Suppose the retrieval of three CRLs is required, the *OWACRLConnectionTimeout* value is set to 60000 (60 seconds), and the *OWACRLRetrievalTimeout* is set to 120000 (2 minutes). In this

			example, if any individual CRL retrieval takes more than 60 seconds, the operation fails. Also, if all the CRL retrievals together take more than 120 seconds, the operation fails.
OWADisableCRLCheck	Optional	System.Boolean	The *OWADisableCRLCheck* parameter enables or disables CRL checking in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When set to `$true`, this parameter disables CRL checks when validating certificates. Disabling CRL checking can decrease the time that's required to validate the signatures of signed email messages, but it also validates email messages signed with revoked certificates.
OWADLExpansionTimeout	Optional	System.UInt32	The *OWADLExpansionTime*

			out parameter specifies the time in milliseconds that Outlook Web App waits when sending encrypted messages to members of a distribution group that requires expansion. Valid input for this parameter is an integer between 0 and 4294967295 (`UInt32`). The default value is 60000 (60 seconds). If the operation doesn't complete in the time specified by this parameter, the operation fails and the message is not sent. When sending an encrypted message to a distribution group, Exchange expands the distribution group to retrieve the encryption certificate of each recipient. While the distribution group is being expanded, the sender receives no response from Outlook Web App.

			The timeout value that's specified by this parameter is applied to the expansion of each distribution group. For example, if an encrypted message is sent to three distribution group, and the value of this parameter is 60000 (60 seconds), the entire operation can take no more than 180 seconds.
OWAEncryptionAlgorithms	Optional	System.String	The *OWAEncryptionAlgorithms* parameter specifies a list of algorithms that are used by Outlook Web App to encrypt messages. Valid input for this parameter is a semicolon-separated list of symmetric encryption algorithm identifiers. When you use an algorithm that supports multiple key lengths, you need to

			specify the key length. Note that RC2 is the only supported algorithm that that offers multiple key lengths. You can specify the object identifier (OID) of the cryptographic service provider (CSP) when using third-party CSPs. An OID must be specified together with an algorithm ID. Outlook Web App needs an algorithm ID so that it can infer how the algorithm should be used. For example, to provide a custom replacement for the 3DES algorithm, you would specify the algorithm ID of 3DES (6603) and the custom OID of the replacement algorithm by using the value `6603,<OID>`. The encryption algorithms, key length values, and algorithm IDs that you can use with this parameter are

			described in the following list: • RC2 algorithm ID `6602` (supported key lengths are 40, 56, 64, and 128) • DES (56-bit) algorithm ID `6601` • 3DES (168-bit) algorithm ID `6603` • AES128 algorithm ID `660E` • AES192 algorithm ID `660F` • AES256 algorithm ID `6610` This parameter uses the following syntax: `{Algorithm ID} |` For example, to set the encryption algorithms to 3DES, RC2-128, RC2-64, DES, and RC2-56, use the following value: `6603;6602:128;6602:64;6601;6602:56`. The algorithm specified by *OWAEncryptionAlgorithms* is always used. If the parameter is not

			specified or is not formatted correctly, Outlook Web App uses the default value `6610` (AES256). If the encryption algorithm or minimum key length is not available on a client, Outlook Web App does not allow encryption.
OWAEncryptTemporaryBuffers	Optional	System.Boolean	The *OWAEncryptTemporaryBuffers* parameter specifies whether the Outlook Web App client-side temporary message storage buffers are encrypted. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, all client-side temporary buffers that store message data are encrypted using an ephemeral key and the 3DES algorithm. Setting this parameter to `$false` disables temporary

			buffer encryption. **Note:** Disabling encryption of the buffers can increase performance of the Outlook Web App client but also leaves information unencrypted in the client's buffer. Consult your organization's security policy before you disable this feature.
OWAForceSMIMEClient Upgrade	Optional	System.Boolean	The *OWAForceSMIMEClient Upgrade* parameter specifies whether or not users are forced to upgrade an S/MIME control that's older than their current version in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default is `$true`. If the parameter is set to `$true`, users need to download and install the new control before they can use S/MIME. If this parameter is set to `$false`, users receive a

			warning if the S/MIME control on their computer is not current, but they can still use S/MIME without updating the control.
OWAIncludeCertificateChainAndRootCertificate	Optional	System.Boolean	The *OWAIncludeCertificateChainAndRootCertificate* parameter specifies whether the certificate chains and root certificates of the signing or encryption certificates are included in the message in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
OWAIncludeCertificateChainWithoutRootCertificate	Optional	System.Boolean	The *OWAIncludeCertificateChainWithoutRootCertificate* parameter specifies whether the certificate chains of the signing or encryption certificates are included in messages in Outlook Web App.

			Valid input for this parameter is `$true` or `$false`. The default is `$false`. By default, Outlook Web App includes only the signing and encrypting certificates, not their corresponding certificate chains. When this parameter is set to `$true`, signed or encrypted messages include the full certificate chain, but not the root certificate.
OWAIncludeSMIMECapabilitiesInMessage	Optional	System.Boolean	The *OWAIncludeSMIMECapabilitiesInMessage* parameter specifies whether signed and encrypted messages in Outlook Web App include attributes that describe the supported encryption and signing algorithms. Valid input for this parameter is `$true` or `$false`. The default is `$false`. Enabling this option

			increases the size of messages, but may make it easier for some email clients to interact with encrypted messages in Outlook Web App.
OWAOnlyUseSmartCard	Optional	System.Boolean	The *OWAOnlyUseSmartCard* parameter specifies whether smartcard-based certificates are required for Outlook Web App message signing and decryption. Valid input for this parameter is `$true` or `$false`. The default is `$false` When this parameter is set to `$true`, the use of smartcard-based certificates for signing and decryption is required when you use Outlook Web App and the S/MIME control.
OWASenderCertificateAttributesToDisplay	Optional	System.String	The *OWASenderCertificateAttributesToDisplay* parameter controls which certificate

			attributes are displayed when signature verification proceeds despite a mismatch between the sender's email address and the email address in sender's certificate. The parameter accepts a comma-separated list of object identifiers (OIDs). This setting is blank (`$null`) by default.
OWASignedEmailCertificateInclusion	Optional	System.Boolean	The *OWASignedEmailCertificateInclusion* parameter specifies whether the sender's encryption certificate is excluded from a signed email message in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default is `$true`. By default, Outlook Web App and the S/MIME control include both signing and encrypting certificates

			with signed email messages. When this parameter is set to `$false`, the size of encrypted messages is reduced. However, recipients don't have access to the sender's encryption certificate in the message. Recipients need to retrieve the certificate from a directory, or from the sender.
OWASigningAlgorithms	Optional	System.String	The *OWASigningAlgorithms* parameter specifies the list of signing algorithms that are used by Outlook Web App to sign messages with the S/MIME control. Valid input for this parameter is a semicolon-separated list of symmetric encryption algorithm identifiers. You can specify the object identifier (OID) of the cryptographic

			service provider (CSP) when using third-party CSPs. An OID must be specified together with an algorithm ID. Outlook Web App needs an algorithm ID so that it can infer how the algorithm should be used. For example, to provide a custom replacement for the SHA1 algorithm, you would specify the algorithm ID of SHA1 (8804) and the custom OID of the replacement algorithm by using the value 8804,<OID>. This parameter supports the following algorithms. • **CALG_SHA_512** Type: 512 bit secure hashing algorithm (SHA). Algorithm ID: 800E. • **CALG_SHA_384** Type: 384 bit SHA. Algorithm ID: 800D. • **CALG_SHA_256** Type: 256 bit SHA. Algorithm ID: 800C.

			• **SHA1** Type: SHA. Algorithm ID: 8004. • **CALG_MD5** Type: MD5 hashing algorithm. Algorithm ID: 8003. This parameter uses the following syntax: {Algorithm ID} | For example, to set the signing algorithms to CALG_SHA_512, SHA1, and CALG_MD5, use the value 800E;8004;8003. The algorithm specified by *OWASigningAlgorithms* is always used. If this parameter is not specified or is not formatted correctly, Outlook Web App defaults to 8004 (SHA1).
OWATripleWrapSignedEncryptedMail	Optional	System.Boolean	The *OWATripleWrapSignedEncryptedMail* parameter specifies whether signed and encrypted email messages in Outlook

			Web App are triple-wrapped. Valid input for this parameter is `$true` or `$false`. The default is `$true`. A *triple-wrapped* message is a signed message that is encrypted, and then the encrypted message is signed (signed-encrypted-signed). When this parameter is set to `$false`, the signed message is encrypted only (there is no additional signing of the encrypted message). Triple-wrapped messages offer the highest level of security for messages under the S/MIME standard, but are larger in size.
OWAUseKeyIdentifier	Optional	System.Boolean	The *OWAUseKeyIdentifier* parameter specifies whether a certificate's key identifier is used to encode the

			asymmetrically encrypted token in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default is `$false`. By default, Outlook Web App encodes the asymmetrically encrypted token (sometimes called a lockbox) that's required to decrypt the rest of the message by indicating the issuer and serial number of each recipient's certificate. The issuer and serial number can then be used to locate the certificate and private key for decrypting the message. This parameter causes the use of a certificate's key identifier when encoding the asymmetrically encrypted token. Because a key pair can

			be reused in new certificates, using the key identifier for encrypted email messages means that users need to keep only the most recent certificate and associated private key, rather than all old certificates. Because some email clients do not support finding certificates with a key identifier, Outlook Web App uses the issuer and serial number of each recipient's certificate by default.
OWAUseSecondaryProxiesWhenFindingCertificates	Optional	System.Boolean	The *OWAUseSecondaryProxiesWhenFindingCertificates* parameter specifies whether alternative proxies are used during the certificate search in Outlook Web App. Valid input for this parameter is `$true` or `$false`. The default is `$true`.

			Outlook Web App attempts to find the correct certificate for a recipient when sending encrypted messages. The certificate subject or subject alternative name values can each contain an email address. Because a recipient can have multiple proxy addresses, the certificate's subject or subject alternative name values may not match the recipient's primary SMTP address. When this parameter is set to `$true`, and the certificate subject or subject alternative name values do not match the recipient's primary SMTP address, Outlook Web App tries to match the certificate's subject to one of the recipient's proxy addresses.
SMIMECertificateIssuingCA	Optional	System.Byte[]	The *SMIMECertificateIssuingCA* parameter specifies

			the serialized certificate store (SST) that contains the Certificate Authority (CA) signing and intermediate certificate information. You need to read the file to a byte-encoded object using the **Get-Content** cmdlet. For example: `-SMIMECertificateIssuingCA $([byte[]](Get-Content -Encoding byte -Path "C:\Temp\CACertificateSerializedStore.sst" -ReadCount 0)` Each certificate is checked, and if any certificates are expired, the operation will fail.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Clear-TextMessagingAccount

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Clear-TextMessagingAccount** cmdlet to remove the text messaging settings from a user's account.

```
Clear-TextMessagingAccount -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreDefaultScope
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example clears the text messaging account settings and notification settings from Tony Smith's mailbox.

```
Clear-TextMessagingAccount -Identity "TonySmith"
```

EXAMPLE 2

This example clears the text messaging account settings and notification settings from Tony Smith's mailbox and displays a confirmation message.

```
Clear-TextMessagingAccount -Identity "Contoso\TonySmith" -
Confirm $true
```

EXAMPLE 3

This example clears the text messaging account settings and notification settings from Tony Smith's mailbox.

```
Clear-TextMessagingAccount -Identity "tony@contoso.com"
```

Detailed Description

The **Clear-TextMessagingAccount** cmdlet clears all of a user's text messaging settings, including communication and notification settings.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Text messaging settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the target mailbox. You can use one of the following values: • CommonName • DisplayName • FirstName • LastName • Alias
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before

			processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the

			DomainController parameter. The command uses an appropriate global catalog server automatically.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-TextMessagingAccount

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-TextMessagingAccount** cmdlet to return a user's Short Message Service (SMS) settings.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-TextMessagingAccount -Identity <MailboxIdParameter> [-Credential
<PSCredential>] [-DomainController <Fqdn>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns the text messaging settings for Tony Smith's mailbox using his alias.

```
Get-TextMessagingAccount -Identity "TonySmith" -
DomainController server.contoso.com
```

EXAMPLE 2

This example returns the text messaging settings for Tony Smith's mailbox using his domain and username.

```
Get-TextMessagingAccount -Identity "contoso\tonysmith" -
DomainController DC1.contoso.com
```

EXAMPLE 3

This example returns the text messaging settings for Tony Smith's mailbox using his email address.

```
Get-TextMessagingAccount -Identity "tony@contoso.com" -
DomainController gc.contoso.com
```

Detailed Description

The **Get-TextMessagingAccount** cmdlet displays the SMS settings for a specific user. These settings include whether Microsoft Exchange ActiveSync is enabled, the user's country or region ID, mobile operator ID, service provider ID, and notification phone number.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Text

messaging settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identity of the mailbox.
Credential	Optional	System.Management.Automation.PSCredential	This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	The *ReadFromDomainController* parameter specifies whether the cmdlet returns data from the domain controller.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the amount of data returned.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-TextMessagingAccount

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-TextMessagingAccount** cmdlet to configure text messaging notification settings for a user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-TextMessagingAccount -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-CountryRegionId <RegionInfo>] [-DomainController
<Fqdn>] [-IgnoreDefaultScope <SwitchParameter>] [-MobileOperatorId
<Int32>] [-NotificationPhoneNumber <E164Number>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the notification phone number for the text messaging account for Jeff Hay.

```
Set-TextMessagingAccount -Identity 'JeffHay' -
NotificationPhoneNumber 4255550100
```

EXAMPLE 2

This example sets the region, mobile operator, and notification phone number for the text messaging account for Jeff Hay.

```
Set-TextMessagingAccount -Identity 'JeffHay' -
CountryRegionId US -MobileOperatorId 15001 -
NotificationPhoneNumber +14255550199
```

Detailed Description

The **Set-TextMessagingAccount** cmdlet configures a user's account for text messaging notifications. You can configure several settings, including the mobile phone number and country or region ID.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Text messaging settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox identity.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CountryRegionId	Optional	System.Globalization.RegionInfo	The *CountryRegionId* parameter specifies the country or region in

			which the user's mobile operator resides.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter isn't yet implemented.
MobileOperatorId	Optional	System.Int32	The *MobileOperatorId* parameter specifies the mobile operator ID for the user.
NotificationPhoneNumber	Optional	Microsoft.Exchange.Data.Storage.Management.E164Number	The *NotificationPhoneNumber* parameter specifies the phone number to use for notifications.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Compare-TextMessagingVerificationCode

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Compare-TextMessagingVerificationCode** cmdlet to verify the text messaging verification code that the user specified.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Compare-TextMessagingVerificationCode -VerificationCode <String> [-
```

```
Identity <MailboxIdParameter>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example compares the verification code 111555 to the code sent to Tony Smith's mobile phone.

```
Compare-TextMessagingVerificationCode -Identity TonySmith -
VerificationCode 111555
```

EXAMPLE 2

This example compares the verification code 123456 to the code sent to Tony Smith's mobile phone.

```
Compare-TextMessagingVerificationCode -Identity
tony@contoso.com -VerificationCode 123456
```

EXAMPLE 3

This example compares the verification code 111555 to the code sent to Tony Smith's mobile phone after confirmation is given.

```
Compare-TextMessagingVerificationCode -Identity TonySmith -
VerificationCode 111555 -Confirm $true
```

Detailed Description

The **Compare-TextMessagingVerificationCode** cmdlet returns true if the code matches the stored code generated by the **Send-TextMessagingVerificationCode** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Text messaging settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
VerificationCode	Required	System.String	The *VerificationCode* parameter contains the

			verification code that the user specified.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox ID for the user.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Send-TextMessagingVerificationCode

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Send-TextMessagingVerificationCode** cmdlet to send a text messaging verification code to the user's mobile phone.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Send-TextMessagingVerificationCode [-Identity <MailboxIdParameter>] [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sends the text messaging verification code to Tony Smith's mobile phone.

```
Send-TextMessagingVerificationCode -Identity "TonySmith"
```

EXAMPLE 2

This example sends the text messaging verification code to Tony Smith's mobile phone after confirmation is given.

```
Send-TextMessagingVerificationCode -Identity "TonySmith" -
Confirm $true
```

EXAMPLE 3

This example sends the text messaging verification code to Tony Smith's mobile phone.

```
Send-TextMessagingVerificationCode -Identity
"tony@contoso.com"
```

Detailed Description

The **Send-TextMessagingVerificationCode** cmdlet generates a verification code and sends it to a user's mobile phone.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Text messaging settings" entry in the Clients and mobile devices permissions topic.

Caution:
This cmdlet returns an error if the user requests a verification code more than three times within a 24-hour period.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox ID for the user.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-WebServicesConnectivity

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-19*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-WebServicesConnectivity** cmdlet to perform basic operations to verify the functionality of Exchange Web Services on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-WebServicesConnectivity -MonitoringContext <SwitchParameter> <COMMON
PARAMETERS>


Test-WebServicesConnectivity [-ClientAccessServer
<ClientAccessServerIdParameter>] <COMMON PARAMETERS>


Test-WebServicesConnectivity -AutoDiscoverServer
<ClientAccessServerIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-Identity
<MailboxIdParameter>] [-LightMode <SwitchParameter>] [-MailboxCredential
<PSCredential>] [-TrustAnySSLCertificate <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the connection to Exchange Web Services on the Client Access server named CAS01. The test account that was created by running the **New-TestCasConnectivityUser.ps1** script is used.

```
Test-WebServicesConnectivity -ClientAccessServer CAS01
```

Detailed Description

To use most of the features of this cmdlet, you need to create a test user by running the following command.

```
& $env:ExchangeInstallPath\Scripts\New-
TestCasConnectivityUser.ps1
```

The **Test-WebServicesConnectivity** results are displayed on-screen. The cmdlet returns the following information.

- **Source** Source server.
- **ServiceEndpoint** Destination server.
- **Scenario** The operations that are tested. Values are `Autodiscover: SOAP Provider` and `EWS: GetFolder` or `EWS: ConvertID`.
- **Result** The values returned are typically `Success` or `*FAILURE*`.
- **Latency (MS)** The time required to complete the test in milliseconds

You can write the results to a file by piping the output to **ConvertTo-Html** or **ConvertTo-Csv** and adding "> <filename>" to the command. For example:

```
Test-WebServicesConnectivity -ClientAccessServer CAS01 |
ConvertTo-Html > "C:\My Documents\EWS Test.html"
```

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Test Exchange Web Services" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
AutoDiscoverServer	Required	Microsoft.Exchange.Co	The *AutoDiscoverServer*

		nfiguration.Tasks.ClientAccessServerIdParameter	parameter specifies the Client Access server to use to test Autodiscover connectivity. You can use any value that uniquely identifies the server. For example: • Name Example: Exchange01 • Distinguished name (DN) Example: `CN=Exchange01,CN=Servers,CN=Exchange Administrative Group (FYDIBOHF23SPDLT),CN=Administrative Groups,CN=First Organization,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=contoso,DC=com` • Exchange Legacy DN Example: `/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=Exchange01` • GUID Example: `bc014a0d-1509-4ecc-b569-f077eec54942`
MonitoringContext	Required	System.Management.Automation.SwitchParameter	The *MonitoringContext* switch includes the associated monitoring events and performance counters in the results. You don't need to specify

			a value with this switch. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
ClientAccessServer	Optional	Microsoft.Exchange.Configuration.Tasks.ClientAccessServerIdParameter	The *ClientAccessServer* parameter specifies the Client Access server to use when the command is run. You can use any value that uniquely identifies the server. For example: • Name Example: Exchange01 • Distinguished name (DN) Example: `CN=Exchange01,CN=Servers,CN=Exchange Administrative Group (FYDIBOHF23SPDLT),CN=Administrative Groups,CN=First Organization,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=contoso,DC=com` • Exchange Legacy DN Example: `/o=First Organization/ou=Exchange Administrative Group`

			`(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=Exchange01` • GUID Example: `bc014a0d-1509-4ecc-b569-f077eec54942`
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox to use for the test. When you use this parameter, you also need to use the *MailboxCredential* parameter. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example:

			CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
LightMode	Optional	System.Management.Automation.SwitchParameter	The *LightMode* switch instructs the command to perform only a subset of the connectivity tests. When you use this parameter, the `EWS: ConvertId` operation is tested instead of the `EWS: GetFolder` operation.
MailboxCredential	Optional	System.Management.Automation.PSCredential	The *MailboxCredential* parameter specifies the mailbox credentials to use

			when the command is run. This parameter is required when you use the *Identity* parameter. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	The *TrustAnySSLCertificate* switch allows the test to use any SSL certificate that's available.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-WebServicesVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-WebServicesVirtualDirectory** cmdlet to retrieve information in Active Directory for the Exchange Web Services virtual directory from a computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-WebServicesVirtualDirectory -Server <ServerIdParameter> <COMMON
PARAMETERS>

Get-WebServicesVirtualDirectory [-Identity <VirtualDirectoryIdParameter>]
<COMMON PARAMETERS>

COMMON PARAMETERS: [-ADPropertiesOnly <SwitchParameter>] [-
DomainController <Fqdn>] [-ShowBackEndVirtualDirectories
<SwitchParameter>] [-ShowMailboxVirtualDirectories <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns the settings for the Exchange Web Services virtual directory EWS under the default website in IIS.

```
Get-WebServicesVirtualDirectory -Identity "EWS (Default Web
Site)"
```

EXAMPLE 2

This example uses the *Identity* parameter to retrieve all settings for the Exchange Web Services virtual directories on the server CAS01.

```
Get-WebServicesVirtualDirectory -Identity CAS01
```

Detailed Description

The **Get-WebServicesVirtualDirectory** cmdlet can be run on a local server or run remotely if the server name is specified in the *Identity* or *Server* parameters. It can also be run without parameters to retrieve the configuration settings from all Microsoft Office Outlook Web App virtual directories on all Internet Information Services (IIS) websites located on the Client Access servers in the organization.

The **Get-WebServicesVirtualDirectory** cmdlet can be run on any server that has the Exchange administration tools installed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Web Services virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name or GUID of the server that hosts the virtual directories that you want to display.
ADPropertiesOnly	Optional	System.Management.Automation.SwitchParameter	The *ADPropertiesOnly* switch specifies whether to return only the properties about the virtual directory stored in Active Directory. The properties stored in the

			Internet Information Services (IIS) metabase aren't returned.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the GUID of the server, the name of the website, or the name of the virtual directory that you want to display.
ShowBackendVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ShowMailboxVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowMailboxVirtualDirectories* switch specifies whether the virtual directories on the Mailbox servers within the organization are shown.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-WebServicesVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-WebServicesVirtualDirectory** cmdlet to create an Exchange Web Services virtual directory on a server running Microsoft Exchange Server 2013.

You can use this cmdlet to create multiple virtual directories. However, you can create only one Exchange Web Services virtual directory organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-WebServicesVirtualDirectory [-ApplicationRoot <String>] [-AppPoolId
<String>] [-AppPoolIdForManagement <String>] [-BasicAuthentication <$true
| $false>] [-Confirm [<SwitchParameter>]] [-DigestAuthentication <$true |
$false>] [-DomainController <Fqdn>] [-ExtendedProtectionFlags
<MultiValuedProperty>] [-ExtendedProtectionSPNList <MultiValuedProperty>]
[-ExtendedProtectionTokenChecking <None | Allow | Require>] [-ExternalUrl
<Uri>] [-Force <SwitchParameter>] [-GzipLevel <Off | Low | High | Error>]
[-InternalNLBBypassUrl <Uri>] [-InternalUrl <Uri>] [-MRSProxyEnabled
<$true | $false>] [-OAuthAuthentication <$true | $false>] [-Path <String>]
[-Role <ClientAccess | Mailbox>] [-Server <ServerIdParameter>] [-
WebSiteName <String>] [-WhatIf [<SwitchParameter>]] [-
WindowsAuthentication <$true | $false>] [-WSSecurityAuthentication <$true
| $false>]
```

Examples

EXAMPLE 1

This example creates a virtual directory under the non-default website www.contoso.com. It also sets the external URL to https://www.contoso.com/webservices.aspx.

```
New-WebServicesVirtualDirectory -WebSiteName
"www.contoso.com" -ExternalUrl "https://www.contoso.com/
```

```
webservices.aspx"
```

Detailed Description

You can create multiple virtual directories by using this cmdlet. However, you can only create one Exchange Web Services virtual directory for each website.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Web Services virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
ApplicationRoot	Optional	System.String	The *ApplicationRoot* parameter sets the metabase path of the virtual directory. By default, this path is the same as the website in which the virtual directory is created.
AppPoolId	Optional	System.String	The *AppPoolId* parameter sets the pool of programs that can be used with the virtual directory.
AppPoolIdForManagement	Optional	System.String	The *AppPoolIdForManagement* parameter specifies the pool of programs that manages the virtual directory.
BasicAuthentication	Optional	System.Boolean	The *BasicAuthentication* parameter specifies

			whether Basic authentication is enabled on the Exchange Web Services virtual directory.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DigestAuthentication	Optional	System.Boolean	The *DigestAuthentication* parameter specifies whether Digest authentication is enabled on the virtual directory.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you

			use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain name (FQDN) format, for example ContosoMail. You specify valid SPNs with the

			ExtendedProtectionSPNList parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default

			value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN` value. You specify the domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionTokenChecking* parameter defines how you want to use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't

			enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the client and server will work, but may not be as secure as a connection using Extended Protection for Authentication.

			Note: If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one

			or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the host name used to connect to the Exchange server from outside the firewall. This setting is also important when Secure Sockets Layer (SSL) is used.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress the warning or confirmation messages that appear during specific configuration changes.
GzipLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.GzipLevel	The *GzipLevel* parameter sets Gzip configuration information for the Exchange Web Services virtual directory. This parameter can be set to the following values: • `Off` This value results in no compression. • `Low` This value results in static compression only. Don't use this setting for Exchange Web Services because Exchange Web Services content is dynamic. You'll get a warning if

			you set the *GzipLevel* parameter to this value. If you use this setting, it behaves the same as the `Off` setting. • `High` This value results in static and dynamic compression. Content from Exchange Web Services is compressed if clients have indicated support for Gzip compression in their requests. • `Error` This value identifies errors in the Gzip compression configuration.
InternalNLBBypassUrl	Optional	System.Uri	The *InternalNLBBypassUrl* parameter specifies the URL of the Client Access server, regardless of whether it's behind a Network Load Balancing (NLB) array. Although the *InternalUrl* parameter is set to the URL of the NLB array, the *InternalNLBBypassUrl* parameter should always be set to the URL of the Client Access server.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the host name of the Exchange server for a connection from inside the firewall. This setting is

			also important when SSL is used.
MRSProxyEnabled	Optional	System.Boolean	The *MRSProxyEnabled* parameter specifies whether to enable MRSProxy for the Client Access server. MRSProxy is a service that runs on Client Access servers in a remote forest and helps to proxy a mailbox move. For more information, see Mailbox moves in Exchange 2013.
OAuthAuthentication	Optional	System.Boolean	The *OAuthAuthentication* parameter specifies whether OAuth authentication is enabled.
Path	Optional	System.String	The *Path* parameter sets the path of the virtual directory in the metabase.
Role	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.VirtualDirectoryRole	The *Role* parameter specifies the configuration that should be used when the virtual directory is created. The following are the values that can be used with this parameter: • `FrontEnd` Configures the virtual directory for use on a Client Access server.

			• `BackEnd` Configures the virtual directory for use on a Mailbox server.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to create the virtual directory. You can use any value that uniquely identifies the server, for example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the virtual directory is created on the server where the Remote PowerShell session is established This will always be a Mailbox server. If you want to create the virtual directory on a Client Access server or another Mailbox server, you must use the *Server* parameter.
WebSiteName	Optional	System.String	The *WebSiteName* parameter specifies the name of the website under which to create the

			virtual directory. This parameter shouldn't be used when you're creating a virtual directory under the default website.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsAuthentication	Optional	System.Boolean	The *WindowsAuthentication* parameter specifies whether Integrated Windows authentication is permitted on the new Exchange Web Services virtual directory.
WSSecurityAuthentication	Optional	System.Boolean	The *WSSecurityAuthentication* parameter specifies whether Web Services Security authentication is enabled on the Exchange

			Web Services virtual directory. This parameter can be used with the *BasicAuthentication*, *DigestAuthentication*, and *WindowsAuthentication* parameters.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-WebServicesVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-WebServicesVirtualDirectory** cmdlet to remove an existing virtual directory from a computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-WebServicesVirtualDirectory -Identity <VirtualDirectoryIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Force
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the virtual directory Sales from the server CAS01.

```
Remove-WebServicesVirtualDirectory -Identity CAS01\Sales
```

Detailed Description

You can remove the default Exchange Web Services virtual directory or another Exchange Web Services virtual directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Web Services virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies a virtual directory.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this

			configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to force the command to execute without asking for user confirmation. If set to `$true`, this parameter forces the command to execute without asking for user confirmation.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-WebServicesVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Client Access cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-10*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-WebServicesVirtualDirectory** cmdlet to modify an existing Exchange Web Services virtual directory on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-WebServicesVirtualDirectory -Identity <VirtualDirectoryIdParameter> [-
BasicAuthentication <$true | $false>] [-CertificateAuthentication <$true |
$false>] [-Confirm [<SwitchParameter>]] [-DigestAuthentication <$true |
$false>] [-DomainController <Fqdn>] [-ExtendedProtectionFlags
<MultiValuedProperty>] [-ExtendedProtectionSPNList <MultiValuedProperty>]
[-ExtendedProtectionTokenChecking <None | Allow | Require>] [-ExternalUrl
<Uri>] [-Force <SwitchParameter>] [-GzipLevel <Off | Low | High | Error>]
[-InternalNLBBypassUrl <Uri>] [-InternalUrl <Uri>] [-
LiveIdBasicAuthentication <$true | $false>] [-
LiveIdNegotiateAuthentication <$true | $false>] [-MRSProxyEnabled <$true |
$false>] [-OAuthAuthentication <$true | $false>] [-WhatIf
[<SwitchParameter>]] [-WindowsAuthentication <$true | $false>] [-
WSSecurityAuthentication <$true | $false>]
```

Examples

EXAMPLE 1

This example sets the authentication method to Basic authentication for the virtual directory EWS on the server Contoso. This example also sets the external and internal URLs for this virtual directory.

```
Set-WebServicesVirtualDirectory -Identity Contoso
\EWS(Default Web Site)-ExternalUrl https://www.contoso.com/
EWS/exchange.asmx -BasicAuthentication $true -InternalUrl
https://contoso.internal.com/EWS/exchange.asmx
```

EXAMPLE 2

This example uses a wildcard character instead of "Default Web site" as was used in Example 1.

```
Set-WebServicesVirtualDirectory -Identity Contoso\EWS* -
ExternalUrl https://www.contoso.com/EWS/exchange.asmx
```

EXAMPLE 3

This example enables MRSProxy on the EWS default website. MRSProxy is the service responsible for assisting in remote mailbox moves.

```
Set-WebServicesVirtualDirectory -Identity "EWS (Default Web
Site)" -MRSProxyEnabled $true
```

Detailed Description

If you have a load balanced set of Client Access servers, you need to specify the name of each Client Access server when you run the command.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Web Services virtual directory settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the name of the virtual directory. You can also specify a wildcard character instead of the default website.
BasicAuthentication	Optional	System.Boolean	The *BasicAuthentication* parameter specifies whether Basic authentication is enabled on the Exchange Web Services virtual directory. This parameter can be used with the

			DigestAuthentication, *WindowsAuthentication*, and *WSSecurityAuthentication* parameters.
CertificateAuthentication	Optional	System.Boolean	The *CertificateAuthentication* parameter specifies whether certificate authentication is enabled. This parameter affects the <*Servername*>/ews/management/ virtual directory. It doesn't affect the <*Servername*>/ews/ virtual directory.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DigestAuthentication	Optional	System.Boolean	The *DigestAuthentication* parameter specifies whether Digest authentication is enabled on the virtual directory.

			This parameter can be used with the *BasicAuthentication*, *WindowsAuthentication*, and *WSSecurityAuthentication* parameters.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy

			mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN

			list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionToke*

			nChecking parameter with the `AllowDotlessSPN value`. You specify the domain in SPN format. The SPN format is <*protocol*>/<*FQDN*>. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionTokenChecking* parameter defines how you want to use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for

			Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the client and server will work, but may not be as secure as a connection using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and

			Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the host name used to connect to the Exchange server from outside the firewall. This setting is also important when Secure Sockets Layer (SSL) is used.
Force	Optional	System.Management.Automation.SwitchPar	The *Force* switch specifies whether to suppress the

		ameter	warning or confirmation messages that appear during specific configuration changes.
GzipLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.GzipLevel	The *GzipLevel* parameter sets Gzip configuration information for the Exchange Web Services virtual directory. This parameter can be set to the following values: • `Off` This value results in no compression. • `Low` This value results in static compression only. Don't use this setting for Exchange Web Services because Exchange Web Services content is dynamic. You get a warning if you set the *GzipLevel* parameter to this value. If you use this setting, it behaves the same as the `Off` setting. • `High` This value results in static and dynamic compression. Content from Exchange Web Services is compressed if clients have indicated support for Gzip compression in their requests. • `Error` This value identifies errors in the Gzip compression configuration.

InternalNLBBypassUrl	Optional	System.Uri	The *InternalNLBBypassUrl* parameter specifies the URL of the Client Access server, regardless of whether it's behind a Network Load Balancing (NLB) array. Although the *InternalUrl* parameter is set to the URL of the NLB array, the *InternalNLBBypassUrl* parameter should always be set to the URL of the Client Access server.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the host name of the Exchange server for a connection from inside the firewall. This setting is also important when SSL is used.
LiveIdBasicAuthentication	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LiveIdNegotiateAuthentication	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
MRSProxyEnabled	Optional	System.Boolean	The *MRSProxyEnabled* parameter specifies whether to enable MRSProxy for the Client Access server. MRSProxy is a service that runs on

			Client Access servers in a remote forest and helps to proxy a mailbox move.
OAuthAuthentication	Optional	System.Boolean	The *OAuthAuthentication* parameter specifies whether OAuth authentication is enabled.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsAuthentication	Optional	System.Boolean	The *WindowsAuthentication* parameter specifies whether Integrated Windows authentication is permitted on the Exchange Web Services virtual directory. This parameter can be used with the *BasicAuthentication*, *DigestAuthentication*, and

			WSSecurityAuthentication parameters.
WSSecurityAuthentication	Optional	System.Boolean	The *WSSecurityAuthentication* parameter specifies whether Web Services Security authentication is enabled on the Exchange Web Services virtual directory. This parameter can be used with *BasicAuthentication*, *DigestAuthentication*, and *WindowsAuthentication* parameters.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Cmdlet extension agent cmdlets

Exchange Server 2013 > Exchange Management Shell > Exchange 2013 cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-07*

Disable-CmdletExtensionAgent

Enable-CmdletExtensionAgent

Get-CmdletExtensionAgent

Set-CmdletExtensionAgent

Disable-CmdletExtensionAgent

Exchange Management Shell > Exchange 2013 cmdlets > Cmdlet extension agent cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-CmdletExtensionAgent** cmdlet on a server running Microsoft Exchange Server 2013 to disable an existing cmdlet extension agent.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-CmdletExtensionAgent -Identity <CmdletExtensionAgentIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables a specific cmdlet extension agent.

```
Disable-CmdletExtensionAgent "Scripting Agent"
```

Detailed Description

Run the **Disable-CmdletExtensionAgent** cmdlet on an Exchange 2013 server to disable an existing cmdlet extension agent. When you disable a cmdlet extension agent, the agent is disabled for the entire organization. When an agent is disabled, it's not made available to cmdlets. Cmdlets can no longer use the agent to perform additional operations.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Cmdlet extension agents" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.CmdletExtensionAgentIdParameter	The *Identity* parameter specifies the name of the cmdlet extension agent to disable. If the name contains spaces, enclose the name in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the

			WhatIf switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-CmdletExtensionAgent

Exchange Management Shell > Exchange 2013 cmdlets > Cmdlet extension agent cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-CmdletExtensionAgent** cmdlet on a server running Microsoft Exchange Server 2013 to enable a cmdlet extension agent.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-CmdletExtensionAgent -Identity <CmdletExtensionAgentIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the Scripting Agent cmdlet extension agent. Before you enable the Scripting Agent, you need to make sure that you've first deployed the ScriptingAgentConfig.xml configuration file to all the servers in your organization. If you don't deploy the configuration file first and you enable the Scripting Agent, all non-**Get** cmdlets fail when they're run.

```
Enable-CmdletExtensionAgent "Scripting Agent"
```

Detailed Description

When you enable a cmdlet extension agent, the agent is run on every Exchange 2013 server in the organization. When an agent is enabled, it's made available to cmdlets that can then use the agent to perform additional operations.

Caution:
Before you enable agents, be sure that you're aware of how the agent works and what impact the agent will have on your organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Cmdlet extension agents" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.CmdletExtensionAgentIdParameter	The *Identity* parameter specifies the name of the cmdlet extension agent to enable. If the name contains spaces, enclose the name in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do

			before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-CmdletExtensionAgent

Exchange Management Shell > Exchange 2013 cmdlets > Cmdlet extension agent cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-CmdletExtensionAgent** cmdlet on a server running Microsoft Exchange Server 2013 to display a list of cmdlet extension agents in the organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-CmdletExtensionAgent [-Assembly <String>] [-Enabled <$true | $false>]
<COMMON PARAMETERS>

Get-CmdletExtensionAgent [-Identity <CmdletExtensionAgentIdParameter>]
<COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example displays the details about a specific cmdlet extension agent.

```
Get-CmdletExtensionAgent "Mailbox Permissions Agent"
```

EXAMPLE 2

This example displays a list of all the cmdlet extension agents in the organization. The **Name**, **Enabled**, and **Priority** properties of each agent are displayed in a table. This is done by piping the results of the **Get-CmdletExtensionAgent** cmdlet to the **Format-Table** cmdlet.

For more information about pipelining and the **Format-Table** cmdlet, see the following topics:

- Pipelining
- Working with command output

```
Get-CmdletExtensionAgent | Format-Table Name, Enabled,
Priority
```

Detailed Description

Run the **Get-CmdletExtensionAgent** cmdlet on an Exchange 2013 server to display a list of cmdlet extension agents in the organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Cmdlet extension agents" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Assembly	Optional	System.String	The *Assembly* parameter specifies that only the agents that match the assembly name provided should be listed.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the cmdlet should return a list of enabled agents only, or a list of disabled agents only. Valid values are `$true`

			and `$false`.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.CmdletExtensionAgentIdParameter	The *Identity* parameter specifies the name of the cmdlet extension agent to view. If the name contains spaces, enclose the name in quotation marks (").

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-CmdletExtensionAgent

Exchange Management Shell > Exchange 2013 cmdlets > Cmdlet extension agent cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-CmdletExtensionAgent** cmdlet on a server running Microsoft Exchange Server 2013 to modify a cmdlet extension agent.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-CmdletExtensionAgent -Identity <CmdletExtensionAgentIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-IsSystem <$true
| $false>] [-Name <String>] [-Priority <Byte>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the priority of the fictitious "Validation Agent" cmdlet extension agent to 9.

```
Set-CmdletExtensionAgent "Validation Agent" -Priority 9
```

Detailed Description

The changes applied to an agent are applied to every Exchange 2013 server in the organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Cmdlet extension agents" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.CmdletExtensionAgentIdParameter	The *Identity* parameter specifies the name of the cmdlet extension agent to modify. If the name contains spaces, enclose the name in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IsSystem	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
Name	Optional	System.String	The *Name* parameter specifies the name of the cmdlet extension agent. The maximum length of the agent name is 64 characters. If the name contains spaces, enclose the name in quotation marks (").
Priority	Optional	System.Byte	The *Priority* parameter specifies where in the priority order of the cmdlet extension agent list the agent should be placed. The priority must be between 0 and the maximum number of agents. Agents with a priority closest to 0

			have a higher priority and are run first.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Email address and address book cmdlets

Exchange Server 2013 > Exchange Management Shell > Exchange 2013 cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-07*

Address book policy cmdlets

Get-AddressBookPolicy

New-AddressBookPolicy

Remove-AddressBookPolicy

Set-AddressBookPolicy

Address list cmdlets

Get-AddressList

Move-AddressList

New-AddressList

Remove-AddressList

Set-AddressList

Update-AddressList

Disable-AddressListPaging

Enable-AddressListPaging

Get-DetailsTemplate

Restore-DetailsTemplate

Set-DetailsTemplate

New-GlobalAddressList

Get-GlobalAddressList

Remove-GlobalAddressList

Set-GlobalAddressList

Update-GlobalAddressList

Email address policy cmdlets

Get-EmailAddressPolicy

New-EmailAddressPolicy

Remove-EmailAddressPolicy

Set-EmailAddressPolicy

Update-EmailAddressPolicy

Offline address book cmdlets

Get-OabVirtualDirectory

New-OabVirtualDirectory

Remove-OabVirtualDirectory

Set-OabVirtualDirectory

Get-OfflineAddressBook

New-OfflineAddressBook

Move-OfflineAddressBook

Remove-OfflineAddressBook

Set-OfflineAddressBook

Update-OfflineAddressBook

Get-AddressBookPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-AddressBookPolicy** cmdlet to return address book policies that match the specified conditions.

```
Get-AddressBookPolicy [-Identity <MailboxPolicyIdParameter>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example returns detailed information about all address book policies in your organization by pipelining the **Format-List** cmdlet.

```
Get-AddressBookPolicy | Format-List
```

EXAMPLE 2

This example returns default information about the address book policy All Fabrikam

```
Get-AddressBookPolicy -Identity "All Fabrikam"
```

EXAMPLE 3

This example returns information about all address book policies for which the offline address book (OAB) that the address book policy uses is named Fabrikam All OAB.

```
Get-AddressBookPolicy | where {$_.OfflineAddressBook eq
"\Fabrikam All OAB"}
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address book policies" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the identity of the address book policy.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-AddressBookPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-AddressBookPolicy** cmdlet to create an address book policy. Address book policies define the global address list (GAL), offline address book (OAB), room list, and address lists that will be displayed to mailbox users who are assigned the policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-AddressBookPolicy -AddressLists <AddressListIdParameter[]> -
GlobalAddressList <GlobalAddressListIdParameter> -Name <String> -
OfflineAddressBook <OfflineAddressBookIdParameter> -RoomList
<AddressListIdParameter> [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates an address book policy with the following settings:

- Name: All Fabrikam ABP
- Included address lists: All Fabrikam, All Fabrikam Mailboxes, All Fabrikam DLs, All Fabrikam Contacts
- Included room list: All Fabrikam-Rooms
- Included OAB: Fabrikam-All-OAB
- Included GAL: All Fabrikam

```
New-AddressBookPolicy -Name "All Fabrikam ABP" -
AddressLists "\All Fabrikam","\All Fabrikam
Mailboxes","\All Fabrikam DLs","\All Fabrikam Contacts" -
RoomList "\All Fabrikam-Rooms" -OfflineAddressBook
"\Fabrikam-All-OAB" -GlobalAddressList "\All Fabrikam"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address book policies" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
AddressLists	Required	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter[]	The *AddressLists* parameter specifies the address lists that will be used by mailbox users who are assigned this address book policy. This parameter

			accepts multiple values, which should be separated by a comma. For example, `"\Mr. Munson's Class","Mrs. McKay's Class","Mrs. Count's Class"`.
GlobalAddressList	Required	Microsoft.Exchange.Configuration.Tasks.GlobalAddressListIdParameter	The *GlobalAddressList* parameter specifies the identity of the GAL that will be used by mailbox users who are assigned this address book policy. You can specify only one GAL for each address book policy.
Name	Required	System.String	The *Name* parameter specifies the name of the address book policy.
OfflineAddressBook	Required	Microsoft.Exchange.Configuration.Tasks.OfflineAddressBookIdParameter	The *OfflineAddressBook* parameter specifies the identity of the OAB that will be used by mailbox users who are assigned this address book policy. You can specify only one OAB for each address book policy.
RoomList	Required	Microsoft.Exchange.Co	The *RoomList*

		nfiguration.Tasks.AddressListIdParameter	parameter specifies the room address list that will be used by mailbox users who are assigned this address book policy. You can specify only one room list for each address book policy.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft

			use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-AddressBookPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-AddressBookPolicy** cmdlet to delete an address book policy. You can't remove

the address book policy if it's still assigned to a user's mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-AddressBookPolicy -Identity <MailboxPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the address book policy Murchison's Class.

```
Remove-AddressBookPolicy -Identity "Murchison's Class"
```

Detailed Description

You can't delete an address book policy if it's assigned to a user. To determine if an address book policy is assigned to a user, run the following command:

```
Get-Mailbox | where $._AddressBookPolicy -eq "Murchison's
Class"}
```

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address book policies" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the identity of the address book policy that you want to remove.
Confirm	Optional	System.Management.A	The *Confirm* switch can

		utomation.SwitchParameter	be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AddressBookPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-09*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-AddressBookPolicy** cmdlet to change the settings of an address book policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-AddressBookPolicy -Identity <MailboxPolicyIdParameter> [-AddressLists
<AddressListIdParameter[]>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-GlobalAddressList
<GlobalAddressListIdParameter>] [-Name <String>] [-OfflineAddressBook
<OfflineAddressBookIdParameter>] [-RoomList <AddressListIdParameter>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the OAB that the address book policy All Fabrikam ABP uses to Fabrikam-OAB-2.

```
Set-AddressBookPolicy -Identity "All Fabrikam ABP" -
OfflineAddressBook \Fabrikam-OAB-2 -GlobalAddressList "\All
Fabrikam GAL"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address book policies" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the identity of the address book policy that you want to modify.
AddressLists	Optional	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter[]	The *AddressLists* parameter specifies the address lists that will be used by mailbox users who are assigned this address book policy. This parameter accepts multiple values, which should be separated by a comma.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -

			`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
GlobalAddressList	Optional	Microsoft.Exchange.Configuration.Tasks.GlobalAddressListIdParameter	The *GlobalAddressList* parameter specifies the identity of the global address list (GAL) that will be used by mailbox users who are assigned this address book policy. You can specify only one GAL for each address book policy.
Name	Optional	System.String	The *Name* parameter specifies the name that you want this address book policy to be called. Use this parameter to change the name of the address book policy.
OfflineAddressBook	Optional	Microsoft.Exchange.Configuration.Tasks.Offlin	The *OfflineAddressBook*

		eAddressBookIdParameter	parameter specifies the identity of the offline address book (OAB) that will be used by mailbox users who are assigned this address book policy. You can specify only one OAB for each address book policy.
RoomList	Optional	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *RoomList* parameter specifies the name of the room address list.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AddressList

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-AddressList** cmdlet to retrieve all attributes of an address list.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-AddressList -Container <AddressListIdParameter> <COMMON PARAMETERS>

Get-AddressList [-SearchText <String>] <COMMON PARAMETERS>

Get-AddressList [-Identity <AddressListIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves the attributes of all the address lists under the All Address Lists container.

```
Get-AddressList
```

EXAMPLE 2

This example retrieves the attributes of the address list building4, located under the All Users\Sales\ address list, using the domain controller DomainController01.

```
Get-AddressList -Identity "All Users\Sales\building4" -
DomainController DomainController01
```

Detailed Description

You can pipe the output from the **Get-AddressList** cmdlet to the **Remove-AddressList**, **Set-AddressList**, **Update-Addresslist**, and **Move-AddressList** cmdlets instead of using the *Identity* parameter with each of those cmdlets.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Container	Required	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *Container* parameter specifies the identity of the parent address list of the address list or lists that you want to view. If no parent address list is specified, the command gets all address lists under the root All Address Lists.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Identity	Optional	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or address list name that represents a specific address list. You can also include the path using the format *Path\AddressListName*. You can omit the parameter label so that only the address list name or GUID is supplied.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter specifies the organization in which you'll perform this action. This parameter doesn't accept wildcard characters, and you must use the exact name of the organization.
SearchText	Optional	System.String	The *SearchText* parameter specifies that you can search the names of all organizational units (OUs) in your organization for the

			specified string. Only the OUs that match the string you specify are returned. If the string you specify contains spaces, enclose it in quotation marks ("). This parameter can't be used with the Identity parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Move-AddressList

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Move-AddressList** cmdlet to move an existing address list to a new container under the root address list.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Move-AddressList -Identity <AddressListIdParameter> -Target
<AddressListIdParameter> [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example moves the address list with GUID c3fffd8e-026b-41b9-88c4-8c21697ac8ac to a new location under the parent address list \All Users\Sales\building4.

```
Move-AddressList -Identity c3fffd8e-026b-41b9-88c4-
8c21697ac8ac -Target "\All Users\Sales\building4"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or address list name that represents a specific address list. You can also include the path by using the format *Path**AddressListName*. You can omit the parameter label so that only the address list

			name or GUID is supplied.
Target	Required	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *Target* parameter specifies the path to the parent address list where you want to move this address list.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the

			WhatIf switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-AddressList

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-AddressList** cmdlet to create an address list and apply it to recipients.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-AddressList [-RecipientFilter <String>] <COMMON PARAMETERS>


New-AddressList [-ConditionalCompany <MultiValuedProperty>] [-
ConditionalCustomAttribute1 <MultiValuedProperty>] [-
ConditionalCustomAttribute10 <MultiValuedProperty>] [-
ConditionalCustomAttribute11 <MultiValuedProperty>] [-
ConditionalCustomAttribute12 <MultiValuedProperty>] [-
ConditionalCustomAttribute13 <MultiValuedProperty>] [-
ConditionalCustomAttribute14 <MultiValuedProperty>] [-
ConditionalCustomAttribute15 <MultiValuedProperty>] [-
ConditionalCustomAttribute2 <MultiValuedProperty>] [-
```

```
ConditionalCustomAttribute3 <MultiValuedProperty>] [-
ConditionalCustomAttribute4 <MultiValuedProperty>] [-
ConditionalCustomAttribute5 <MultiValuedProperty>] [-
ConditionalCustomAttribute6 <MultiValuedProperty>] [-
ConditionalCustomAttribute7 <MultiValuedProperty>] [-
ConditionalCustomAttribute8 <MultiValuedProperty>] [-
ConditionalCustomAttribute9 <MultiValuedProperty>] [-ConditionalDepartment
<MultiValuedProperty>] [-ConditionalStateOrProvince <MultiValuedProperty>]
[-IncludedRecipients <None | MailboxUsers | Resources | MailContacts |
MailGroups | MailUsers | AllRecipients>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-Confirm [<SwitchParameter>]] [-
Container <AddressListIdParameter>] [-DisplayName <String>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>] [-
RecipientContainer <OrganizationalUnitIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the address list MyAddressList. The address list includes recipients that are mailbox users and have the **StateOrProvince** property set to Washington or Oregon.

```
New-AddressList -Name MyAddressList -RecipientFilter
{((RecipientType -eq 'MailboxUser') -and ((StateOrProvince
-eq 'Washington') -or (StateOrProvince -eq 'Oregon')))}
```

EXAMPLE 2

This example creates the address list MyAddressList2 that includes mailboxes that have the *ConditionalStateOrProvince* parameter set to Washington.

```
New-AddressList -Name MyAddressList2 -
ConditionalStateOrProvince Washington -IncludedRecipients
MailboxUsers
```

EXAMPLE 3

This example creates the address list AL_AgencyB that includes mailboxes that have the value of the *CustomAttribute15* parameter contains AgencyB.

```
New-AddressList -Name "AL_AgencyB" -RecipientFilter
{((RecipientType -eq 'MailboxUser') -and (CustomAttribute15
-like *AgencyB*))}
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name for the new address list. The name can't exceed 64 characters, and it can't include a carriage return or a backslash (\).
ConditionalCompany	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCompany* parameter is a filter to specify a company. All recipients with a company attribute that matches the value that you input for this parameter are included in the address list. You can use multiple values as a comma-delimited list. You can't use this parameter if you use the *RecipientFilter* parameter. You must use either the *RecipientFilter* parameter or one of these filter parameters: • *ConditionalCompany*

			• *ConditionalDepartment* • *ConditionalStateOrProvince* • *IncludedRecipients*
ConditionalCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute10	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the

			ConditionalCustomAttribute1 value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute11	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.

ConditionalCustomAttribute12	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute13	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is

			`Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute14	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute15	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribu*

			te15 parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a

			Conditional parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the

			ConditionalCustomAttribute1 value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.

ConditionalCustomAttribute6	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute7	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is

			`Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute8	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute9	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribu*

			te15 parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalDepartment	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalDepartment* parameter is a filter to specify a department. All recipients with a department attribute that matches the value that you input for this parameter are included in the address list. You can use multiple values as a comma-delimited list. You can't use this parameter if you use the *RecipientFilter* parameter. You must use either the *RecipientFilter* parameter or one of these

			filter parameters: • *ConditionalCompany* • *ConditionalDepartment* • *ConditionalStateOrProvince* • *IncludedRecipients*
ConditionalStateOrProvince	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalStateOrProvince* parameter is a filter to specify a state or province. All recipients with a *ConditionalStateOrProvince* attribute that matches the value that you input for this parameter are included in the address list. You can use multiple values as a comma-delimited list. You can't use this parameter if you use the *RecipientFilter* parameter. You must use either the *RecipientFilter* parameter or one of these filter parameters: • *ConditionalCompany* • *ConditionalDepartment* • *ConditionalStateOrProvince* • *IncludedRecipients*
Confirm	Optional	System.Management.Automation.SwitchPar	The *Confirm* switch causes the command to

		ameter	pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Container	Optional	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *Container* parameter specifies the identity of the parent address list where this new address list is created. If no parent address list is specified, the address list is created under the root All Address Lists.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the display name of the address list. If no display name is provided, the name of the address list is also the display name.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory.
IncludedRecipients	Optional	Microsoft.Exchange.Data.Directory.Recipient.WellKnownRecipientType	The *IncludedRecipients* parameter is a filter to specify the type of recipient to include in the address list. You can use one or more of the following types: • `None` • `MailboxUsers` • `MailUsers` • `Resources` • `MailGroups` • `MailContacts` • `AllRecipients` You can't use this parameter if you use the *RecipientFilter* parameter. You must use either the *RecipientFilter* parameter or one of these filter parameters: • *ConditionalCompany* • *ConditionalDepartment* • *ConditionalStateOrProvince* • *IncludedRecipients*
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
RecipientContainer	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientContainer* parameter filters the recipients used to build the dynamic distribution

			group based on their location in Active Directory. The value of the *RecipientContainer* parameter can be the canonical name of an organizational unit (OU) or a domain. If a value for *RecipientContainer* isn't specified, the default search filter is the location of the dynamic distribution group in Active Directory.
RecipientFilter	Optional	System.String	The *RecipientFilter* parameter specifies a filter for recipients to include in the address list. You can't use this parameter if you use any of the following filter parameters: • *ConditionalCompany* • *ConditionalDepartment* • *ConditionalStateOrProvince* • *IncludedRecipients* You must use either this parameter or one of the preceding filter parameters.
WhatIf	Optional	System.Management.Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-AddressList

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-AddressList** cmdlet to remove an existing address list.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-AddressList -Identity <AddressListIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Recursive
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the address list Address_List_1 and all of its child address lists.

```
Remove-AddressList -Identity "Address_List_1" -Recursive
```

Detailed Description

The **Remove-AddressList** cmdlet can be used in conjunction with the **Get-AddressList** cmdlet as follows:

- Use the **Get-AddressList** cmdlet to get address list information, and then pipe the output to the **Format-List** cmdlet to get the GUID, distinguished name (DN), or path and name of an existing address list.
- Use the **Get-AddressList** cmdlet to get a specific existing address list, and then pipe the output directly to the **Remove-AddressList** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or address list name that represents a specific address list. You can also include the path by using the format *Path\AddressListName*.

			You can omit the parameter label so that only the address list name or GUID is supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Recursive	Optional	System.Management.Automation.SwitchParameter	The *Recursive* switch specifies whether the command removes all child address lists of the address list specified by the *Identity* parameter. If you don't

			specify this parameter and the address list to remove has child address lists, the command fails.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AddressList

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-AddressList** cmdlet to modify an existing address list.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-AddressList -Identity <AddressListIdParameter> [-ConditionalCompany
<MultiValuedProperty>] [-ConditionalCustomAttribute1
<MultiValuedProperty>] [-ConditionalCustomAttribute10
<MultiValuedProperty>] [-ConditionalCustomAttribute11
<MultiValuedProperty>] [-ConditionalCustomAttribute12
<MultiValuedProperty>] [-ConditionalCustomAttribute13
<MultiValuedProperty>] [-ConditionalCustomAttribute14
<MultiValuedProperty>] [-ConditionalCustomAttribute15
<MultiValuedProperty>] [-ConditionalCustomAttribute2
<MultiValuedProperty>] [-ConditionalCustomAttribute3
<MultiValuedProperty>] [-ConditionalCustomAttribute4
<MultiValuedProperty>] [-ConditionalCustomAttribute5
<MultiValuedProperty>] [-ConditionalCustomAttribute6
<MultiValuedProperty>] [-ConditionalCustomAttribute7
<MultiValuedProperty>] [-ConditionalCustomAttribute8
<MultiValuedProperty>] [-ConditionalCustomAttribute9
<MultiValuedProperty>] [-ConditionalDepartment <MultiValuedProperty>] [-
ConditionalStateOrProvince <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-DisplayName <String>] [-DomainController <Fqdn>] [-
ForceUpgrade <SwitchParameter>] [-IncludedRecipients <None | MailboxUsers
| Resources | MailContacts | MailGroups | MailUsers | AllRecipients>] [-
Name <String>] [-RecipientContainer <OrganizationalUnitIdParameter>] [-
RecipientFilter <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the name of an existing address list.

```
Set-AddressList -Identity "All Users\Sales\building4" -Name
building9
```

EXAMPLE 2

This example modifies the type of recipients and the department of the recipients included in the existing address list identified by its GUID.

```
Set-Addresslist -Identity c3fffd8e-026b-41b9-88c4-
8c21697ac8ac -IncludedRecipients MailboxUsers -
ConditionalDepartment Sales
```

Detailed Description

Use the **Get-AddressList** cmdlet, piped to **Format-List**, to get the GUID, distinguished name (DN), or path and name of an existing address list. Or, use **Get-AddressList** to get a specific existing

address list, and then pipe the output directly to the **Set-AddressList** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *Identity* parameter specifies the GUID, DN, or address list name that represents a specific address list. You can also include the path using the format *Path**AddressListName*. You can omit the parameter label *Identity* so that only the address list name or GUID is supplied.
ConditionalCompany	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCompany* parameter is a filter to specify a company. All recipients with a company attribute that matches the value that you input for this parameter are included in the address list that you're modifying.

			You can use multiple values as a comma-delimited list. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute10	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to

			ConditionalCustomAttribute15 parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute11	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients*

			parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute12	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute13	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For

			example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute14	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the

			RecipientFilter parameter.
ConditionalCustomAttribute15	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose

			CustomAttribute1 value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to

			ConditionalCustomAttribute15 parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients*

			parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute6	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute7	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For

			example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute8	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the

			RecipientFilter parameter.
ConditionalCustomAttribute9	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalDepartment	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalDepartment* parameter is a filter to specify a department. All recipients with a department attribute that matches the value that you input for this parameter are included in the address list. You can use multiple values as a

			comma-delimited list. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalStateOrProvince	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalStateOrProvince* parameter is a filter to specify a state or province. All recipients with a *ConditionalStateOrProvince* attribute that matches the value that you input for this parameter are included in the address list. You can use multiple values as a comma-delimited list. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the desired display name of the address list.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* switch suppresses the following confirmation: "To save changes on object, the object must be upgraded to the current Exchange version. After upgrade, this object can't be managed by a previous version of Exchange System Manager. Do you want to continue to upgrade and save the object?"

IncludedRecipients	Optional	Microsoft.Exchange.Data.Directory.Recipient.WellKnownRecipientType	The *IncludedRecipients* parameter filters the recipient types used to build the global address list (GAL). The following are the available values for the *IncludedRecipients* parameter: • `None` • `MailboxUsers` • `Resources` • `MailContacts` • `MailGroups` • `MailUsers` • `AllRecipients` The `AllRecipients` value can be used only by itself. When multiple values of the *IncludedRecipients* parameter are separated by commas, the **OR** Boolean operator is applied. You must use this parameter if you're using any of the *Conditional* parameters. You can't use this parameter if you use the *RecipientFilter* parameter.
Name	Optional	System.String	The *Name* parameter specifies a new name for the address list. The name can't exceed 64 characters, and it can't

			include a carriage return or a backslash (\).
RecipientContainer	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientContainer* parameter filters the recipients used to build the dynamic distribution group based on their location in Active Directory. The value of the *RecipientContainer* parameter can be the canonical name of an organizational unit (OU) or a domain. If a value for the *RecipientContainer* parameter isn't specified, the default search filter is the location of the dynamic distribution group in Active Directory.
RecipientFilter	Optional	System.String	The *RecipientFilter* parameter specifies a filter for recipients to include in the address list. You can't use this parameter if you use any of the following filter parameters: • *ConditionalCompany* • *ConditionalDepartment* • *ConditionalStateOrProvince* • *IncludedRecipients*

			• *RecipientContainer*
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-AddressList

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Update-AddressList** cmdlet to update the recipients included in the address list that you specify.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-AddressList -Identity <AddressListIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example updates the recipients of the address list building4 and under the container All Users \Sales.

```
Update-AddressList -Identity "All Users\Sales\building4"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or address list name that represents a specific address list. You can also include the path by using the format *Path\AddressListName*. You can omit the parameter label so that

			only the address list name or GUID is supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of

			those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-AddressListPaging

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-AddressListPaging** cmdlet to disable Active Directory virtual list view for address lists. Virtual list view displays address lists in your organization as pages instead of loading and viewing the entire directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-AddressListPaging [-Identity <OrganizationIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables virtual list view for address lists in your organization.

`Disable-AddressListPaging`

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter specifies the identity of the tenant organization.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-AddressListPaging

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-AddressListPaging** cmdlet to enable Active Directory virtual list view for address lists. Virtual list view allows you to display the address lists in your organization as pages instead of loading and viewing the entire directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-AddressListPaging [-Identity <OrganizationIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-DoNotUpdateRecipients
<SwitchParameter>] [-ForceUpdateOfRecipients <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables virtual list view for your organization.

```
Enable-AddressListPaging
```

Detailed Description

The **Enable-AddressListPaging** cmdlet creates the **Address List** container in Active Directory. Recipient cmdlets, such as **Get-Recipient**, use the information written to the container to quickly retrieve recipient data.

Note:
Microsoft Exchange Server 2007 can't read the **Address List** container and will clear its contents while managing address lists. As a result, the Exchange Server 2010 or later recipient cmdlets won't return any recipient data. If you're in a coexistence deployment with Exchange 2007, and you use Exchange 2007 servers to manage address lists, don't enable address list paging. If you've already enabled address list paging, you can disable it by running the Disable-AddressListPaging cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn’t assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the “Add a role to a role assignment policy” section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
DoNotUpdateRecipients	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ForceUpdateOfRecipients	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchPara	The *WhatIf* switch instructs the command

		meter	to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DetailsTemplate

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-DetailsTemplate** cmdlet to retrieve the attributes for details templates.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-DetailsTemplate [-Identity <DetailsTemplateIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example retrieves all attributes for the User details template for the English language.

```
Get-DetailsTemplate -Identity en-us\User
```

EXAMPLE 2

This example retrieves all attributes for all details template types in all languages.

```
Get-DetailsTemplate -Identity *\*
```

EXAMPLE 3

This example retrieves all attributes for all User details template types in all languages.

```
Get-DetailsTemplate -Identity *\User
```

Detailed Description

The **Get-DetailsTemplate** cmdlet retrieves the attributes for one or more details templates. Wildcard characters can be used when specifying the type and language of the details templates.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Details templates" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a

			value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.DetailsTemplateIdParameter	The *Identity* parameter specifies the GUID of the details template or specifies the details template type and language separated by a backslash, for example, en-us\User. Details template types are: • User • Group • PublicFolder • SearchDialog • MailboxAgent • Contact
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply

			any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Restore-DetailsTemplate

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Restore-DetailsTemplate** cmdlet to restore the specified template to its default state.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Restore-DetailsTemplate -Identity <DetailsTemplateIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example restores the default attributes to the User details template for the English language.

```
Restore-DetailsTemplate -Identity en-us\User
```

Detailed Description

Details templates can't be created or deleted, but this task restores the specified template to its default state. All user changes are lost.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Details templates" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DetailsTemplateIdParameter	The *Identity* parameter specifies the GUID or specifies the details template type and language separated by a backslash, for example, en-us\User. Details template types include: • User • Group • PublicFolder • SearchDialog • MailboxAgent • Contact
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You

			must include a colon (:) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-DetailsTemplate

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-DetailsTemplate** cmdlet to modify the attributes of a details template. To make changes to the details template format and layout, you need to use the Details Templates Editor in the Exchange Toolbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-DetailsTemplate -Identity <DetailsTemplateIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Pages
<MultiValuedProperty>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets attributes for the User details template for the U.S. English language.

```
Set-DetailsTemplate -Identity en-us\User
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Details templates" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DetailsTemplateIdParameter	The *Identity* parameter specifies the details template using a GUID or specifies a template type

			and language separated by a slash. The following is an example of the user template type and U.S. English language: en-us \User. Details template types include: • User • Group • PublicFolder • SearchDialog • MailboxAgent • Contact
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Pages	Optional	Microsoft.Exchange.Da	This parameter is reserved

		ta.MultiValuedProperty	for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-EmailAddressPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-EmailAddressPolicy** cmdlet to return all of the attributes on a policy or set of policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-EmailAddressPolicy [-Identity <EmailAddressPolicyIdParameter>] [-
DomainController <Fqdn>] [-IncludeMailboxSettingOnlyPolicy
<SwitchParameter>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves the attributes for Email Address Policy 1.

```
Get-EmailAddressPolicy -Identity "Email Address Policy 1"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Email address policies" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.EmailAddressPolicyIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or email address policy name that represents a specific email address policy. You can also include

			the path using the format *Path**EmailAddressPolicy*. You can omit the parameter label *Identity* so that only the GUID, DN, or email address policy name is supplied.
IncludeMailboxSettingOnlyPolicy	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-EmailAddressPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-EmailAddressPolicy** cmdlet to create an email address policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-EmailAddressPolicy -EnabledEmailAddressTemplates
<ProxyAddressTemplateCollection> -RecipientFilter <String> [-
DisabledEmailAddressTemplates <ProxyAddressTemplateCollection>] <COMMON
PARAMETERS>


New-EmailAddressPolicy -EnabledPrimarySMTPAddressTemplate <String> -
RecipientFilter <String> <COMMON PARAMETERS>


New-EmailAddressPolicy -EnabledEmailAddressTemplates
<ProxyAddressTemplateCollection> -IncludedRecipients <None | MailboxUsers
| Resources | MailContacts | MailGroups | MailUsers | AllRecipients> [-
ConditionalCompany <MultiValuedProperty>] [-ConditionalCustomAttribute1
<MultiValuedProperty>] [-ConditionalCustomAttribute10
<MultiValuedProperty>] [-ConditionalCustomAttribute11
<MultiValuedProperty>] [-ConditionalCustomAttribute12
<MultiValuedProperty>] [-ConditionalCustomAttribute13
<MultiValuedProperty>] [-ConditionalCustomAttribute14
<MultiValuedProperty>] [-ConditionalCustomAttribute15
<MultiValuedProperty>] [-ConditionalCustomAttribute2
<MultiValuedProperty>] [-ConditionalCustomAttribute3
<MultiValuedProperty>] [-ConditionalCustomAttribute4
<MultiValuedProperty>] [-ConditionalCustomAttribute5
<MultiValuedProperty>] [-ConditionalCustomAttribute6
<MultiValuedProperty>] [-ConditionalCustomAttribute7
<MultiValuedProperty>] [-ConditionalCustomAttribute8
<MultiValuedProperty>] [-ConditionalCustomAttribute9
<MultiValuedProperty>] [-ConditionalDepartment <MultiValuedProperty>] [-
ConditionalStateOrProvince <MultiValuedProperty>] [-
DisabledEmailAddressTemplates <ProxyAddressTemplateCollection>] <COMMON
PARAMETERS>


New-EmailAddressPolicy -EnabledPrimarySMTPAddressTemplate <String> -
IncludedRecipients <None | MailboxUsers | Resources | MailContacts |
MailGroups | MailUsers | AllRecipients> [-ConditionalCompany
<MultiValuedProperty>] [-ConditionalCustomAttribute1
<MultiValuedProperty>] [-ConditionalCustomAttribute10
<MultiValuedProperty>] [-ConditionalCustomAttribute11
<MultiValuedProperty>] [-ConditionalCustomAttribute12
<MultiValuedProperty>] [-ConditionalCustomAttribute13
<MultiValuedProperty>] [-ConditionalCustomAttribute14
<MultiValuedProperty>] [-ConditionalCustomAttribute15
<MultiValuedProperty>] [-ConditionalCustomAttribute2
<MultiValuedProperty>] [-ConditionalCustomAttribute3
<MultiValuedProperty>] [-ConditionalCustomAttribute4
<MultiValuedProperty>] [-ConditionalCustomAttribute5
<MultiValuedProperty>] [-ConditionalCustomAttribute6
<MultiValuedProperty>] [-ConditionalCustomAttribute7
<MultiValuedProperty>] [-ConditionalCustomAttribute8
<MultiValuedProperty>] [-ConditionalCustomAttribute9
<MultiValuedProperty>] [-ConditionalDepartment <MultiValuedProperty>] [-
ConditionalStateOrProvince <MultiValuedProperty>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>] [-
Priority <EmailAddressPolicyPriority>] [-RecipientContainer
<OrganizationalUnitIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates an email address policy that includes mailbox users in the southeast offices who have email addresses that include their last name combined with the first two letters of their first name.

```
New-EmailAddressPolicy -Name "southeast offices" -
IncludedRecipients UserMailbox -ConditionalStateOrProvince
"Georgia","Alabama","Louisiana" -
EnabledEmailAddressTemplates "SMTP:%s%
2g@southeast.contoso.com"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Email address policies" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
EnabledEmailAddressTemplates	Required	Microsoft.Exchange.Data.ProxyAddressTemplateCollection	The *EnabledEmailAddressTemplates* parameter specifies the proxy addresses included in an email address policy and are enabled. Separate multiple values with commas. The domain part of each proxy address needs to match an existing accepted domain.
EnabledPrimarySMTPAddressTemplate	Required	System.String	The *EnabledPrimarySMTPAddressTemplate* parameter

			specifies the proxy address enabled and included in an email address policy. The domain part of each proxy address needs to match an existing accepted domain.
IncludedRecipients	Required	Microsoft.Exchange.Data.Directory.Recipient.WellKnownRecipientType	The *IncludedRecipients* parameter specifies how to filter recipients used to set the email address policy. The AllRecipients value can be used only by itself. When multiple values of the *IncludedRecipients* parameter are separated by commas, the **OR** Boolean operator is applied.
Name	Required	System.String	The *Name* parameter specifies the name of the email address policy.
RecipientFilter	Required	System.String	The *RecipientFilter* parameter specifies a filter for recipients to include in the email address policy. You can't use this parameter if you use any of the following filter parameters:

			• *ConditionalCompany* • *ConditionalDepartment* • *IncludedRecipients* • *ConditionalStateOrProvince*
ConditionalCompany	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCompany* parameter specifies a filter by company. It takes multiple values as a comma-delimited list.
ConditionalCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute10	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *ConditionalCustomAttribu*

		y	*te1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute11	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use

			the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute12	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute13	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients'

			custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute14	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this

			parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute15	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all

			included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *ConditionalCustomAttribu*

		y	*te1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use

			the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute6	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute7	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients'

			custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute8	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this

			parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute9	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalDepartment	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalDepartment* parameter specifies a filter by department. It takes multiple values as a comma-delimited list.
ConditionalStateOrProvince	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalStateOrProvince* parameter specifies a

			filter by state or province. It takes multiple values as a comma-delimited list.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisabledEmailAddressTemplates	Optional	Microsoft.Exchange.Data.ProxyAddressTemplateCollection	The *DisabledEmailAddressTemplates* parameter specifies the proxy addresses included in an email address policy and are disabled. It takes multiple values as a comma-delimited list.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Co	The *Organization*

		nfiguration.Tasks.OrganizationIdParameter	parameter is reserved for internal Microsoft use.
Priority	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.EmailAddressPolicyPriority	The *Priority* parameter specifies that the current priority is higher than the priority specified as an argument for this parameter.
RecipientContainer	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientContainer* parameter specifies how to filter the recipients used to build the email address policies based on their location in Active Directory Domain Services (AD DS). The value of the *RecipientContainer* parameter can be the canonical name of an organizational unit (OU) or a domain. If you don't specify a value for the *RecipientContainer* parameter, the default search filter is the location of the dynamic distribution group in AD DS.
WhatIf	Optional	System.Management.Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-EmailAddressPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-EmailAddressPolicy** cmdlet to remove an existing email address policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-EmailAddressPolicy -Identity <EmailAddressPolicyIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the email address policy Email_Address_Policy_1.

```
Remove-EmailAddressPolicy -Identity Email_Address_Policy_1
```

Detailed Description

The **Remove-EmailAddressPolicy** cmdlet removes an existing email address policy and updates the recipients.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Email address policies" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.EmailAddressPolicyIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or email address policy name that represents a specific email address policy. You can also include the path by using the format *Path**EmailAddressPolicy*. You can omit the parameter label *Identity* so that only the email address policy name or GUID is

			supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-EmailAddressPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-EmailAddressPolicy** cmdlet to set Active Directory attributes for an email address policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-EmailAddressPolicy -Identity <EmailAddressPolicyIdParameter> [-
ConditionalCompany <MultiValuedProperty>] [-ConditionalCustomAttribute1
<MultiValuedProperty>] [-ConditionalCustomAttribute10
<MultiValuedProperty>] [-ConditionalCustomAttribute11
<MultiValuedProperty>] [-ConditionalCustomAttribute12
<MultiValuedProperty>] [-ConditionalCustomAttribute13
<MultiValuedProperty>] [-ConditionalCustomAttribute14
<MultiValuedProperty>] [-ConditionalCustomAttribute15
<MultiValuedProperty>] [-ConditionalCustomAttribute2
<MultiValuedProperty>] [-ConditionalCustomAttribute3
<MultiValuedProperty>] [-ConditionalCustomAttribute4
<MultiValuedProperty>] [-ConditionalCustomAttribute5
<MultiValuedProperty>] [-ConditionalCustomAttribute6
<MultiValuedProperty>] [-ConditionalCustomAttribute7
<MultiValuedProperty>] [-ConditionalCustomAttribute8
<MultiValuedProperty>] [-ConditionalCustomAttribute9
<MultiValuedProperty>] [-ConditionalDepartment <MultiValuedProperty>] [-
ConditionalStateOrProvince <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-DisabledEmailAddressTemplates
<ProxyAddressTemplateCollection>] [-DomainController <Fqdn>] [-
EnabledEmailAddressTemplates <ProxyAddressTemplateCollection>] [-
EnabledPrimarySMTPAddressTemplate <String>] [-ForceUpgrade
<SwitchParameter>] [-IncludedRecipients <None | MailboxUsers | Resources |
```

```
MailContacts | MailGroups | MailUsers | AllRecipients>] [-Name <String>]
[-Priority <EmailAddressPolicyPriority>] [-RecipientContainer
<OrganizationalUnitIdParameter>] [-RecipientFilter <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the email address policy's name from EMAIL_ADDRESS_POLICY01 to EMAILADDRESSPOLICY02.

```
Set-EmailAddressPolicy -Identity EMAIL_ADDRESS_POLICY01 -
Name EMAILADDRESSPOLICY02
```

EXAMPLE 2

This example edits the South East Offices email address policy that currently includes recipients in Georgia, Alabama, and Louisiana to also include recipients in Texas.

```
Set-EmailAddressPolicy -Identity "South East Offices" -
ConditionalStateorProvince
"Georgia","Alabama","Louisiana","Texas"
```

Note:
Although the email address policy is already applied to recipients in Georgia, Alabama, and Louisiana, you must include them in the parameter because the parameter overwrites values; it doesn't append new values to existing values.

Detailed Description

The **Set-EmailAddressPolicy** cmdlet doesn't apply the changes to the email address policy. Use the **Update-EmailAddressPolicy** cmdlet following a **Set-EmailAddressPolicy** cmdlet for the changes to be applied.

For more information about the **Update-EmailAddressPolicy** cmdlet, see Update-EmailAddressPolicy.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Email address policies" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.EmailAddressPolicyIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or email address policy name that represents a specific email address policy. You can also include the path using the format *Path\EmailAddressPolicy*. You can omit the parameter label so that only the email address policy name or GUID is supplied.
ConditionalCompany	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCompany* parameter specifies the company for the email address policy. It takes multiple values as a comma-delimited list. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribu*

			te15 parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute10	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a

			Conditional parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute11	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute12	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the

			ConditionalCustomAttribute1 value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute13	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.

ConditionalCustomAttribute14	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute15	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is

			`Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribu*

			te15 parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a

			Conditional parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute6	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the

			ConditionalCustomAttribute1 value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute7	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.

ConditionalCustomAttribute8	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute9	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is

			`Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalDepartment	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalDepartment* parameter specifies the department for the email address policy. It takes multiple values as a comma-delimited list. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalStateOrProvince	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalStateOrProvince* parameter specifies the state or province for the email address policy. It takes multiple values as a comma-delimited list. You must use the *IncludedRecipients* parameter if you use a

			Conditional parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisabledEmailAddressTemplates	Optional	Microsoft.Exchange.Data.ProxyAddressTemplateCollection	The *DisabledEmailAddressTemplates* parameter specifies the proxy addresses included in an email address policy that are disabled. It takes multiple values as a comma-delimited list.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

EnabledEmailAddressTemplates	Optional	Microsoft.Exchange.Data.ProxyAddressTemplateCollection	The *EnabledEmailAddressTemplates* parameter specifies the proxy addresses included in an email address policy that are enabled. It takes multiple values as a comma-delimited list.
EnabledPrimarySMTPAddressTemplate	Optional	System.String	The *EnabledPrimarySMTPAddressTemplate* parameter specifies the proxy addresses included in an email address policy that are enabled. It takes multiple values as a comma-delimited list.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* switch specifies whether to suppress the following confirmation: "To save changes on object, the object must be upgraded to the current Exchange version. After upgrade, this object cannot be managed by a previous version of Exchange System Manager. Do you want to continue to upgrade and save the

			object?"
IncludedRecipients	Optional	Microsoft.Exchange.Data.Directory.Recipient.WellKnownRecipientType	The *IncludedRecipients* parameter filters the recipient types used to build the global address list (GAL). The available values for the *IncludedRecipients* parameter include the following: • `None` • `MailboxUsers` • `Resources` • `MailContacts` • `MailGroups` • `MailUsers` • `AllRecipients` The `AllRecipients` value can be used only by itself. When multiple values of the *IncludedRecipients* parameter are separated by commas, the **OR** Boolean operator is applied. You must use this parameter if you're using any of the *Conditional* parameters. You can't use this parameter if you use the *RecipientFilter* parameter.
Name	Optional	System.String	The *Name* parameter specifies the new name

			for the email address policy.
Priority	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.EmailAddressPolicyPriority	The *Priority* parameter specifies that the current priority is higher than the priority specified as an argument for this parameter.
RecipientContainer	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientContainer* parameter filters the recipients used to build the email address policy based on their location in Active Directory Domain Services (AD DS). The value of the *RecipientContainer* parameter can be the canonical name of an organizational unit (OU) or a domain. If you don't specify a value for the *RecipientContainer* parameter, the default search filter is the location of the dynamic distribution group in AD DS.
RecipientFilter	Optional	System.String	The *RecipientFilter* parameter filters the recipients contained in a particular email address

			policy. The *RecipientFilter* parameter can't be used if any of the following parameters are specified: • *ConditionalCompany* • *ConditionalDepartment* • *ConditionalStateOrProvince* • *IncludedRecipients*
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-EmailAddressPolicy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Update-EmailAddressPolicy** cmdlet to apply an email address policy to all recipients.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-EmailAddressPolicy -Identity <EmailAddressPolicyIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-FixMissingAlias
<SwitchParameter>] [-UpdateSecondaryAddressesOnly <SwitchParameter>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example applies the email address policy EMAIL_ADDRESS_POLICY01 to all affected recipients.

```
Update-EmailAddressPolicy -Identity EMAIL_ADDRESS_POLICY01
```

Detailed Description

The **Update-EmailAddressPolicy** cmdlet queries for all recipients that match the specified email address policy and saves the objects to Active Directory. Use the **Update-EmailAddressPolicy** cmdlet after you use the **Set-EmailAddressPolicy** cmdlet to apply all changes. For more information about the **Set-EmailAddressPolicy** cmdlet, see Set-EmailAddressPolicy.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Email address policies" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.EmailAddressPolicyIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or email address policy name that

			represents a specific email address policy. You can also include the path by using the format *Path\EmailAddressPolicy*. You can omit the parameter label *Identity* so that only the email address policy name or GUID is supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

FixMissingAlias	Optional	System.Management.Automation.SwitchParameter	The *FixMissingAlias* switch repairs recipients that don't have an alias. The alias is generated based on the name of the recipient. You need to use the *FixMissingAlias* parameter if you receive an error message when you attempt to update an email address policy, global address list, or address list.
UpdateSecondaryAddressesOnly	Optional	System.Management.Automation.SwitchParameter	The *UpdateSecondaryAddressesOnly* switch parameter specifies to update the secondary email addresses only. If you specify this parameter, the primary proxy email address isn't updated.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes

			would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-GlobalAddressList

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-GlobalAddressList** cmdlet to return all the attributes of a global address list (GAL) or a set of GALs.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-GlobalAddressList -DefaultOnly <SwitchParameter> <COMMON PARAMETERS>

Get-GlobalAddressList [-Identity <GlobalAddressListIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example doesn't use parameters and returns all the attributes of all GALs.

```
Get-GlobalAddressList
```

EXAMPLE 2

This example uses the *DefaultOnly* parameter to return all the attributes of the default GAL only.

```
Get-GlobalAddressList -DefaultOnly
```

Detailed Description

The **Get-GlobalAddressList** cmdlet is mainly used to populate the GAL property pages in the Exchange Administration Center. This command doesn't provide a filtering capability. If filtering is required, you should use a WHERE clause with the command.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Global address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
DefaultOnly	Required	System.Management.Automation.SwitchParameter	The *DefaultOnly* parameter specifies the default GAL only.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Identity	Optional	Microsoft.Exchange.Configuration.Tasks.GlobalAddressListIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or GAL name that represents a specific address list. You can also include the path by using the format *Path\GlobalAddressListName*. You can omit the parameter label so that only the GAL name or GUID is supplied.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-GlobalAddressList

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-GlobalAddressList** cmdlet to create a global address list (GAL).

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-GlobalAddressList [-RecipientFilter <String>] <COMMON PARAMETERS>


New-GlobalAddressList [-ConditionalCompany <MultiValuedProperty>] [-
ConditionalCustomAttribute1 <MultiValuedProperty>] [-
ConditionalCustomAttribute10 <MultiValuedProperty>] [-
ConditionalCustomAttribute11 <MultiValuedProperty>] [-
ConditionalCustomAttribute12 <MultiValuedProperty>] [-
ConditionalCustomAttribute13 <MultiValuedProperty>] [-
ConditionalCustomAttribute14 <MultiValuedProperty>] [-
ConditionalCustomAttribute15 <MultiValuedProperty>] [-
ConditionalCustomAttribute2 <MultiValuedProperty>] [-
ConditionalCustomAttribute3 <MultiValuedProperty>] [-
ConditionalCustomAttribute4 <MultiValuedProperty>] [-
ConditionalCustomAttribute5 <MultiValuedProperty>] [-
ConditionalCustomAttribute6 <MultiValuedProperty>] [-
ConditionalCustomAttribute7 <MultiValuedProperty>] [-
ConditionalCustomAttribute8 <MultiValuedProperty>] [-
ConditionalCustomAttribute9 <MultiValuedProperty>] [-ConditionalDepartment
<MultiValuedProperty>] [-ConditionalStateOrProvince <MultiValuedProperty>]
[-IncludedRecipients <None | MailboxUsers | Resources | MailContacts |
MailGroups | MailUsers | AllRecipients>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>] [-
RecipientContainer <OrganizationalUnitIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the GAL NewGAL.

```
New-GlobalAddressList -Name NewGAL
```

EXAMPLE 2

This example creates the GAL_AgencyB GAL by using the *RecipientFilter* parameter to include all mailbox users whose custom attribute 15 equals AgencyB.

```
New-GlobalAddressList -Name GAL_AgencyB -RecipientFilter
{((RecipientType -eq "UserMailbox") -and (CustomAttribute15
-eq "AgencyB"))}
```

Detailed Description

The GAL is created under the Microsoft Exchange Organization configuration container in Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Global address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name of the new GAL.
ConditionalCompany	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCompany* parameter specifies the company attribute on the GAL.
ConditionalCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use

			the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute10	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute11	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients'

			custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute12	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this

			parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute13	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute14	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all

			included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute15	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *ConditionalCustomAttribu*

		y	*te1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use

			the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients'

			custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute6	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this

			parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute7	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute8	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all

			included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute9	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalDepartment	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *ConditionalDepartment*

		y	parameter specifies the department attribute on the GAL.
ConditionalStateOrProvince	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalStateOrProvince* parameter specifies the attribute in the GAL for the state or province. It takes multiple values as a comma-delimited list.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IncludedRecipients	Optional	Microsoft.Exchange.Data.Directory.Recipient.WellKnownRecipientT	The *IncludedRecipients* parameter sets the included recipients in the

		ype	GAL.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
RecipientContainer	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientContainer* parameter filters the recipients used to build the dynamic distribution group based on their location in Active Directory. The value of the *RecipientContainer* parameter can be the canonical name of an organizational unit (OU) or a domain. If a value for the *RecipientContainer* parameter isn't specified, the default search filter is the location of the dynamic distribution group in Active Directory.
RecipientFilter	Optional	System.String	The *RecipientFilter* parameter specifies the recipients contained in the GAL. The *RecipientFilter* parameter can't be used if any of the following parameters are specified: • *IncludedRecipients* • *ConditionalCompany* • *ConditionalDepartment*

			• *ConditionalStateOrProvince*
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-GlobalAddressList

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-GlobalAddressList** cmdlet to remove an existing global address list (GAL).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-GlobalAddressList -Identity <GlobalAddressListIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the GAL OldGAL.

```
Remove-GlobalAddressList OldGAL
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.GlobalAddressListIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or GAL name that represents a specific address list. You can also include the path by using the format *Path\GlobalAddressListName*. You can omit the parameter label so that only the GAL name or

			GUID is supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-GlobalAddressList

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-GlobalAddressList** cmdlet to modify the attributes in Active Directory for a global address list (GAL).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-GlobalAddressList -Identity <GlobalAddressListIdParameter> [-
ConditionalCompany <MultiValuedProperty>] [-ConditionalCustomAttribute1
<MultiValuedProperty>] [-ConditionalCustomAttribute10
<MultiValuedProperty>] [-ConditionalCustomAttribute11
<MultiValuedProperty>] [-ConditionalCustomAttribute12
<MultiValuedProperty>] [-ConditionalCustomAttribute13
<MultiValuedProperty>] [-ConditionalCustomAttribute14
<MultiValuedProperty>] [-ConditionalCustomAttribute15
<MultiValuedProperty>] [-ConditionalCustomAttribute2
<MultiValuedProperty>] [-ConditionalCustomAttribute3
<MultiValuedProperty>] [-ConditionalCustomAttribute4
<MultiValuedProperty>] [-ConditionalCustomAttribute5
<MultiValuedProperty>] [-ConditionalCustomAttribute6
<MultiValuedProperty>] [-ConditionalCustomAttribute7
<MultiValuedProperty>] [-ConditionalCustomAttribute8
<MultiValuedProperty>] [-ConditionalCustomAttribute9
<MultiValuedProperty>] [-ConditionalDepartment <MultiValuedProperty>] [-
ConditionalStateOrProvince <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ForceUpgrade
<SwitchParameter>] [-IncludedRecipients <None | MailboxUsers | Resources |
MailContacts | MailGroups | MailUsers | AllRecipients>] [-Name <String>]
[-RecipientContainer <OrganizationalUnitIdParameter>] [-RecipientFilter
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example assigns a new name, GALwithNewName, to the GAL with the GUID 96d0c505-eba8-4103-ad4f-577a1bf4ad7b.

```
Set-GlobalAddressList 96d0c505-eba8-4103-ad4f-577a1bf4ad7b
-Name GALwithNewName
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Global address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.GlobalAddressListIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or GAL name that represents a specific GAL. You can also include the path by using the format *Path**GlobalAddressListName*. You can omit the parameter label so that only the GAL name or GUID is supplied.

ConditionalCompany	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCompany* parameter specifies a company. You can use multiple values as a comma-delimited list. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.

ConditionalCustomAttribute10	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute11	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is

			`Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute12	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute13	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribu*

			te15 parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute14	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a

			Conditional parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute15	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the

			ConditionalCustomAttribute1 value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.

ConditionalCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is

			`Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute6	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute7	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribu*

			te15 parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute8	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a

			Conditional parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute9	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalDepartment	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalDepartment* parameter specifies a department. All GALs with a department attribute that matches the value that you input for this parameter are included.

			You can use multiple values as a comma-delimited list. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalStateOrProvince	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalStateOrProvince* parameter specifies the state or province for the GAL. You can use multiple values as a comma-delimited list. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* parameter suppresses the following confirmation: "To save changes on object, the object must be upgraded to the current Exchange version. After upgrade, this object cannot be managed by a previous version of Exchange System Manager. Do you want to continue to upgrade and save the object?"
IncludedRecipients	Optional	Microsoft.Exchange.Data.Directory.Recipient.WellKnownRecipientType	The *IncludedRecipients* parameter filters the recipient types that are used to build the GAL. The available values for the *IncludedRecipients* parameter are `None`, `AllRecipients`, `MailboxUsers`, `MailUsers`, `Resources`, `MailContacts`,

			and `MailGroups`. The `AllRecipients` value can be used only by itself. When multiple values of the *IncludedRecipients* parameter are separated by commas, the **OR** Boolean operator is applied. You must use this parameter if you're using any of the *Conditional* parameters. You can't use this parameter if you use the *RecipientFilter* parameter.
Name	Optional	System.String	The *Name* parameter specifies the new name of the GAL.
RecipientContainer	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientContainer* parameter filters the recipients used to build the dynamic distribution group based on their location in Active Directory. The value of the *RecipientContainer* parameter can be the canonical name of an organizational unit (OU) or a domain. If a value for

			the *RecipientContainer* parameter isn't specified, the default search filter is the location of the dynamic distribution group in Active Directory.
RecipientFilter	Optional	System.String	The *RecipientFilter* parameter specifies a filter for recipients to include in the GAL. You can't use this parameter if you use any of the following filter parameters: • *ConditionalCompany* • *ConditionalDepartment* • *IncludedRecipients* • *ConditionalStateOrProvince* You must use either this parameter or one of the previously listed filter parameters.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-GlobalAddressList

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Update-GlobalAddressList** cmdlet to update a global address list (GAL).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-GlobalAddressList -Identity <GlobalAddressListIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example updates the default GAL.

```
Update-GlobalAddressList -Identity "Global Address List"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address lists" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.GlobalAddressListIdParameter	The *Identity* parameter specifies a unique identifier of the GAL being updated. These unique identifiers include the common name (CN), GUID, or distinguished name (DN).
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-OabVirtualDirectory

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-OabVirtualDirectory** cmdlet to return configuration information about offline address book (OAB) distribution points.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-OabVirtualDirectory -Server <ServerIdParameter> <COMMON PARAMETERS>

Get-OabVirtualDirectory [-Identity <VirtualDirectoryIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-ADPropertiesOnly <SwitchParameter>] [-
DomainController <Fqdn>] [-ShowBackEndVirtualDirectories
<SwitchParameter>] [-ShowMailboxVirtualDirectories <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns all OAB virtual directory web distribution points.

```
Get-OabVirtualDirectory
```

EXAMPLE 2

This example returns all OAB virtual directory web distribution points on the server CAS-01-007.

```
Get-OabVirtualDirectory -Server CAS-01-007
```

EXAMPLE 3

This example returns OAB virtual directories on Client Access servers. By default, this cmdlet only returns virtual directories on Mailbox servers.

```
Get-OabVirtualDirectory -ShowBackEndVirtualDirectories
```

Detailed Description

The **Get-OabVirtualDirectory** cmdlet queries a distribution point by identity or by web distribution points on a specific server, or queries for all web distribution points. By default, this cmdlet only returns virtual directories on Mailbox servers. To view virtual directories on Client Access servers, use the *ShowBackEndVirtualDirectories* parameter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Offline address book connectivity" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which to perform the selected operation. You can't use this parameter in conjunction with the *Identity* parameter.
ADPropertiesOnly	Optional	System.Management.Automation.SwitchParameter	The *ADPropertiesOnly* switch specifies whether to return only the properties about the virtual directory stored in Active Directory. The properties stored in the Internet Information Services (IIS) metabase aren't returned.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from

			Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or OAB virtual directory name that represents a specific OAB virtual directory. You can also include the path using the format *Server\OfflineAddressBookVirtualDirectoryName*. You can omit the parameter label so that only the OAB virtual directory name, DN, or GUID is supplied. You can't use this parameter in conjunction with the *Server* parameter.
ShowBackEndVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowBackEndVirtualDirectories* switch specifies that OAB virtual directories on Client Access servers are returned. If you don't use this switch, only virtual directories

			on Mailbox servers are returned.
ShowMailboxVirtualDirectories	Optional	System.Management.Automation.SwitchParameter	The *ShowMailboxVirtualDirectories* switch specifies whether the OAB virtual directories on Mailbox servers are returned. This switch should only be used with the direction of Microsoft support.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-OabVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-OABVirtualDirectory** cmdlet to configure a server as a web distribution point for an offline address book (OAB).

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-OabVirtualDirectory [-AppPoolId <String>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ExtendedProtectionFlags
<MultiValuedProperty>] [-ExtendedProtectionSPNList <MultiValuedProperty>]
[-ExtendedProtectionTokenChecking <None | Allow | Require>] [-ExternalUrl
<Uri>] [-InternalUrl <Uri>] [-Path <String>] [-PollInterval <Int32>] [-
Recovery <SwitchParameter>] [-RequireSSL <$true | $false>] [-Role
<ClientAccess | Mailbox>] [-Server <ServerIdParameter>] [-WebSiteName
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates an OAB virtual directory on CASServer01 and configures the distribution service to poll the generation server every two hours.

```
New-OABVirtualDirectory -Server CASServer01 -PollInterval
120
```

Detailed Description

The **New-OABVirtualDirectory** cmdlet configures a web distribution point for an OAB and creates the OAB virtual directory.

Note:
You have to manually create the file system folder on the server that hosts the OAB files.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the " Offline address book connectivity " entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
AppPoolId	Optional	System.String	The *AppPoolId* parameter specifies the pool of programs that can be used with the OAB virtual directory.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy

			mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN

			list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionToke*

			nChecking parameter with the `AllowDotlessSPN value`. You specify the domain in SPN format. The SPN format is <*protocol*>/<*FQDN*>. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenCheckingMode	The *ExtendedProtectionTokenChecking* parameter defines how you want to use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for

			Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the client and server will work, but may not be as secure as a connection using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and

			Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the external URL that the OAB virtual directory points to.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the internal URL that the OAB virtual directory points to.
Path	Optional	System.String	The *Path* parameter specifies the path to the

			folder that hosts the OAB virtual directory.
PollInterval	Optional	System.Int32	The *PollInterval* parameter specifies the time interval (in minutes) that the distribution service should poll the generation server for new updates.
Recovery	Optional	System.Management.Automation.SwitchParameter	The *Recovery* parameter specifies whether to support Setup recovery mode and is used implicitly by Setup when recovery is performed. It's never necessary to specify this parameter.
RequireSSL	Optional	System.Boolean	The *RequireSSL* parameter specifies whether to require Secure Sockets Layer (SSL) to access the OAB virtual directory. If set to `$true`, the *RequireSSL* parameter requires SSL to access the OAB virtual directory.
Role	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.VirtualDirectoryRole	The *Role* parameter specifies the configuration that should be used when the virtual directory is created. The following are

			the values that can be used with this parameter: • `FrontEnd` Configure the virtual directory for use on a Client Access server. • `BackEnd` Configure the virtual directory for use on a Mailbox server.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name of the server to connect. Enter the *Server* parameter as a host name or an FQDN. If this parameter isn't used, the local server is queried.
WebSiteName	Optional	System.String	The *WebSiteName* parameter specifies the name of the Internet Information Services (IIS) website under which to create the virtual directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-OabVirtualDirectory

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-OabVirtualDirectory** cmdlet to remove a server from the offline address book (OAB) distribution points list.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-OabVirtualDirectory -Identity <VirtualDirectoryIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Force
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the default OAB virtual directory from the local server.

```
Remove-OabVirtualDirectory -Identity "OAB (Default Web
Site)"
```

EXAMPLE 2

This example removes the default OAB virtual directory from Server1.

```
Remove-OabVirtualDirectory -Identity "Server1\OAB (Default
Web Site)"
```

Detailed Description

Some situations require the removal of an OAB virtual directory. For example, to uninstall a Client Access server that contains an OAB distribution points list, you must remove the OAB virtual directory and then re-create it on another Client Access server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or OAB virtual directory name that represents a specific OAB virtual directory. You can also include the path by using the format *Server\OfflineAddressBookVirtualDirectoryName*. You can omit the parameter label so that only the OAB virtual directory name or GUID is supplied.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're

			prompted for administrative input. You don't have to specify a value with this parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-OabVirtualDirectory

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-OABVirtualDirectory** cmdlet to change configuration settings for an offline address book (OAB) virtual directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-OabVirtualDirectory -Identity <VirtualDirectoryIdParameter> [-
BasicAuthentication <$true | $false>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-ExtendedProtectionFlags <MultiValuedProperty>]
[-ExtendedProtectionSPNList <MultiValuedProperty>] [-
ExtendedProtectionTokenChecking <None | Allow | Require>] [-ExternalUrl
<Uri>] [-InternalUrl <Uri>] [-OAuthAuthentication <$true | $false>] [-
PollInterval <Int32>] [-RequireSSL <$true | $false>] [-WhatIf
[<SwitchParameter>]] [-WindowsAuthentication <$true | $false>]
```

Examples

EXAMPLE 1

This example changes the external URL of the OAB virtual directory OAB (Default Web Site) to https://www.contoso.com/OAB.

```
Set-OABVirtualDirectory -Identity "Server1\OAB (Default Web
Site)" -ExternalUrl "https://www.contoso.com/OAB"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Offline address book connectivity" entry in the Email address and address book permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or OAB virtual directory name that represents a specific OAB virtual

			directory. You can also include the path by using the format *Server\OfflineAddressBookVirtualDirectoryName*.
BasicAuthentication	Optional	System.Boolean	The *BasicAuthentication* parameter specifies whether Basic authentication is enabled on the OAB virtual directory. This parameter can be used with the *WindowsAuthentication* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory.
ExtendedProtectionFlags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionFlags* parameter is used to customize the options you use if you're using Extended Protection for Authentication. The possible values are: • **None** Default setting. • **Proxy** Specifies that a proxy is terminating the SSL channel. A Service Principal Name (SPN) must be registered in the *ExtendedProtectionSPNList* parameter if proxy mode is configured. • **ProxyCoHosting** Specifies that both HTTP and HTTPS traffic may be accessing the Client Access server and that a proxy is located between at least some of the clients and the Client Access server. • **AllowDotlessSPN** Specifies whether you want to support valid SPNs that aren't in the fully qualified domain

			name (FQDN) format, for example ContosoMail. You specify valid SPNs with the *ExtendedProtectionSPNList* parameter. This option makes extended protection less secure because dotless certificates aren't unique, so it isn't possible to ensure that the client-to-proxy connection was established over a secure channel. • **NoServiceNameCheck** Specifies that the SPN list won't be checked to validate a channel binding token. This option makes Extended Protection for Authentication less secure. We generally don't recommend this setting.
ExtendedProtectionSPNList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtendedProtectionSPNList* parameter specifies a list of valid Service Principal Names (SPNs) if you're

			using Extended Protection for Authentication on the specified virtual directory. The possible values are: • **Null** This is the default value. • **Single SPN or comma delimited list of valid SPNs** By default, you must specify the fully qualified domain name (FQDN) (for example mail.contoso.com) for each SPN. If you want to add an SPN that's not an FQDN (for example, ContosoMail), you must also use the *ExtendedProtectionTokenChecking* parameter with the `AllowDotlessSPN value`. You specify the domain in SPN format. The SPN format is *<protocol>/<FQDN>*. For example, a valid entry could be HTTP/mail.contoso.com.
ExtendedProtectionTokenChecking	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionTokenChecki	The *ExtendedProtectionTokenChecking* parameter defines how you want to

		ngMode	use Extended Protection for Authentication on the specified Exchange virtual directory. Extended Protection for Authentication isn't enabled by default. The available settings are: • **None** Extended Protection for Authentication won't be used. Connections between the client and Exchange won't use Extended Protection for Authentication on this virtual directory. This is the default setting. • **Allow** Extended Protection for Authentication will be used for connections between the client and Exchange on this virtual directory if both the client and server support Extended Protection for Authentication. Connections that don't support Extended Protection for Authentication on the

			client and server will work, but may not be as secure as a connection using Extended Protection for Authentication. **Note:** If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more Service Principal Names (SPNs) by using the *ExtendedProtectionSPNList* parameter. • **Require** Extended Protection for Authentication will be used for all connections between clients and Exchange servers for this virtual directory. If either the client or server doesn't support Extended Protection for Authentication, the connection between the client and server will fail. If you set this option, you must also set a value for the *ExtendedProtectionSPNList* parameter.

			Note: If you have a proxy server between the client and the Client Access server that's configured to terminate the client-to-proxy SSL channel, you must also configure one or more SPNs using the parameter *ExtendedProtectionSPNList*.
ExternalUrl	Optional	System.Uri	The *ExternalUrl* parameter specifies the external URL that the OAB virtual directory points to.
InternalUrl	Optional	System.Uri	The *InternalUrl* parameter specifies the internal URL that the OAB virtual directory points to.
OAuthAuthentication	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
PollInterval	Optional	System.Int32	The *PollInterval* parameter specifies the time interval that the distribution service should poll the generation server for new updates (in minutes).
RequireSSL	Optional	System.Boolean	The *RequireSSL* parameter specifies whether to require Secure Sockets Layer (SSL) when accessing the OAB virtual

			directory. When set to `$true`, the *RequireSSL* parameter requires SSL when accessing the OAB virtual directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsAuthentication	Optional	System.Boolean	The *WindowsAuthentication* parameter specifies whether Integrated Windows authentication is permitted on the OAB virtual directory. This parameter can be used with the *BasicAuthentication* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-OfflineAddressBook

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-OfflineAddressBook** cmdlet to obtain the settings of one or more offline address books (OABs).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-OfflineAddressBook -Server <ServerIdParameter> <COMMON PARAMETERS>

Get-OfflineAddressBook [-Identity <OfflineAddressBookIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example uses the **Get-OfflineAddressBook** command without parameters to obtain all the configuration details of all the existing OABs.

```
Get-OfflineAddressBook | Format-List
```

Detailed Description

If a parameter isn't passed with the **Get-OfflineAddressBook** cmdlet, the command returns all of the OABs in the organization. If the *Identity* parameter is passed, the command returns the OAB for the specified identity. If the *Server* parameter is passed, the command returns the OABs for the specified server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the GUID, domain name, or fully qualified domain name (FQDN) of the server generating the OAB.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OfflineAddressBookIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or OAB name that represents a specific OAB. You can also include the path using the format *Server\OfflineAddressBookName*.

			You can omit the parameter label *Identity* so that only the OAB name or GUID is supplied.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-OfflineAddressBook

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-OfflineAddressBook** cmdlet to create an offline address book (OAB).

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-OfflineAddressBook -AddressLists <AddressBookBaseIdParameter[]> -Name
<String> [-Confirm [<SwitchParameter>]] [-DiffRetentionPeriod <Unlimited>]
[-DomainController <Fqdn>] [-GeneratingMailbox <MailboxIdParameter>] [-
GlobalWebDistributionEnabled <$true | $false>] [-IsDefault <$true |
$false>] [-Organization <OrganizationIdParameter>] [-VirtualDirectories
<VirtualDirectoryIdParameter[]>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example uses two commands to create the OAB named OAB_AgencyB that includes all address lists where AgencyB is part of the name. With the settings shown, an OAB is generated by myserver.contoso.com on Mondays and Wednesdays from 01:00 (1:00 A.M.) to 02:00 (2:00 A.M.). The command also creates the default OAB for the organization.

```
$a = Get-AddressList | Where {$_.Name -Like "*AgencyB*"
New-OfflineAddressBook -Name "OAB_AgencyB" -AddressLists $a
-Schedule "Mon.01:00-Mon.02:00, Wed.01:00-Wed.02:00"
```

EXAMPLE 2

This example creates the OAB New OAB that uses Web-based distribution for Microsoft Office Outlook 2007 or later by using the default virtual directory.

```
New-OfflineAddressBook -Name "New OAB" -AddressLists
"\Default Global Address List" -VirtualDirectories
"SERVER01\OAB (Default Web Site)"
```

Detailed Description

The **New-OfflineAddressBook** cmdlet allows administrators to create OABs. For example, if you uninstall a Client Access server that contains an OAB, you need to re-create it on another Client Access server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
AddressLists	Required	Microsoft.Exchange.Configuration.Tasks.Addre	The *AddressLists* parameter specifies an

		ssBookBaseIdParameter[]	array of address list identities included in the OAB.
Name	Required	System.String	The *Name* parameter specifies the name that describes the OAB object in Exchange System Manager. This value can contain a maximum of 64 characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DiffRetentionPeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *DiffRetentionPeriod* parameter specifies the length of time, in days, that the OAB difference files are retained on the OAB-generating server and the Client Access server. To retain the

			OAB difference files indefinitely, use the value `unlimited`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
GeneratingMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *GeneratingMailbox* parameter specifies the *OABGenCapability* arbitration mailbox where the offline address books will be generated. This allows on-premises admins to load balance OAB generation.
GlobalWebDistributionEnabled	Optional	System.Boolean	The *GlobalWebDistributionEnabled* parameter specifies whether distribution occurs to all virtual directories in the organization. If the value of the *GlobalWebDistributionEnabled* parameter is

			`$true`, distribution occurs to all virtual directories in the organization and the *VirtualDirectories* parameter can't be used.
IsDefault	Optional	System.Boolean	The *IsDefault* parameter specifies whether the OAB is set as the default OAB for all new mailbox databases. If the value of the *IsDefault* parameter is `$true`, the OAB is set as the default OAB for all new mailbox databases. If the value is `$false`, the OAB isn't set as the default OAB.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
VirtualDirectories	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter[]	The *VirtualDirectories* parameter specifies the array of **OABVirtualDirectory** objects. If the *VirtualDirectories* parameter is specified,

			`version4` of the OAB must be generated.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Move-OfflineAddressBook

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Move-OfflineAddressBook** cmdlet to designate a new server responsible for generating the offline address book (OAB).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Move-OfflineAddressBook -Identity <OfflineAddressBookIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Server
<ServerIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example moves OAB generation to the server Server1.

```
Move-OfflineAddressBook -Identity "My OAB" -Server
"Server1"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.OfflineAddressBookIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or OAB name that represents a specific OAB. You can also include the path by using the format *Server\OfflineAddressBookName*.

			You can omit the parameter label so that only the OAB name or GUID is supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Mailbox server on which to perform the selected operation.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-OfflineAddressBook

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-OfflineAddressBook** cmdlet to remove (delete) offline address books (OABs).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-OfflineAddressBook -Identity <OfflineAddressBookIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Force
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the OAB My OAB from Active Directory.

```
Remove-OfflineAddressBook -Identity "\My OAB"
```

Detailed Description

The **Remove-OfflineAddressBook** cmdlet removes an existing OAB. For example, to uninstall a Client Access server that contains an OAB, you have to remove the OAB and re-create it on another Client Access server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.OfflineAddressBookIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or OAB name that represents a specific OAB. You can also include the path by using the format *Server\OfflineAddressBookName*. You can omit the parameter label so that

			only the OAB name or GUID is supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the

			Force switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-OfflineAddressBook

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-OfflineAddressBook** cmdlet to modify offline address book (OAB) settings.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-OfflineAddressBook -Identity <OfflineAddressBookIdParameter> [-
AddressLists <AddressBookBaseIdParameter[]>] [-ApplyMandatoryProperties
<SwitchParameter>] [-ConfiguredAttributes <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-DiffRetentionPeriod <Unlimited>] [-DomainController
<Fqdn>] [-GeneratingMailbox <MailboxIdParameter>] [-
GlobalWebDistributionEnabled <$true | $false>] [-IsDefault <$true |
$false>] [-MaxBinaryPropertySize <Int32>] [-
MaxMultivaluedBinaryPropertySize <Int32>] [-
MaxMultivaluedStringPropertySize <Int32>] [-MaxStringPropertySize <Int32>]
[-Name <String>] [-PublicFolderDistributionEnabled <$true | $false>] [-
Schedule <Schedule>] [-UseDefaultAttributes <SwitchParameter>] [-Versions
<MultiValuedProperty>] [-VirtualDirectories <VirtualDirectoryIdParameter[]
>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the name of the OAB.

```
Set-OfflineAddressBook -Identity "\Default Offline Address
Book" -Name "My Offline Address Book"
```

Detailed Description

The **Set-OfflineAddressBook** cmdlet modifies the settings of an existing OAB. An OAB is valuable for users who spend time disconnected from the network. An OAB is a snapshot of one or more address lists, which MAPI clients can download to provide access to the address lists while offline.

The generation of an OAB doesn't stop even if more time is required than specified in the *Schedule* parameter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn’t assigned to any role groups. To use any cmdlets that require the Address List role, you need to add the role to a role group. For details, see

the “Add a role to a role assignment policy” section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.OfflineAddressBookIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or OAB name that represents a specific OAB. You can also include the path by using the format *Server\OfflineAddressBookName*. You can omit the parameter label so that only the OAB name or GUID is supplied.
AddressLists	Optional	Microsoft.Exchange.Configuration.Tasks.AddressBookBaseIdParameter[]	The *AddressLists* parameter specifies an array of address list identities included in the OAB.
ApplyMandatoryProperties	Optional	System.Management.Automation.SwitchParameter	The *ApplyMandatoryProperties* parameter specifies whether to modify the mandatory properties of a legacy OAB to the version in Microsoft Exchange Server 2013.

ConfiguredAttributes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConfiguredAttributes* parameter specifies the attributes to be displayed for the OAB.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DiffRetentionPeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *DiffRetentionPeriod* parameter specifies the length of time, in days, that the OAB difference files are retained on the OAB-generating server and the Client Access server. To retain the OAB difference files indefinitely, use the value `Unlimited`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration

			change to Active Directory.
GeneratingMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *GeneratingMailbox* parameter specifies the *OABGenCapability* arbitration mailbox where the offline address books will be generated. This allows on-premises admins to load balance OAB generation.
GlobalWebDistributionEnabled	Optional	System.Boolean	The *GlobalWebDistributionEnabled* parameter specifies whether distribution occurs to all virtual directories in the organization. If the value of the *GlobalWebDistributionEnabled* parameter is `$true`, distribution occurs to all virtual directories in the organization. If you have *GlobalWebDistributionEnabled* set to `$true`, you can't add values to the *VirtualDirectories* parameter.
IsDefault	Optional	System.Boolean	The *IsDefault* parameter specifies whether this

			OAB is the default OAB for all new mailbox stores.
MaxBinaryPropertySize	Optional	System.Int32	The *MaxBinaryPropertySize* parameter specifies the maximum size for binary attributes before they're truncated.
MaxMultivaluedBinaryPropertySize	Optional	System.Int32	The *MaxMultivaluedBinaryPropertySize* parameter specifies the maximum size for multivalued binary attributes before they're truncated.
MaxMultivaluedStringPropertySize	Optional	System.Int32	The *MaxMultivaluedStringPropertySize* parameter specifies the maximum size for multivalued string attributes before they're truncated.
MaxStringPropertySize	Optional	System.Int32	The *MaxStringPropertySize* parameter specifies the maximum size for string attributes before they're truncated.
Name	Optional	System.String	The *Name* parameter specifies the new name of

			the OAB. A *Name* parameter value can include up to 64 characters.
PublicFolderDistributionEnabled	Optional	System.Boolean	The *PublicFolderDistributionEnabled* parameter specifies whether the OAB is to be distributed via public folders. Setting the *PublicFolderDistributionEnabled* parameter to a value of `$true` sets the OAB to be distributed via public folders. The default value is `$true`.
Schedule	Optional	Microsoft.Exchange.Data.Schedule	The *Schedule* parameter specifies the interval scheduled for generating the OAB. You can use the following values for the start and end days: • Full name of the day • Abbreviated name of the day • Integer from 0 through 6, where 0 = Sunday The start time and end time must be at least 15 minutes apart. Minutes will be rounded down to

			0, 15, 30, or 45. If you specify more than one interval, there must be at least 15 minutes between each interval. The following are examples: • "Sun.11:30 PM-Mon.1:30 AM" • 6.22:00-6.22:15 (The assistant will run from Saturday at 10:00 PM until Saturday at 10:15 PM.) • "Monday.4:30 AM-Monday.5:30 AM","Wednesday.4:30 AM-Wednesday.5:30 AM" (The assistant will run on Monday and Wednesday mornings from 4:30 until 5:30.) • "Sun.1:15 AM-Monday.23:00"
UseDefaultAttributes	Optional	System.Management.Automation.SwitchParameter	The *UseDefaultAttributes* parameter specifies whether to revert the OAB attributes to the default list.
Versions	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Versions* parameter specifies the OAB versions that are generated for

			client download. The available options are Version4, Version3, and Version2. Version4 is for Microsoft Outlook 2003 Service Pack 2 (SP2) or later clients. Version3 is for Outlook clients that support Unicode.
VirtualDirectories	Optional	Microsoft.Exchange.Configuration.Tasks.VirtualDirectoryIdParameter[]	The *VirtualDirectories* parameter specifies the array of OABVirtualDirectory objects. If the *VirtualDirectories* parameter is specified, Version4 of the OAB must be generated.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-OfflineAddressBook

Exchange Management Shell > Exchange 2013 cmdlets > Email address and address book cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Update-OfflineAddressBook** cmdlet to update the offline address books (OABs) used by Microsoft Outlook clients.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-OfflineAddressBook -Identity <OfflineAddressBookIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example updates the OAB MyOAB.

```
Update-OfflineAddressBook -Identity MyOAB
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Offline address books" entry in the Email address and address book permissions topic.

By default in Exchange Online, the Address List role isn't assigned to any role groups. To use any

cmdlets that require the Address List role, you need to add the role to a role group. For details, see the "Add a role to a role assignment policy" section of Manage role assignment policies.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.OfflineAddressBookIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or OAB name that represents a specific OAB. You can also include the path by using the format *Server\OfflineAddressBookName*. You can omit the parameter label *Identity* so that only the OAB name or GUID is supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Dat	The *DomainController*

		a.Fqdn	parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Federation and hybrid cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-08*

Federation cmdlets

Add-FederatedDomain

Remove-FederatedDomain

Get-FederatedDomainProof

Get-FederatedOrganizationIdentifier

Set-FederatedOrganizationIdentifier

Get-FederationInformation

Get-FederationTrust

New-FederationTrust

Remove-FederationTrust

Set-FederationTrust

Test-FederationTrust

Test-FederationTrustCertificate

Get-PendingFederatedDomain

Set-PendingFederatedDomain

Update-Recipient

Hybrid configuration cmdlets

Get-HybridConfiguration

New-HybridConfiguration

Remove-HybridConfiguration

Set-HybridConfiguration

Update-HybridConfiguration

Get-IntraOrganizationConfiguration

Get-IntraOrganizationConnector

New-IntraOrganizationConnector

Remove-IntraOrganizationConnector

Set-IntraOrganizationConnector

Disable-RemoteMailbox

Enable-RemoteMailbox

Get-RemoteMailbox

New-RemoteMailbox

Remove-RemoteMailbox

Set-RemoteMailbox

Add-FederatedDomain

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Add-FederatedDomain** cmdlet to configure a secondary domain with the federated organization identifier in the federation trust for the Exchange organization.

Important:
The domains being added to the federation trust must exist as accepted domains in the Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-FederatedDomain -DomainName <SmtpDomain> [-Identity
<OrganizationIdParameter>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds the domain Contoso.co.uk to the existing federation trust.

```
Add-FederatedDomain -DomainName Contoso.co.uk
```

Detailed Description

You can add any registered Internet domain to the federated organization identifier. You must prove domain ownership by creating a TXT record in the Domain Name System (DNS) zone of each domain you add.

For more details, see Federation.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainName	Required	Microsoft.Exchange.Data.SmtpDomain	The *DomainName* parameter specifies the secondary domain to be configured.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.OrganizationIdParameter	is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-FederatedDomain

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-FederatedDomain** cmdlet to remove a federated domain from the federated

organization identifier in the federation trust for the Exchange organization.

Caution:
If you remove a domain configured for federated sharing, federated sharing for that domain is disabled.

For more information, see Federation.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-FederatedDomain -DomainName <SmtpDomain> [-Identity
<OrganizationIdParameter>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Force <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the federated domain contoso.co.uk from the federated organization identifier.

```
Remove-FederatedDomain -DomainName contoso.co.uk
```

Detailed Description

An Exchange organization's federated organization identifier is generally created using the organization's primary domain name. Additional domain names can be added and removed. The **Remove-FederatedDomain** cmdlet removes a federated domain from the federated organization identifier.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainName	Required	Microsoft.Exchange.Data.SmtpDomain	The *DomainName* parameter specifies the federated domain name to be removed from the federated

			organization identifier.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages when removing a federated domain. This parameter can be used when the removal of the federated domain from Windows Live fails, but the

			configuration of this domain as a federated domain in Exchange should be removed regardless. The result of running this task with the *Force* switch is that the Exchange configuration is removed but the domain may not be released in Windows Live. We recommend that you not use the *Force* switch unless the release of the domain from Windows Live continues to fail.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-FederatedDomainProof

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-FederatedDomainProof** cmdlet to generate a cryptographically secure string for the domain used for federated sharing in your Microsoft Exchange Server 2013 organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-FederatedDomainProof -DomainName <SmtpDomain> [-DomainController
<Fqdn>] [-Thumbprint <String>]
```

Examples

EXAMPLE 1

This example generates a cryptographically secure string for the domain contoso.com.

```
Get-FederatedDomainProof -DomainName "contoso.com"
```

EXAMPLE 2

This example uses a specific certificate for the domain contoso.com.

```
Get-FederatedDomainProof -DomainName "contoso.com" -
Thumbprint AC00F35CBA8359953F4126E0984B5CCAFA2F4F17
```

Detailed Description

The **Get-FederatedDomainProof** cmdlet generates a cryptographically secure string for the domain used for federated sharing. The resulting string is used to manually configure a text (TXT) record in the Domain Name System (DNS) zone for the domain used by the administrator when running the cmdlet. A TXT record needs to be added to DNS for all accepted domains used for federated sharing. If the thumbprint of a certificate isn't provided, the task generates strings for all the certificates currently configured for the federation trust. Upon initial configuration of federated sharing, the proof string generated for the current certificate needs to be put into the TXT record for the federated domain in DNS. We recommend you update the TXT records whenever a new certificate is configured for the federation trust.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainName	Required	Microsoft.Exchange.Data.SmtpDomain	The *DomainName* parameter specifies the domain name for which the cryptographically secure string is generated.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Thumbprint	Optional	System.String	The *Thumbprint* parameter specifies the thumbprint of an existing certificate.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-FederatedOrganizationIdentifier

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-FederatedOrganizationIdentifier** cmdlet to retrieve the Exchange organization's federated organization identifier and related details, such as federated domains, organization contact, and status.

For more information, see Federation.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-FederatedOrganizationIdentifier [-Identity <OrganizationIdParameter>]
[-DomainController <Fqdn>] [-IncludeExtendedDomainInfo <SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves the Exchange organization's federated organization identifier.

```
Get-FederatedOrganizationIdentifier
```

EXAMPLE 2

This example retrieves the Exchange organization's federated organization identifier. The *IncludeExtendedDomainInfo* switch is used to return the status of federated domains from the Microsoft Federation Gateway.

```
Get-FederatedOrganizationIdentifier -
IncludeExtendedDomainInfo
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter specifies the organization ID.
IncludeExtendedDomai	Optional	System.Management.A	The

nInfo		utomation.SwitchParameter	*IncludeExtendedDomainInfo* switch specifies that the command query Microsoft Federation Gateway for the status of each accepted domain that's federated. The status is returned with each domain in the **Domains** property.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-FederatedOrganizationIdentifier

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-FederatedOrganizationIdentifier** cmdlet to configure the federated organization identifier for the Exchange organization.

For more details, see Federation.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-FederatedOrganizationIdentifier [-Identity <OrganizationIdParameter>]
[-AccountNamespace <SmtpDomain>] [-Confirm [<SwitchParameter>]] [-
DefaultDomain <SmtpDomain>] [-DelegationFederationTrust
<FederationTrustIdParameter>] [-DomainController <Fqdn>] [-Enabled <$true
| $false>] [-OrganizationContact <SmtpAddress>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures a federated organization identifier for the Exchange organization.

```
Set-FederatedOrganizationIdentifier -
DelegationFederationTrust "Microsoft Federation Gateway" -
AccountNamespace "Contoso.com" -Enabled $true
```

EXAMPLE 2

This example temporarily disables federation for the Exchange organization.

```
Set-FederatedOrganizationIdentifier -Enabled $false
```

EXAMPLE 3

This example enables the organization identifier. This enables federation for the Exchange organization.

```
Set-FederatedOrganizationIdentifier -Enabled $true
```

Detailed Description

You must configure a federated organization identifier to create an account namespace for your Exchange organization with the Microsoft Federation Gateway and enable federation for the purpose of sharing calendars or contacts, accessing free/busy information across Exchange organizations, and securing cross-premises email delivery using federated delivery. When you create a federation trust, a value for the *AccountNamespace* parameter is automatically created with the Microsoft Federation Gateway. The *AccountNamespace* parameter is a combination of a pre-defined string and the domain specified. For example, if you specify the federated domain contoso.com as the domain, "FYDIBOHF25SPDLT.contoso.com" is automatically created as the value for the *AccountNamespace* parameter. You can add and remove Additional domain names later by using the **Add-FederatedDomain** and **Remove-FederatedDomain** cmdlets.

You can temporarily disable federation by disabling the organization identifier.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
AccountNamespace	Optional	Microsoft.Exchange.Data.SmtpDomain	The *AccountNamespace* parameter specifies the federated domain to be used to establish the organization identifier with the Microsoft Federation Gateway.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DefaultDomain	Optional	Microsoft.Exchange.Data.SmtpDomain	The *DefaultDomain* parameter specifies the federated domain used for delegation tokens issued by the Microsoft Federation Gateway for user accounts in the Exchange organization.

			If the *DefaultDomain* parameter isn't set, the primary SMTP domain for each user account is used in delegation tokens issued by the Microsoft Federation Gateway. Only a single domain or subdomain for the Exchange organization should be configured, and it applies to all delegation tokens issued for the Exchange organization, for example, contoso.com.
DelegationFederationTrust	Optional	Microsoft.Exchange.Configuration.Tasks.FederationTrustIdParameter	The *DelegationFederationTrust* parameter specifies the identity of the federation trust to be used by the organization identifier.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the organization identifier is enabled. Valid values include `$true` or `$false`. Setting the parameter to `$false` disables federation.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter specifies the federated organization identifier.
OrganizationContact	Optional	Microsoft.Exchange.Data.SmtpAddress	The *OrganizationContact* parameter specifies the SMTP address of the federation contact.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-FederationInformation

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-06*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-FederationInformation** cmdlet to get federation information, including federated domain names and target URLs, from an external Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-FederationInformation -DomainName <SmtpDomain> [-
BypassAdditionalDomainValidation <SwitchParameter>] [-Force
<SwitchParameter>] [-TrustedHostnames <MultiValuedProperty>]
```

Examples

EXAMPLE 1

This example gets federation information from the domain contoso.com.

```
Get-FederationInformation -DomainName contoso.com
```

Detailed Description

The **Get-FederationInformation** cmdlet retrieves federation information from the domain specified. Results from the cmdlet can be piped to the **New-OrganizationRelationship** cmdlet to establish an organization relationship with the Exchange organization being queried.

The domain specified should have federation enabled.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainName	Required	Microsoft.Exchange.Data.SmtpDomain	The *DomainName* parameter specifies the domain name for which federation information is to be retrieved.
BypassAdditionalDomainValidation	Optional	System.Management.Automation.SwitchParameter	The *BypassAdditionalDomainValidation* switch specifies that the command skip validation of domains from the external Exchange organization. We recommend that you only use this parameter when retrieving federation information in a hybrid deployment between

			on-premises and Exchange Online organizations that are part of a single, larger Exchange deployment. Don't use this parameter when retrieving federation information from external Exchange organizations in a cross-organization arrangement. The default value is `$false`.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies that the command overrides the prompt and fail immediately if the host name in the Autodiscover endpoint of the domain doesn't match the Secure Sockets Layer (SSL) certificate presented by the endpoint, and the host name isn't specified in the *TrustedHostnames* parameter.
TrustedHostnames	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *TrustedHostnames* parameter specifies the

			fully qualified domain name (FQDN) of federation endpoints. Federation endpoints are Client Access servers in an organization with federation enabled. Explicitly specifying the *TrustedHostnames* parameter allows the cmdlet to bypass prompting if the certificate presented by the endpoint doesn't match the domain name specified in the *DomainName* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-FederationTrust

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-FederationTrust** cmdlet to view the federation trust configured for the Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-FederationTrust [-Identity <FederationTrustIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves properties of the federation trust configured for the Exchange organization.

```
Get-FederationTrust | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.FederationTrustIdParameter	The *Identity* parameter specifies a federation trust ID. If not specified, the cmdlet returns all federation trusts configured for the Exchange organization.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-FederationTrust

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-FederationTrust** cmdlet to set up a federation trust between your Exchange organization and the Microsoft Federation Gateway.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-FederationTrust -ApplicationUri <String> -
SkipNamespaceProviderProvisioning <SwitchParameter> -Thumbprint <String>
[-AdministratorProvisioningId <String>] [-ApplicationIdentifier <String>]
[-MetadataUrl <Uri>] <COMMON PARAMETERS>


New-FederationTrust -Thumbprint <String> [-MetadataUrl <Uri>] [-
UseLegacyProvisioningService <SwitchParameter>] <COMMON PARAMETERS>
```

```
COMMON PARAMETERS: -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the federation trust Microsoft Federation Gateway with a certificate with the thumbprint AC00F35CBA8359953F4126E0984B5CCAFA2F4F17.

```
New-FederationTrust -Name "Microsoft Federation Gateway" -
Thumbprint AC00F35CBA8359953F4126E0984B5CCAFA2F4F17
```

Detailed Description

Federation trusts are trusts created between an Exchange organization and the Microsoft Federation Gateway. A federation trust is required to configure a federated organization identifier for federated sharing.

For more information, see Federation.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
ApplicationUri	Required	System.String	The *ApplicationUri* parameter specifies the primary domain used for the federated organization identifier. If you specify the *ApplicationUri* parameter, you must use the *SkipNamespaceProvide*

			rProvisioning switch and also specify the *AdministratorProvisioningId* and *ApplicationIdentifier* parameters.
Name	Required	System.String	The *Name* parameter specifies a friendly name for the federation trust.
SkipNamespaceProviderProvisioning	Required	System.Management.Automation.SwitchParameter	The *SkipNamespaceProviderProvisioning* switch specifies that the trust and federated organization identifier are provisioned externally without using federation functionality in Microsoft Exchange Server 2013. If you use this switch, you must specify the *ApplicationIdentifier*, *ApplicationUri*, and *AdministratorProvisioningId* parameters.
Thumbprint	Required	System.String	The *Thumbprint* parameter specifies the thumbprint of a

			certificate issued by a public certification authority (CA) trusted by the Microsoft Federation Gateway. For more details, see Federation.
AdministratorProvisioningId	Optional	System.String	The *AdministratorProvisioningId* parameter specifies the provisioning key returned by the Microsoft Federation Gateway when an organization has already registered a **SiteID** or **ApplicationID**. If you specify the *AdministratorProvisioningId* parameter, you must use the *SkipNamespaceProviderProvisioning* switch and also specify the *ApplicationIdentifier* and *ApplicationUri* parameters.
ApplicationIdentifier	Optional	System.String	The *ApplicationIdentifier* parameter specifies

			the **SiteID** or **ApplicationID** when an organization has already registered a **SiteID** or **ApplicationID**. If you specify the *ApplicationIdentifier* parameter, you must use the *SkipNamespaceProvider Provisioning* switch and also specify the *AdministratorProvisioningId* and *ApplicationUri* parameters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
MetadataUrl	Optional	System.Uri	The *MetadataUrl* parameter specifies the URL where WS-FederationMetadata is published by the Microsoft Federation Gateway.
UseLegacyProvisioningService	Optional	System.Management.Automation.SwitchParameter	The *UseLegacyProvisioningService* parameter specifies if the legacy interface on the Microsoft Federation Gateway will be used for managing the federation trust, including federated domains, certificates, and federation metadata. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When using a self-signed certificate for configuring a federation trust with the Microsoft

			Federation Gateway, the trust needs to be created with the parameter set to `$true`. After the federation trust is created, this behavior can't be changed and requires the deletion and re-creation of the federation trust. We recommend you always use the default value of `$false`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-FederationTrust

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-FederationTrust** cmdlet to remove an existing federation trust from an Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-FederationTrust -Identity <FederationTrustIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the federation trust `Microsoft Federation Gateway`.

```
Remove-FederationTrust "Microsoft Federation Gateway"
```

Detailed Description

Federation trusts are set up with Microsoft Federation Gateway to enable calendar sharing and free/busy sharing with external Exchange organizations or individuals. The **Remove-FederationTrust** cmdlet removes a federation trust.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.FederationTrustIdParameter	The *Identity* parameter specifies the identity of the federation trust being removed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can

			view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-FederationTrust

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-FederationTrust** cmdlet to modify an existing federation trust.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-FederationTrust -Identity <FederationTrustIdParameter> -
PublishFederationCertificate <SwitchParameter> <COMMON PARAMETERS>


Set-FederationTrust -Identity <FederationTrustIdParameter> [-MetadataUrl
<Uri>] [-RefreshMetadata <SwitchParameter>] [-Thumbprint <String>] <COMMON
PARAMETERS>


Set-FederationTrust -ApplicationUri <String> -Identity
<FederationTrustIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Name <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures the federation trust Microsoft Federation Gateway to use the certificate with the thumbprint AC00F35CBA8359953F4126E0984B5CCAFA2F4F17 as the next certificate.

```
Set-FederationTrust -Identity "Microsoft Federation
Gateway" -Thumbprint
AC00F35CBA8359953F4126E0984B5CCAFA2F4F17
```

EXAMPLE 2

This example configures the federation trust Microsoft Federation Gateway to use the next certificate as the current certificate.

Important:
Before you configure a federation trust to use the next certificate as the current certificate, you must use the **Test-FederationTrust** cmdlet to verify that the certificate is available on all Mailbox and Client Access servers.

```
Set-FederationTrust -Identity "Microsoft Federation
Gateway" -PublishFederationCertificate
```

Detailed Description

You can use the **Set-FederationTrust** cmdlet to manage the certificates used for the federation trust. You can also use the **Set-FederationTrust** cmdlet to refresh the metadata document from the Microsoft Federation Gateway and download its certificate.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
ApplicationUri	Required	System.String	The *ApplicationUri* parameter specifies the primary domain used for the federation

			organization identifier.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.FederationTrustIdParameter	The *Identity* parameter specifies the name of the federation trust being modified.
PublishFederationCertificate	Required	System.Management.Automation.SwitchParameter	The *PublishFederationCertificate* switch specifies the next certificate as the current certificate for the federation trust and publishes it to the Microsoft Federation Gateway. The certificate is used to encrypt tokens with the Microsoft Federation Gateway. **Caution:** Before setting the next certificate to be used as the current certificate, ensure that the certificate is deployed on all Mailbox and Client Access servers. Use the **Test-FederationCertificate** cmdlet to check the deployment status of the certificate.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
MetadataUrl	Optional	System.Uri	The *MetadataUrl* parameter specifies the URL where WS-FederationMetadata is published by the Microsoft Federation Gateway.
Name	Optional	System.String	The *Name* parameter specifies a name for the federation trust.
RefreshMetadata	Optional	System.Management.Automation.SwitchParameter	The *RefreshMetadata* switch specifies that the metadata document and certificate is retrieved again from the Microsoft Federation Gateway.
Thumbprint	Optional	System.String	The *Thumbprint*

			parameter specifies the thumbprint of the X.509 certificate to be configured as the next certificate for the federation trust. After the certificate is deployed on all Mailbox and Client Access servers in the Exchange organization, you can use the *PublishFederationCertificate* switch to configure the trust to use this certificate.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-FederationTrust

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-FederationTrust** cmdlet to verify that the federation trust is properly configured and functioning as expected.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-FederationTrust [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-MonitoringContext <$true | $false>] [-UserIdentity
<RecipientIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example validates the federation trust deployed in the Exchange organization and checks whether a security token can be retrieved from the Microsoft Federation Gateway.

```
Test-FederationTrust
```

Detailed Description

You can run the **Test-FederationTrust** cmdlet from the Exchange Management Shell, or a monitoring system can run the test periodically.

The **Test-FederationTrust** cmdlet runs the following series of tests to ensure that federation is working as expected:

- A connection to the Microsoft Federation Gateway is established. This test ensures that communication between the local Exchange server and the Microsoft Federation Gateway is working correctly.
- Certificates are checked to ensure they're valid and can be used with the Microsoft Federation

Gateway.

- A security token is requested from the Microsoft Federation Gateway. This test ensures that a token can be properly retrieved and used.

You must run the **Test-FederationTrust** cmdlet from either an Exchange Mailbox or Client Access server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
MonitoringContext	Optional	System.Boolean	The *MonitoringContext* parameter includes or excludes the associated

			monitoring events and performance counters in the results. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you specify the value `$true`, the monitoring events and performance counters are included in the command results. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
UserIdentity	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *UserIdentity* parameter specifies a mailbox user to request a token for. If a mailbox user isn't specified, the command uses the default test mailbox. If the default test mailbox isn't present,

			the test fails. You can create the default test mailbox using the `New-TestCasConnectivityUser.ps1` script found in the Scripts folder of the Exchange installation. The test mailbox only needs to be created once.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-FederationTrustCertificate

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-FederationTrustCertificate** cmdlet to check the status of certificates used for federation on all Mailbox and Client Access servers.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-FederationTrustCertificate [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example reports the status of federation certificates on all Mailbox and Client Access servers.

```
Test-FederationTrustCertificate
```

Detailed Description

The certificate used to establish a federation trust is propagated to all Mailbox and Client Access servers in the Exchange organization. The **Test-FederationTrustCertificate** cmdlet reports the status of the certificate on each Mailbox and Client Access server.

The cmdlet doesn't require any parameters.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchPara	The *Confirm* switch causes the command to

		meter	pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-HybridConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-HybridConfiguration** cmdlet to view the hybrid configuration for the Microsoft Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-HybridConfiguration [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns detailed information about the hybrid deployment configuration.

```
Get-HybridConfiguration
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Hybrid deployment configuration" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-HybridConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-28*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-HybridConfiguration** cmdlet to create the HybridConfiguration object and set up a hybrid deployment between your on-premises Exchange organization and a Microsoft Office 365 for enterprises Exchange Online organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-HybridConfiguration [-ClientAccessServers <MultiValuedProperty>] [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Domains
<MultiValuedProperty>] [-EdgeTransportServers <MultiValuedProperty>] [-
ExternalIPAddresses <MultiValuedProperty>] [-Features
<MultiValuedProperty>] [-OnPremisesSmartHost <SmtpDomain>] [-
ReceivingTransportServers <MultiValuedProperty>] [-SendingTransportServers
<MultiValuedProperty>] [-ServiceInstance <Int32>] [-TlsCertificateName
<SmtpX509Identifier>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the hybrid configuration `Hybrid Configuration` with the default hybrid configuration settings.

```
New-HybridConfiguration
```

Detailed Description

A hybrid deployment offers organizations the ability to extend the feature-rich experience and administrative control they have with their existing on-premises Microsoft Exchange organization to the cloud. The **New-HybridConfiguration** cmdlet is used with the Hybrid Configuration wizard and is typically configured when the hybrid deployment is initially created by the wizard. We strongly recommend that you use the Hybrid Configuration wizard to create the HybridConfiguration object and configure your hybrid deployment with the Exchange Online organization.

For more information, see **Exchange Server 2013 Hybrid Deployments**.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Hybrid deployment configuration" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
ClientAccessServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ClientAccessServers* parameter is deprecated and will be removed from Microsoft Exchange Server 2013.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Domains	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Domains* parameter specifies the domain namespaces that are used in the hybrid deployment. These domains must be configured as accepted domains in either the on-premises Exchange organization or the Exchange Online service. The domains are used in configuring the organization relationships and Send and Receive connectors used by the hybrid configuration.
EdgeTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *EdgeTransportServers* parameter specifies the Exchange servers with the

		y	Exchange 2010 Service Pack 2 (SP2) Edge Transport server role installed that are configured to support the hybrid deployment features. The Edge Transport server must be externally accessible from the Internet on port 25. The accepted values for the *EdgeTransportServers* parameter are either the full or short computer name of an Edge Transport server, for example, either edge.corp.contoso.com or edge. Separate server names with a comma if defining more than one Edge Transport server. When configuring the *EdgeTransportServers* parameter, you must configure the *ReceivingTransportServers* and *SendingTransportServers* parameter values to `$null`.
ExternalIPAddresses	Optional	Microsoft.Exchange.Da	The *ExternalIPAddresses*

		ta.MultiValuedProperty	parameter is a legacy parameter that specifies the publicly accessible inbound IP address of Microsoft Exchange Server 2010 Hub Transport servers. The only configuration change that should be made with this parameter is to change or clear the legacy Exchange 2010 Hub Transport server IP address value. The IP address must be Internet Protocol version 4 (IPv4) based only.
Features	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Features* parameter specifies the features that are enabled for the hybrid configuration. One or more of the following values separated by commas can be entered. When using the Hybrid Configuration wizard, all features are enabled by default. • `OnlineArchive` Enables the Exchange Online archive for on-premises Exchange and Exchange Online organization users.

			• `FreeBusy` Enables free/busy calendar information to be shared between on-premises Exchange and Exchange Online organization users. • `MailTips` Enables MailTips information to be shared between on-premises Exchange and Exchange Online organization users. • `MessageTracking` Enables message tracking information to be shared between on-premises Exchange and Exchange Online organization users. • `OWARedirection` Enables automatic Microsoft Office Outlook Web App redirection to either the on-premises Exchange or Exchange Online organizations depending on where the user mailbox is located. • `SecureMail` Enables secure message transport via Transport Layer Security (TLS) between the on-premises Exchange and Exchange Online organizations. • `Centralized` Enables the on-premises servers to handle all message transport between the on-premises Exchange and Exchange Online

			organizations, including message delivering to the Internet for both organizations. If this value is `$false`, the on-premises server and Exchange Online organization are each responsible for their own Internet message delivery. • `Photos` Enables the sharing of user photo data between the on-premises Exchange and Exchange Online organizations. This feature works in tandem with the *PhotosEnabled* parameter in the **OrganizationRelationship** cmdlets in a hybrid deployment. If the *Photos* parameter is `$true`, the *PhotosEnabled* parameter is automatically set to `$true`. If the *Photos* parameter is `$false`, the *PhotosEnabled* parameter is automatically set to `$false`. When running the Hybrid Configuration wizard for the first time, the default value is `$true`.
OnPremisesSmartHost	Optional	Microsoft.Exchange.Data.SmtpDomain	The *OnPremisesSmartHost* parameter specifies the FQDN of the on-premises

			Mailbox server used for secure mail transport for messages sent between the on-premises Exchange and Exchange Online organizations.
ReceivingTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ReceivingTransportServers* parameter specifies the Exchange servers with the Client Access server role installed that are defined in the outbound connector configuration of the Microsoft Exchange Online Protection (EOP) service included as part of the Office 365 for enterprises tenant. The servers defined in the *ReceivingTransportServers* parameter are designated as the receiving servers for secure mail messages sent from the Exchange Online organization to the on-premises Exchange organization in a hybrid deployment. At least one Client Access server must be defined and be externally accessible from the Internet for secure

			mail to be enabled between the on-premises Exchange and Exchange Online organizations. The accepted values for the *ReceivingTransportServers* parameter are either the full or short computer name of a Client Access server, for example, either CAS.corp.contoso.com or CAS. Separate server names with a comma if defining more than one Client Access server. If configuring the *EdgeTransportServers* parameter in the hybrid deployment, the *ReceivingTransportServers* parameter value must be `$null`.
SendingTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SendingTransportServers* parameter specifies the Exchange servers with the Mailbox server role installed that are defined in the inbound connector configuration of the EOP service included as part of the Office 365 for

			enterprises tenant. The servers defined in the *SendingTransportServers* parameter are designated as the receiving servers for secure mail messages sent from the on-premises Exchange organization to the Exchange Online organization in a hybrid deployment. At least one Mailbox server must be defined and be externally accessible from the Internet for secure mail to be enabled between the on-premises Exchange and Exchange Online organizations. The accepted values for the *SendingTransportServers* parameter are either the full or short computer name of a Mailbox server, for example, either MBX.corp.contoso.com or MBX. Separate server names with a comma if defining more than one Mailbox server. If configuring the *EdgeTransportServers*

			parameter in the hybrid deployment, the *SendingTransportServers* parameter value must be `$null`.
ServiceInstance	Optional	System.Int32	The *ServiceInstance* parameter should only be used by organizations manually configuring hybrid deployments with Office 365 operated by 21Vianet in China. All other organizations should use the Hybrid Configuration wizard to configure a hybrid deployment with Office 365. The valid values for this parameter are 0 (null) or 1. The default value is 0 (null).For organizations connecting with Office 365 operated by 21Vianet in China, set this value to 1 when manually configuring your hybrid deployment.
TlsCertificateName	Optional	Microsoft.Exchange.Data.SmtpX509Identifier	The *TlsCertificateName* parameter specifies the X.509 certificate to use with TLS sessions and secure mail. Valid input

			for this parameter is `[I]`*Issuer*`[S]`*Subject*. The *Issuer* value is found in the certificate's `Issuer` field, and the *Subject* value is found in the certificate's `Subject` field. You can find these values by running the **Get-ExchangeCertificate** cmdlet.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-HybridConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-HybridConfiguration** cmdlet to delete the **HybridConfiguration** Active Directory object for your on-premises Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-HybridConfiguration [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the **HybridConfiguration** object for the hybrid deployment.

```
Remove-HybridConfiguration
```

Detailed Description

Removing a **HybridConfiguration** object should typically only be performed in circumstances where the hybrid deployment state is corrupt and under the direction of Microsoft Customer Service and Support. After removing the **HybridConfiguration** object, your existing hybrid deployment configuration settings aren't disabled or removed. However, when the Hybrid Configuration wizard is run again after removing the **HybridConfiguration** object, the wizard won't have a hybrid configuration reference point for your existing feature settings. As a result, it will automatically create a **HybridConfiguration** object and record the new hybrid deployment configuration feature values defined in the wizard. The feature settings associated with the hybrid deployment, such as organization relationship or Send and Receive connector parameters, which were configured with the **HybridConfiguration** object that's removed, aren't removed or modified until the Hybrid Configuration wizard is run again.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Hybrid deployment configuration" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can

			view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-HybridConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-28*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-HybridConfiguration** cmdlet to modify the hybrid deployment between your on-premises Microsoft Exchange organization and Exchange Online in a Microsoft Office 365 for enterprises organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-HybridConfiguration [-ClientAccessServers <MultiValuedProperty>] [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Domains
<MultiValuedProperty>] [-EdgeTransportServers <MultiValuedProperty>] [-
ExternalIPAddresses <MultiValuedProperty>] [-Features
<MultiValuedProperty>] [-Name <String>] [-OnPremisesSmartHost
<SmtpDomain>] [-ReceivingTransportServers <MultiValuedProperty>] [-
SendingTransportServers <MultiValuedProperty>] [-ServiceInstance <Int32>]
[-TlsCertificateName <SmtpX509Identifier>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the secure mail and centralized transport hybrid deployment features, but keeps the Exchange Online Archive, MailTips, Outlook Web App redirection, free/busy, and message tracking features enabled between the on-premises Exchange and Exchange Online organizations.

```
Set-HybridConfiguration -Features
OnlineArchive,MailTips,OWARedirection,FreeBusy,MessageTrack
ing
```

EXAMPLE 2

This example specifies that the hybrid deployment uses a defined TLS certificate, referenced by the combination of the Issuer and Subject attributes of the CA issued X.509 certificate.

```
Set-HybridConfiguration -TlsCertificateName "<I>CN=A. Datum
Corporation CA-3, OU=www.adatum.com, O=A.Datum Corp,
C=US<S>CN=mail.contoso.com, O=Barbara Zighetti, L=Seattle,
S=WA, C=US"
```

Detailed Description

The **Set-HybridConfiguration** cmdlet modifies the hybrid configuration features, such as enabling secure mail, designating a specific Client Access server for hybrid functionality, or enabling or disabling free/busy information sharing and message tracking between the on-premises Exchange and Exchange Online organizations.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Hybrid deployment configuration" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
ClientAccessServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ClientAccessServers* parameter is deprecated and will be removed from

			Microsoft Exchange Server 2013.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Domains	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Domains* parameter specifies the domain namespaces that will be used in the hybrid deployment. These domains must be configured as accepted domains in either the on-premises Exchange organization or the Exchange Online

			organization. The domains will be used in configuring the organization relationships and Send and Receive connectors used by the hybrid configuration.
EdgeTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *EdgeTransportServers* parameter specifies the Exchange servers with the Microsoft Exchange Server 2010 Service Pack 2 (SP2) Edge Transport server role installed that are configured to support the hybrid deployment features. The Edge Transport server must be externally accessible from the Internet on port 25. The accepted values for the *EdgeTransportServers* parameter are either the full or short computer name of an Edge Transport server, for example, either edge.corp.contoso.com or edge. Separate server names with a comma if defining more than one Edge Transport server.

			When configuring the *EdgeTransportServers* parameter, you must configure the *ReceivingTransportServers* and *SendingTransportServers* parameter values to `$null`.
ExternalIPAddresses	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalIPAddresses* parameter is a legacy parameter that specifies the publicly accessible inbound IP address of Exchange 2010 Hub Transport servers. The only configuration change that should be made with this parameter is to change or clear the legacy Exchange 2010 Hub Transport server IP address value. The IP address must be Internet Protocol version 4 (IPv4) based only.
Features	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Features* parameter specifies the features enabled for the hybrid configuration. One or more of the following values separated by

			commas can be entered. When using the Hybrid Configuration wizard, all features are enabled by default. • `Centralized` Enables transport servers to handle all message transport between the on-premises Exchange and Exchange Online organizations, including external message delivery to the Internet for both organizations. If this value is `$false`, the on-premises transport servers and Exchange Online organization are each responsible for their own Internet message delivery. • `FreeBusy` Enables free/busy calendar information to be shared between on-premises Exchange and Exchange Online organization users. • `MailTips` Enables MailTips information to be shared between on-premises Exchange and Exchange Online organization users. • `MessageTracking` Enables message tracking information to be shared between on-premises Exchange and Exchange Online organization users.

			• `OnlineArchive` Enables the Exchange Online archive feature so that Exchange Online supports hosting archive mailboxes for on-premises users. • `OWARedirection` Enables automatic Microsoft Office Outlook Web App redirection to either the on-premises Exchange or Exchange Online organizations depending on where the user mailbox is located. • `SecureMail` Enables secure message transport via Transport Layer Security (TLS) between the on-premises Exchange and Exchange Online organizations. • `Photos` Enables the sharing of user photo data between the on-premises Exchange and Exchange Online organizations. This feature works in tandem with the *PhotosEnabled* parameter in the **OrganizationRelationship** cmdlets in a hybrid deployment. If the *Photos* parameter is `$true`, the *PhotosEnabled* parameter is automatically set to `$true`. If the *Photos* parameter is `$false`,

			the *PhotosEnabled* parameter is automatically set to `$false`. When running the Hybrid Configuration wizard for the first time, the default value is `$true`.
Name	Optional	System.String	The *Name* parameter isn't used and will be deprecated from the **Set-HybridConfiguration** cmdlet in a future release. There can be only one HybridConfiguration object in an Exchange organization.
OnPremisesSmartHost	Optional	Microsoft.Exchange.Data.SmtpDomain	The *OnPremisesSmartHost* parameter specifies the FQDN of the on-premises Mailbox servers used for secure mail transport for messages sent between the on-premises Exchange and Exchange Online organizations.
ReceivingTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ReceivingTransportServers* parameter specifies the Exchange servers with the Client Access server role installed that are defined in the outbound

			connector configuration of the Microsoft Exchange Online Protection (EOP) service included as part of the Office 365 for enterprises tenant. The servers defined in the *ReceivingTransportServers* parameter are designated as the receiving servers for secure mail messages sent from the Exchange Online organization to the on-premises Exchange organization in a hybrid deployment. At least one Client Access server must be defined and be externally accessible from the Internet for secure mail to be enabled between the on-premises Exchange and Exchange Online organizations. The accepted values for the *ReceivingTransportServers* parameter are either the full or short computer name of a Client Access server, for example, either CAS.corp.contoso.com or CAS. Separate server names with a comma if

			defining more than one Client Access server. If configuring the *EdgeTransportServers* parameter in the hybrid deployment, the *ReceivingTransportServers* parameter value must be `$null`.
SendingTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SendingTransportServers* parameter specifies the Exchange servers with the Mailbox server role installed that are defined in the inbound connector configuration of the EOP service included as part of the Office 365 for enterprises tenant. The servers defined in the *SendingTransportServers* parameter are designated as the receiving servers for secure mail messages sent from the on-premise organization to the Exchange Online organization in a hybrid deployment. At least one Mailbox server must be defined and be externally

			accessible from the Internet for secure mail to be enabled between the on-premises Exchange and Exchange Online organizations. The accepted values for the *SendingTransportServers* parameter are either the full or short computer name of a Mailbox server, for example, either MBX.corp.contoso.com or MBX. Separate server names with a comma if defining more than one Mailbox server. If configuring the *EdgeTransportServers* parameter in the hybrid deployment, the *SendingTransportServers* parameter value must be `$null`.
ServiceInstance	Optional	System.Int32	The *ServiceInstance* parameter should only be used by organizations manually configuring hybrid deployments with Office 365 operated by 21Vianet in China. All other organizations

			should use the Hybrid Configuration wizard to configure a hybrid deployment with Office 365. The valid values for this parameter are 0 (null) or 1. The default value is 0 (null).For organizations connecting with Office 365 operated by 21Vianet in China, set this value to 1 when manually configuring your hybrid deployment.
TlsCertificateName	Optional	Microsoft.Exchange.Data.SmtpX509Identifier	The *TlsCertificateName* parameter specifies the X.509 certificate to use with TLS sessions and secure mail. Valid input for this parameter is `[I]`*Issuer*`[s]`*Subject*. The *Issuer* value is found in the certificate's `Issuer` field, and the *Subject* value is found in the certificate's `subject` field. You can find these values by running the **Get-ExchangeCertificate** cmdlet.
WhatIf	Optional	System.Management.Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-HybridConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Update-HybridConfiguration** cmdlet to define the credentials used for updating the hybrid configuration object.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-HybridConfiguration -OnPremisesCredentials <PSCredential> -
TenantCredentials <PSCredential> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-ForceUpgrade <SwitchParameter>] [-
SuppressOAuthWarning <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example defines the credentials used when updating the hybrid configuration object and connecting to the Microsoft Office 365 for enterprises tenant.

Use this command to specify your on-premises credentials. For example, run this command and then enter *<domain>***\admin@contoso.com** and the associated account password in the credentials dialog box when prompted.

```
$OnPremisesCreds = Get-Credential
```

Use this command to specify your Office 365 for enterprises tenant credentials. For example, run this command and then enter admin@contoso.onmicrosoft.com and the associated account password in the credentials dialog box when prompted.

```
$TenantCreds = Get-Credential
```

Use this command to define the specified credentials that will be used when updating the hybrid configuration object and connecting to the Office 365 for enterprises tenant.

```
Update-HybridConfiguration -OnPremisesCredentials
$OnPremisesCreds -TenantCredentials $TenantCreds
```

Detailed Description

You can use the **Update-HybridConfiguration** cmdlet to designate the accounts and credentials that will be used when updating the hybrid configuration Active Directory object and connecting to the Microsoft Office 365 for enterprises tenant.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Hybrid deployment configuration" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
OnPremisesCredentials	Required	System.Management.Automation.PSCredential	The *OnPremisesCredentials* parameter specifies the

			on-premises Active Directory account and credentials that will be used to configure hybrid configuration features. This account must have the Organization Management management role group assigned.
TenantCredentials	Required	System.Management.Automation.PSCredential	The *TenantCredentials* parameter identifies the Office 365 for enterprises tenant organization account and credentials that will be used to configure hybrid configuration features. This is often the administrator account assigned when the Office 365 tenant was created. This account must have the Organization Management management role group assigned.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* parameter suppresses the prompt to confirm an upgrade of the HybridConfiguration Active Directory object. This confirmation prompt is only displayed when using the **Update-HybridConfiguration** cmdlet on a server

			running Microsoft Exchange Server 2013 and when the existing HybridConfiguration Active Directory object version is Exchange 2010.
SuppressOAuthWarning	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IntraOrganizationConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-02*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-IntraOrganizationConfiguration** cmdlet to view the component settings of a hybrid Exchange deployment.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-IntraOrganizationConfiguration [-Organization
<OrganizationIdParameter>] [-OrganizationGuid
<OnPremisesOrganizationIdParameter>]
```

Examples

Example 1

This example returns the settings of the intra-organization configuration.

```
Get-IntraOrganizationConfiguration
```

Detailed Description

A hybrid Exchange deployment results in one logical organization made up of a number of physical Exchange instances. Hybrid Exchange environments contain more than one Exchange instance and support topologies like two on-premises Microsoft Exchange Server 2013 forests in an organization, an Exchange 2013 on-premises organization and an Exchange Online organization, or two Exchange Online organizations.

Hybrid environments are enabled by Intra-Organization connectors. The connectors can be created and managed by cmdlets like **New-IntraOrganizationConnector**, but we strongly recommend that you use the Hybrid Configuration wizard when configuring a hybrid deployment with an Exchange Online organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Intra-Organization connectors" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationGuid	Optional	Microsoft.Exchange.Configuration.Tasks.OnPremisesOrganizationIdParameter	This parameter is available only in the cloud-based service. The *OrganizationGuid* parameter specifies the on-premises organization in a hybrid deployment that has multiple on-premises organizations defined. If you don't use the *OrganizationGuid* parameter for these types of hybrid deployments, the **Get-IntraOrganizationConfiguration** cmdlet will generate errors. To view the on-premises organization GUID values that are required for this parameter, use the **Get-OnPremisesOrganiza**

			tion cmdlet.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IntraOrganizationConnector

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-IntraOrganizationConnector** cmdlet to view the settings of Intra-Organization connectors.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-IntraOrganizationConnector [-Identity
<IntraOrganizationConnectorIdParameter>] [-DomainController <Fqdn>] [-
Organization <OrganizationIdParameter>]
```

Examples

Example 1

This example returns a summary list of all Intra-Organization connectors.

```
Get-IntraOrganizationConnector
```

Example 2

This example returns details about the Intra-Organization connector named "MainCloudConnector".

```
Get-IntraOrganizationConnector "MainCloudConnector" |
Format-List
```

Detailed Description

Intra-Organizational connectors enable features and services between divisions in your Exchange organization. It allows for the expansion of organizational boundaries for features and services across different hosts and network boundaries, such as between Active Directory forests, between on-premises and cloud-based organizations, or between tenants hosted in the same or different datacenters.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Intra-Organization connectors" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.IntraOrganizationConnectorIdParameter	The *Identity* parameter specifies the Intra-Organization connector that you want to view. You can use any value that uniquely identifies the connector. For example:

			• Name • Distinguished name (DN) • GUID
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-IntraOrganizationConnector

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-IntraOrganizationConnector** cmdlet to create an Intra-Organization connector between two on-premises Microsoft Exchange Server 2013 forests in an organization, between an Exchange 2013 on-premises organization and an Exchange Online organization, or between two Exchange Online organizations. This connector enables feature availability and service connectivity across the organizations using a common connector and connection endpoints.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-IntraOrganizationConnector -DiscoveryEndpoint <Uri> -Name <String> -
TargetAddressDomains <MultiValuedProperty> [-Confirm [<SwitchParameter>]]
[-DomainController <Fqdn>] [-Enabled <$true | $false>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example creates an Intra-Organization connector named "MainCloudConnector" between an on-premises Exchange 2013 organization and an Exchange Online organization containing two domains, Cloud1.contoso.com and Cloud2.contoso.com.

```
New-IntraorganizationConnector -DiscoveryEndpoint http://
ExternalDiscovery.Contoso.com -Name MainCloudConnector -
TargetAddressDomains Cloud1.contoso.com,Cloud2.contoso.com
```

Detailed Description

The **New-IntraOrganizationConnector** cmdlet is used to create a connection for features and services between divisions in your Exchange organization. It allows for the expansion of organizational boundaries for features and services across different hosts and network boundaries, such as between Active Directory forests, between on-premises and cloud-based organizations, or between tenants hosted in the same or different datacenters.

For hybrid deployments between on-premises Exchange 2013 and Exchange Online organizations, the **New-IntraOrganizationConnector** cmdlet is used by the Hybrid Configuration wizard. Typically, the Intra-Organization connector is configured when the hybrid deployment is initially created by the wizard. We strongly recommend that you use the Hybrid Configuration wizard to create the Intra-Organization connector when configuring a hybrid deployment with an Exchange Online organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Intra-Organization connectors" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DiscoveryEndpoint	Required	System.Uri	The *DiscoveryEndpoint* parameter specifies the externally-accessible URL that's used for the Autodiscover service

			for the domain that's configured in the Intra-Organization connector.
Name	Required	System.String	The *Name* parameter specifies a friendly name for the Intra-Organization connector. If the value contains spaces, enclose the value in double quotation marks.
TargetAddressDomains	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *TargetAddressDomains* parameter specifies the domain namespaces that will be used in the Intra-organization connector. These domains must have valid Autodiscover endpoints defined in their organizations. The domains and their associated Autodiscover endpoints are used by the Intra-Organization connector for feature and service connectivity. You specify multiple

			domain values separated by commas.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter enables or disabled the Intra-organization connector. The valid values for this parameter are `$true` or

			`$false`. The default value is `$true`. When you set the value to `$false`, you completely stop connectivity for the specific connection.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-IntraOrganizationConnector

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-IntraOrganizationConnector** cmdlet to remove existing Intra-Organization connectors.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-IntraOrganizationConnector -Identity
<IntraOrganizationConnectorIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example removes the existing Intra-Organization connector named "Contoso On-premises-Exchange Online".

```
Remove-IntraOrganizationConnector "Contoso On-premises-Exchange Online"
```

Detailed Description

Intra-Organizational connectors enable features and services between divisions in your Exchange organization. It allows for the expansion of organizational boundaries for features and services across different hosts and network boundaries, such as between Active Directory forests, between on-premises and cloud-based organizations, or between tenants hosted in the same or different datacenters.

The **Remove-IntraOrganizationConnector** cmdlet removes the connector objects. To stop feature or service connectivity without removing the connector object, run the command `Set-IntraOrganizationConnector <ConnectorIdentity> -Enabled $false`.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Intra-Organization connectors" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.IntraOrganizationConnectorIdParameter	The *Identity* parameter specifies the Intra-Organization connector that you want to remove. You can use any value that uniquely identifies the connector. For example: • Name • Distinguished name (DN) • GUID
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-IntraOrganizationConnector

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-IntraOrganizationConnector** cmdlet to modify an existing Intra-Organization connector between two on-premises Microsoft Exchange Server 2013 forests in an organization, or between an Exchange 2013 on-premises organization and an Exchange Online organization, or between two Exchange Online organizations.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-IntraOrganizationConnector -Identity
<IntraOrganizationConnectorIdParameter> [-Confirm [<SwitchParameter>]] [-
DiscoveryEndpoint <Uri>] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-Name <String>] [-TargetAddressDomains <MultiValuedProperty>] [-
WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example disables the Intra-Organization connector named "MainCloudConnector".

```
Set-IntraOrganizationConnector "MainCloudConnector" -
Enabled $false
```

Detailed Description

Intra-Organizational connectors enable features and services between divisions in your Exchange organization. It allows for the expansion of organizational boundaries for features and services across different hosts and network boundaries, such as between Active Directory forests, between on-premises and cloud-based organizations, or between tenants hosted in the same or different datacenters.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Intra-Organization connectors" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter specifies the Intra-

		nfiguration.Tasks.IntraOrganizationConnectorIdParameter	Organization connector that you want to modify. You can use any value that uniquely identifies the connector. For example: • Name • Distinguished name (DN) • GUID
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DiscoveryEndpoint	Optional	System.Uri	The *DiscoveryEndpoint* parameter specifies the externally accessible URL used for the Autodiscover service for the domain configured in the IntraOrganization Connector. This parameter is automatically populated with the *TargetAutodiscoverEpr* value from the **Get-**

			FederationInformation cmdlet for the defined domain.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter enables or disabled the Intra-organization connector. The valid values for this parameter are `$true` or `$false`. The default value is `$true`. When you set the value to `$false`, you completely stop connectivity for the specific connection.
Name	Optional	System.String	The *Name* parameter specifies a friendly name for the Intra-Organization connector. If the value contains spaces, enclose the value in double

			quotation marks.
TargetAddressDomains	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *TargetAddressDomains* parameter specifies the domain namespaces that will be used in the Intra-Organization connector. The domains must have valid Autodiscover endpoints defined in their organizations. The domains and their associated Autodiscover endpoints are used by the Intra-Organization connector for feature and service connectivity.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PendingFederatedDomain

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-PendingFederatedDomain** cmdlet to display a list of pending federated domains for the federation trust for your Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PendingFederatedDomain
```

Examples

EXAMPLE 1

This example retrieves the pending federated domain information for the federation trust for your Exchange organization.

```
Get-PendingFederatedDomain
```

Detailed Description

The **Get-PendingFederatedDomain** cmdlet is used as part of the Exchange Administration Center (EAC) functionality and shouldn't be used by administrators manually configuring a federation trust.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-PendingFederatedDomain

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-PendingFederatedDomain** cmdlet to configure pending domains with the federated organization identifier in the federation trust for the Exchange organization.

Important:
The domains being added to the federation trust must exist as accepted domains in the Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-PendingFederatedDomain [-Identity <OrganizationIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-PendingAccountNamespace
<SmtpDomain>] [-PendingDomains <SmtpDomain[]>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds the pending domains contoso.com and sales.contoso.com to the existing federation trust.

```
Set-PendingFederatedDomain -PendingDomains
contoso.com,sales.contoso.com
```

Detailed Description

The **Set-PendingFederatedDomain** cmdlet is used as part of the Exchange Administration Center (EAC) functionality and shouldn't be used by administrators manually configuring a federation trust. This cmdlet allows the EAC to save domains selected as the *FederatedOrganizationIdentifier* or federated domains when proof of domain ownership hasn't been completed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Federation trusts" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.OrganizationIdParameter	is reserved for internal Microsoft use.
PendingAccountNamespace	Optional	Microsoft.Exchange.Data.SmtpDomain	The *PendingAccountNamespace* parameter specifies the pending domain used as the account namespace for the federation trust.
PendingDomains	Optional	Microsoft.Exchange.Data.SmtpDomain[]	The *PendingDomains* parameter specifies the pending federated domains configured for the federation trust.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-Recipient

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Update-Recipient** cmdlet to add Microsoft Exchange attributes to recipient objects created by the global address list (GAL) synchronization management agent in Microsoft Forefront Identity Manager (FIM) 2010. The recipient objects you modify using this cmdlet must reside on a server running Microsoft Exchange Server 2010 or later.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-Recipient -Identity <RecipientIdParameter> [-Confirm
[<SwitchParameter>]] [-Credential <PSCredential>] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds Exchange attributes to the mail contact that represents John Smith's mailbox.

```
Update-Recipient -Identity "John Smith"
```

EXAMPLE 2

This example updates all contacts in a specific organizational unit (OU). This example assumes that recipients are synchronized between two forests, contoso.com and fabrikam.com, and all the synchronized recipients from the fabrikam.com domain are stored in a specific OU called fabrikam.com Users in the contoso.com domain.

```
Get-MailContact -OrganizationalUnit "contoso.com/
fabrikam.com Users" | Update-Recipient
```

Detailed Description

Because of mergers, acquisitions, or legal requirements, customers may need to deploy Exchange in a multiple Exchange forest topology. These deployments require the synchronization of recipient objects across disparate Active Directory forests.

Microsoft provides the GAL synchronization management agent for synchronizing recipient objects. The version of the GAL synchronization management agent included in Microsoft Identity Integration Server (MIIS) 2003 was designed to work with Exchange Server 2003 and relied on the Recipient Update Service. Because the Recipient Update Service is a deprecated feature and is no longer required, the new GAL synchronization management agent included in FIM 2010 is designed to function without the Recipient Update Service.

As part of the synchronization process, the FIM 2010 GAL synchronization management agent creates recipient objects in both Active Directory forests. After the recipients are created, the management agent uses the **Update-Recipient** cmdlet to add the attributes required by Microsoft Exchange to complete the provisioning of these recipients.

In Exchange, before you can run the **Update-Recipient** cmdlet to convert an Active Directory user object into an Exchange mailbox, you must stamp the user object with the following three mandatory Exchange attributes:

- **homeMDB**
- **mailNickname**
- **msExchHomeServerName**

Note:
If you're using MIIS 2003, you must run various cmdlets to complete the provisioning process of the mail contacts created by the GAL synchronization management agent. The **Update-Recipient** cmdlet provides an alternate and more efficient method to do this. You can run the **Update-Recipient** cmdlet against the recipient objects created by the MIIS 2003 GAL synchronization management agent to complete the provisioning process.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *Identity* parameter specifies the recipient. This parameter accepts

			the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-RemoteMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-RemoteMailbox** cmdlet to remove a mailbox from the cloud-based service. The associated user object in the on-premises Active Directory isn't removed.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-RemoteMailbox <COMMON PARAMETERS>

Disable-RemoteMailbox [-Archive <SwitchParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: -Identity <RemoteMailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreDefaultScope
<SwitchParameter>] [-IgnoreLegalHold <SwitchParameter>] [-
PreventRecordingPreviousDatabase <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the mailbox in the service associated with the on-premises mail-enabled user Kim Akers. The mail-enabled user is automatically converted to a regular user. This example assumes directory synchronization has been configured.

```
Disable-RemoteMailbox "Kim Akers"
```

EXAMPLE 2

This example removes an archive mailbox in the service but keeps the mailbox in the service associated with the on-premises mail-enabled user David Strome. This example assumes directory synchronization has been configured.

```
Disable-RemoteMailbox "David Strome" -Archive
```

Detailed Description

Use the **Disable-RemoteMailbox** cmdlet if you want to remove the mailbox from the service but keep the associated on-premises mail-enabled user. You can also use it to disconnect the archive mailbox in the service from the mailbox in the service. If you want to remove both the mailbox from the service and the on-premises mail-enabled user, use the Remove-RemoteMailbox cmdlet.

Directory synchronization must be configured correctly for a mailbox to be removed from the service. Removal of the mailbox from the service isn't immediate and depends on the directory synchronization schedule.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remote mailboxes" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RemoteMailboxIdParameter	The *Identity* parameter takes one of the following values: • ADObjectID • GUID

			• Distinguished name (DN) • *Domain**SamAccountName* • User principal name (UPN) • LegacyExchangeDN • Email address • User alias
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* switch specifies whether to disconnect the archive mailbox in the service from the associated mailbox in the service. The on-premises mail-enabled user and its associated mailbox in the service aren't removed if you use this switch.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as

			alias or GUID, aren't accepted.
IgnoreLegalHold	Optional	System.Management.Automation.SwitchParameter	The *IgnoreLegalHold* switch ignores the legal hold status of the mailbox user and allows you to disable the cloud-based mailbox that's on legal hold. **Caution:** When you disable a mailbox, the mailbox is disconnected from the user account. After you disable a mailbox, you can't include it in a discovery search. Disconnected mailboxes are permanently deleted from the mailbox database after the deleted mailbox retention period expires. Check with your organization's legal or Human Resources department before disabling a mailbox that's on legal hold.
PreventRecordingPreviousDatabase	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what

			changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-RemoteMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-RemoteMailbox** cmdlet to create a mailbox in the cloud-based service for an existing user in the on-premises Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-RemoteMailbox [-RemoteRoutingAddress <ProxyAddress>] <COMMON
PARAMETERS>

Enable-RemoteMailbox -Room <SwitchParameter> [-RemoteRoutingAddress
<ProxyAddress>] <COMMON PARAMETERS>

Enable-RemoteMailbox -Equipment <SwitchParameter> [-RemoteRoutingAddress
<ProxyAddress>] <COMMON PARAMETERS>

Enable-RemoteMailbox [-RemoteRoutingAddress <ProxyAddress>] <COMMON
PARAMETERS>

Enable-RemoteMailbox [-Archive <SwitchParameter>] [-ArchiveName
```

```
<MultiValuedProperty>] <COMMON PARAMETERS>

COMMON PARAMETERS: -Identity <UserIdParameter> [-Alias <String>] [-Confirm
[<SwitchParameter>]] [-DisplayName <String>] [-DomainController <Fqdn>] [-
PrimarySmtpAddress <SmtpAddress>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example mail-enables an existing on-premises user and creates an associated mailbox in the service. The remote routing address doesn't need to be specified because mail flow between the on-premises organization and the service has been configured. Using this configuration, the **Enable-RemoteMailbox** cmdlet automatically calculates the SMTP address of the mailbox to be used with the *RemoteRoutingAddress* parameter. This example also assumes directory synchronization has been configured.

Note:

To mail-enable an existing user and create an associated mailbox in the service, run the New-RemoteMailbox cmdlet and specify the identity of the existing user.

```
Enable-RemoteMailbox "Kim Akers"
```

After the user is mail-enabled, directory synchronization synchronizes the mail-enabled user to the service and the associated mailbox is created.

EXAMPLE 2

This example does the following:

- Mail-enables an existing on-premises user.
- Creates the associated mailbox in the service.
- Creates an archive mailbox in the service for the mailbox.

As in Example 1, this example assumes that mail flow and directory synchronization have been properly configured.

To mail-enable an on-premises user, create the associated mailbox in the service, enable the archive mailbox in the service, and include the *Archive* switch with the **Enable-RemoteMailbox** cmdlet.

```
Enable-RemoteMailbox "Kim Akers" -Archive
```

Detailed Description

The **Enable-RemoteMailbox** cmdlet mail-enables an existing on-premises user. The mail-enabled user contains a specific attribute that indicates that an associated mailbox in the service should be

created when the user is synchronized to the service using directory synchronization.

Directory synchronization must be configured correctly for a mailbox to be created in the service. Creation of the mailbox in the service isn't immediate and depends on the directory synchronization schedule.

Important:
The policies that you apply to recipients in the on-premises Exchange organization, such as Unified Messaging or compliance policies, aren't applied to mailboxes in the service. You must configure policies in the service if you want policies to be applied to recipients in the service.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remote mailboxes" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Equipment	Required	System.Management.Automation.SwitchParameter	The *Equipment* switch specifies that the mailbox in the service should be created as an equipment resource mailbox. You can't use the *Equipment* switch if you specified the *Room* switch.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	The *Identity* parameter specifies the identity of the existing on-premises user. The *Identity* parameter can have one of the following values: • ADObjectID • GUID • Distinguished name (DN) • *Domain*

			SamAccountName • User principal name (UPN) • LegacyExchangeDN • User alias
Room	Required	System.Management.Automation.SwitchParameter	The *Room* switch specifies that the mailbox in the service should be created as a room resource mailbox. You can't use the *Room* switch if you specified the *Equipment* switch.
Alias	Optional	System.String	The *Alias* parameter specifies the alias of the user and its associated mailbox in the service. An alias can contain letters, numbers, and the following punctuation marks and symbols: • ! • # • $ • % • ^ • & • * • + • - • . • /

			• = • ? • _ • { • } • | • ~
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* switch specifies whether to create an archive mailbox in the service in addition to the mailbox created in the service. You don't have to specify a value with this switch.
ArchiveName	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ArchiveName* parameter specifies the name of the archive mailbox. Use this parameter to change the name of the archive.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DisplayName	Optional	System.String	The *DisplayName* parameter specifies the display name for the mailbox that's created in the service.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the primary SMTP address for the mail user. By default, the primary SMTP address is generated based on the default email address policy. If you specify a primary SMTP address by using this parameter, the cmdlet sets the **EmailAddressPolicyEnabled** attribute of the mail user to `$false`, and the email addresses of this mail user aren't automatically updated based on email address policies.

RemoteRoutingAddress	Optional	Microsoft.Exchange.Data.ProxyAddress	The *RemoteRoutingAddress* parameter specifies the SMTP address of the mailbox in the service that this user is associated with. If you've configured mail flow between the on-premises organization and the service, you don't need to specify this parameter. The remote routing address is calculated automatically.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-RemoteMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-RemoteMailbox** cmdlet to retrieve the mail-related attributes of one or more users in the on-premises Active Directory associated with mailboxes in the cloud-based service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-RemoteMailbox [-Anr <String>] <COMMON PARAMETERS>

Get-RemoteMailbox [-Identity <RemoteMailboxIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-Archive <SwitchParameter>] [-Credential
<PSCredential>] [-DomainController <Fqdn>] [-Filter <String>] [-
IgnoreDefaultScope <SwitchParameter>] [-OnPremisesOrganizationalUnit
<OrganizationalUnitIdParameter>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example retrieves a complete list of remote mailboxes for the entire Exchange organization.

```
Get-RemoteMailbox
```

EXAMPLE 2

This example uses alternate credentials to retrieve a list of one or more mail-enabled users with mailboxes in the service. This is useful if the account you typically use doesn't have administrative permissions. The credentials are used to access the on-premises Active Directory domain controllers.

First, run the following command to prompt you for your credentials, and then store them in a

variable.

```
$Credentials = Get-Credential
```

Then retrieve a list of remote mailboxes using the credentials you provided by using the following command.

```
Get-RemoteMailbox -Credential $Credentials
```

Detailed Description

The **Get-RemoteMailbox** cmdlet retrieves the mail-related attributes of a mail user in the on-premises Active Directory. It doesn't retrieve the attributes of the associated mailbox in the service. Most of the mail-related attributes of the on-premises mail user and the associated mailbox in the service should be the same. However, the mailbox in the service has additional attributes that you can't view by using the **Get-RemoteMailbox** cmdlet. To view the attributes of the mailbox, you must instead either view the mailbox by opening the appropriate forest in the Exchange Administration Center or connect to the service using the Exchange Management Shell and use the **Get-Mailbox** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remote mailboxes" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)**

			• **DisplayName** • **FirstName** • **LastName** • **Alias**
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* switch specifies whether to return information about the recipient's archive mailbox.
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password to use to access the on-premises Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter

			used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.RemoteMailboxIdParameter	The *Identity* parameter identifies the remote mailbox. You can use one of the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the

			IgnoreDefaultScope parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias, aren't accepted. • You can't use the *OnPremisesOrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
OnPremisesOrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OnPremisesOrganizationalUnit* parameter specifies a container in the on-premises organization in which to limit the results. You can specify either an organizational unit (OU) or a domain. The canonical name should be specified, for example: • OU:

			westcoast.contoso.com/users • Domain: westcoast.contoso.com
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest and don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Exchange.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all mailboxes that match the query, use `unlimited` for the value of

			this parameter. The default value is 1000.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. You can sort by only one attribute at a time. You can sort by the following attributes: • Alias • Display name • Name The results are sorted in ascending order.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-RemoteMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-08*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-RemoteMailbox** cmdlet to create a mail-enabled user in the on-premises Active Directory and also create an associated mailbox in the cloud-based service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-RemoteMailbox -Password <SecureString> -UserPrincipalName <String>
<COMMON PARAMETERS>


New-RemoteMailbox -Room <SwitchParameter> [-Password <SecureString>] [-
UserPrincipalName <String>] <COMMON PARAMETERS>


New-RemoteMailbox -Equipment <SwitchParameter> [-Password <SecureString>]
[-UserPrincipalName <String>] <COMMON PARAMETERS>


New-RemoteMailbox -AccountDisabled <SwitchParameter> [-Password
<SecureString>] [-UserPrincipalName <String>] <COMMON PARAMETERS>


New-RemoteMailbox [-Password <SecureString>] [-UserPrincipalName <String>]
<COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-Alias <String>] [-ArbitrationMailbox
<MailboxIdParameter>] [-Archive <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DisplayName <String>] [-DomainController <Fqdn>] [-
FirstName <String>] [-ImmutableId <String>] [-Initials <String>] [-
Languages <MultiValuedProperty>] [-LastName <String>] [-
MailboxProvisioningConstraint <MailboxProvisioningConstraint>] [-
MailboxProvisioningPreferences <MultiValuedProperty>] [-ModeratedBy
<MultiValuedProperty>] [-ModerationEnabled <$true | $false>] [-
OnPremisesOrganizationalUnit <OrganizationalUnitIdParameter>] [-
OverrideRecipientQuotas <SwitchParameter>] [-PrimarySmtpAddress
<SmtpAddress>] [-RemotePowerShellEnabled <$true | $false>] [-
RemoteRoutingAddress <ProxyAddress>] [-ResetPasswordOnNextLogon <$true |
$false>] [-SamAccountName <String>] [-SendModerationNotifications <Never |
Internal | Always>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates an on-premises mail-enabled user and its associated mailbox in the service. The remote routing address doesn't need to be specified because mail flow between the on-premises organization and the service has been configured. Using this configuration, the **New-RemoteMailbox** cmdlet automatically calculates the SMTP address of the mailbox to be used with the *RemoteRoutingAddress* parameter. This example also assumes directory synchronization has been configured.

First, store the password to use with the new remote mailbox in a variable by using the **Get-Credential** cmdlet.

```
$Credentials = Get-Credential
```

Then run the **New-RemoteMailbox** cmdlet to create the mail user.

```
New-RemoteMailbox -Name "Kim Akers" -Password
$Credentials.Password -UserPrincipalName
kim@corp.contoso.com
```

After the new mail user is created, directory synchronization synchronizes the new mail user to the service and the associated mailbox is created.

EXAMPLE 2

This example shows how to do the following:

- Creates an on-premises mail-enabled user. The mail-enabled user is placed in the contoso.com/Archive Users OU. The OU has no effect on the mailbox in the service.
- Creates the associated mailbox in the service.
- Creates an archive mailbox in the service for the mailbox.

As in Example 1, this example assumes that mail flow and directory synchronization have been properly configured.

First, store the password to use with the new remote mailbox in a variable by using the **Get-Credential** cmdlet.

```
$Credentials = Get-Credential
```

Then run the **New-RemoteMailbox** cmdlet to create the mail user.

```
New-RemoteMailbox -Name "Kim Akers" -Password
$Credentials.Password -UserPrincipalName
kim@corp.contoso.com -OnPremisesOrganizationalUnit
"corp.contoso.com/Archive Users" -Archive
```

Detailed Description

The **New-RemoteMailbox** cmdlet creates an on-premises mail-enabled user. The mail-enabled user contains a specific attribute, which indicates that an associated mailbox in the service should be created when the user is synchronized to the service using directory synchronization.

Directory synchronization must be configured correctly for a mailbox to be created in the service. Creation of the mailbox in the service isn't immediate and depends on the directory synchronization schedule.

Important:
The policies that you apply to recipients in the on-premises Exchange organization, such as Unified Messaging or compliance policies, aren't applied to mailboxes in the service. You must configure policies in the service if you want policies to be applied to recipients in the service.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remote mailboxes" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountDisabled	Required	System.Management.Automation.SwitchParameter	The *AccountDisabled* switch specifies whether to create the mail-enabled user in a disabled state. You don't have to specify a value with this parameter.
Equipment	Required	System.Management.Automation.SwitchParameter	The *Equipment* switch specifies that the mailbox in the service should be created as an equipment resource mailbox. You can't use the *Equipment* switch if you specified the *Room* switch.
Name	Required	System.String	The *Name* parameter specifies the common name (CN) of the on-premises mail-enabled user and its associated mailbox in the service.
Password	Required	System.Security.SecureString	The *Password* parameter specifies the password used by the mail user to secure his or her account and associated mailbox in the service.
Room	Required	System.Management.	The *Room* switch specifies

		Automation.SwitchParameter	that the mailbox in the service should be created as a room resource mailbox. You can't use the *Room* switch if you specified the *Equipment* switch.
UserPrincipalName	Required	System.String	The *UserPrincipalName* parameter specifies the logon name for the user. This is the name the user will use for authentication. The UPN can be different than the user's email address. For example. a user could have a UPN of user@corp.contoso.com, but their email address could be user@contoso.com.
Alias	Optional	System.String	The *Alias* parameter specifies the email alias of the user and its associated mailbox in the service that you're creating. The alias can be a combination of characters separated by a period with no intervening spaces. Don't use special characters in the alias.

ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *ArbitrationMailbox* parameter specifies the mailbox used to manage the moderation process.
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* switch specifies whether to create an archive mailbox in the service in addition to the mailbox that's created in the service. You don't have to specify a value with this switch.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the name displayed in Microsoft Outlook for the mail user and its associated mailbox in the service.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
FirstName	Optional	System.String	The *FirstName* parameter specifies the first name of the user that you create.
ImmutableId	Optional	System.String	The *ImmutableId* parameter is used by GAL Synchronization (GALSync) and specifies a unique and immutable identifier in the form of an SMTP address for an Exchange mailbox that's used for federated delegation when requesting Security Assertion Markup Language (SAML) tokens. If federation is configured for this mailbox and you don't set this parameter when you create the mailbox, Exchange creates the value for the immutable identifier based upon the mailbox's **ExchangeGUID** and the federated account

			namespace, for example, 7a78e7c8-620e-4d85-99d3-c90d90f29699@mail.contoso.com. You must set the *ImmutableId* parameter if Active Directory Federation Services (AD FS) is deployed to allow single sign-on into an off-premises mailbox and AD FS is configured to use a different attribute than **ExchangeGUID** for sign-on token requests. Both, Exchange and AD FS must request the same token for the same user to ensure proper functionality for a cross-premise Exchange deployment scenario.
Initials	Optional	System.String	The *Initials* parameter specifies the initials of the user that you create.
Languages	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
LastName	Optional	System.String	The *LastName* parameter specifies the last name of

			the user that you create.
MailboxProvisioningConstraint	Optional	Microsoft.Exchange.Data.Directory.MailboxProvisioningConstraint	This parameter is reserved for internal Microsoft use.
MailboxProvisioningPreferences	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users responsible for moderating the messages sent to this mail user and its associated mailbox in the service. To designate more than one user, separate the users with commas. This parameter is required if you set the *ModerationEnabled* parameter to `$true`.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable or disable moderation for the mail user. The two possible values for this parameter are `$true` or `$false`. To enable moderation, set this parameter to `$true`. To

			disable moderation, set this parameter to `$false`. The default value is `$false`.
OnPremisesOrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OnPremisesOrganizationalUnit* parameter specifies the organizational unit (OU) in the on-premises organization in which the new mailbox is added (for example, redmond.contoso.com/Users). This parameter has no effect on the mailbox in the service.
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the primary SMTP address for the mail user. By default, the primary SMTP address is generated based on the default email address policy. If you specify a primary SMTP address by using this parameter, the command sets the

			EmailAddressPolicyEnabled attribute of the mail user to `$false`, and the email addresses of this mail user aren't automatically updated based on email address policies.
RemotePowerShellEnabled	Optional	System.Boolean	The *RemotePowerShellEnabled* parameter specifies whether the user can use remote Windows PowerShell. Remote Windows PowerShell is required to open the Exchange Management Shell on Mailbox and Client Access servers. Access to remote Windows PowerShell is required even if you're trying to open the Shell on the local server. The valid values are `$True` and `$False`. The default value is `$True`.
RemoteRoutingAddress	Optional	Microsoft.Exchange.Data.ProxyAddress	The *RemoteRoutingAddress* parameter specifies the SMTP address of the mailbox in the service that

			this user is associated with. This address is created automatically when the service is initially configured in the format of <your domain>.mail.onmicrosoft.com. If you've configured mail flow between the on-premises organization and the service, such as in a hybrid deployment, you don't need to specify this parameter. The remote routing address is calculated automatically and assigned to the email address policy for the on-premises organization by the Hybrid Configuration wizard.
ResetPasswordOnNextLogon	Optional	System.Boolean	The *ResetPasswordOnNextLogon* parameter specifies whether the password in the *Password* parameter must be reset the next time the user logs on. If set to `$true`, the *ResetPasswordOnNextLogon* parameter specifies

			that the password in the *Password* parameter must be reset the next time the user logs on.
SamAccountName	Optional	System.String	The *SamAccountName* parameter defines the logon name used to support clients and servers running older versions of the operating system. This attribute must contain fewer than 20 characters. An account name can contain letters, numbers, and the following punctuation marks and symbols: • ! • # • $ • % • ^ • & • - • . • _ • { • } • | • ~
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.	The *SendModerationNotificati*

		TransportModerationNotificationFlags	*ons* parameter specifies whether status notifications are sent to users when they send a message to the moderated distribution group. You can specify one of the following values: • `Always` • `Internal` • `Never` Set this parameter to `Always` if you want notifications to be sent to all senders. Set this parameter to `Internal` if you want notifications to be sent only to the senders who are internal to your organization. Set this parameter to `Never` to disable all status notifications. The default value is `Never`. **Note:** The sender is always notified if the message is rejected by the moderators, regardless of the value of this parameter.
WhatIf	Optional	System.Management.	The *WhatIf* switch

		Automation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-RemoteMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-RemoteMailbox** cmdlet to remove a mail-enabled user in the on-premises Active Directory and the associated mailbox in the cloud-based service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-RemoteMailbox -Identity <RemoteMailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreDefaultScope
<SwitchParameter>] [-IgnoreLegalHold <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the on-premises mail-enabled user Kim Akers and the associated mailbox from the service. This example assumes directory synchronization has been configured.

```
Remove-RemoteMailbox "Kim Akers"
```

Detailed Description

With the **Remove-RemoteMailbox** cmdlet, you can remove an on-premises mail-enabled user and the mailbox from the service. If you only want to remove the mailbox from the service and keep the associated on-premises user, use the Disable-RemoteMailbox cmdlet.

Directory synchronization must be configured correctly for a mailbox to be removed from the service. Removal of the mailbox from the service isn't immediate and depends on the directory synchronization schedule.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remote mailboxes" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RemoteMailboxIdParameter	The *Identity* parameter identifies the mail-enabled user and the associated mailbox in the service that you want to remove. You can use one of the following values: • ADObjectID • Distinguished name (DN) • Legacy DN • GUID

			• *Domain\Account name* • User principal name (UPN) • Email address • Alias
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to

			access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
IgnoreLegalHold	Optional	System.Management.Automation.SwitchParameter	The *IgnoreLegalHold* switch ignores the legal hold status of the mailbox user and allows you to remove the cloud-based mailbox on legal hold. **⚠Warning:** After you remove a mailbox, you can't include it in a discovery search. Depending on the command parameters you use, removed mailboxes are either purged immediately or

			when the deleted mailbox retention period expires. Check with your organization's legal or Human Resources department before disabling a mailbox that's on legal hold.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-RemoteMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Federation and hybrid cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-RemoteMailbox** cmdlet to modify the mail-related attributes of an existing user in Active Directory that's associated with a mailbox in the cloud-based service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-RemoteMailbox -Identity <RemoteMailboxIdParameter> [-
AcceptMessagesOnlyFrom <MultiValuedProperty>] [-
AcceptMessagesOnlyFromDLMembers <MultiValuedProperty>] [-
AcceptMessagesOnlyFromSendersOrMembers <MultiValuedProperty>] [-
AggregatedMailboxGuids <MultiValuedProperty>] [-Alias <String>] [-
ArbitrationMailbox <MailboxIdParameter>] [-ArchiveGuid <Guid>] [-
ArchiveName <MultiValuedProperty>] [-ArchiveQuota <Unlimited>] [-
ArchiveWarningQuota <Unlimited>] [-BypassModerationFromSendersOrMembers
<MultiValuedProperty>] [-CalendarVersionStoreDisabled <$true | $false>] [-
Confirm [<SwitchParameter>]] [-CreateDTMFMap <$true | $false>] [-
CustomAttribute1 <String>] [-CustomAttribute10 <String>] [-
CustomAttribute11 <String>] [-CustomAttribute12 <String>] [-
CustomAttribute13 <String>] [-CustomAttribute14 <String>] [-
CustomAttribute15 <String>] [-CustomAttribute2 <String>] [-
CustomAttribute3 <String>] [-CustomAttribute4 <String>] [-CustomAttribute5
<String>] [-CustomAttribute6 <String>] [-CustomAttribute7 <String>] [-
CustomAttribute8 <String>] [-CustomAttribute9 <String>] [-DisplayName
<String>] [-DomainController <Fqdn>] [-EmailAddresses
<ProxyAddressCollection>] [-EmailAddressPolicyEnabled <$true | $false>] [-
EndDateForRetentionHold <DateTime>] [-ExchangeGuid <Guid>] [-
ExtensionCustomAttribute1 <MultiValuedProperty>] [-
ExtensionCustomAttribute2 <MultiValuedProperty>] [-
ExtensionCustomAttribute3 <MultiValuedProperty>] [-
ExtensionCustomAttribute4 <MultiValuedProperty>] [-
ExtensionCustomAttribute5 <MultiValuedProperty>] [-GrantSendOnBehalfTo
<MultiValuedProperty>] [-HiddenFromAddressListsEnabled <$true | $false>]
[-IgnoreDefaultScope <SwitchParameter>] [-ImmutableId <String>] [-
JournalArchiveAddress <SmtpAddress>] [-LitigationHoldDate <DateTime>] [-
LitigationHoldEnabled <$true | $false>] [-LitigationHoldOwner <String>] [-
MailboxContainerGuid <Guid>] [-MailTip <String>] [-MailTipTranslations
<MultiValuedProperty>] [-MaxReceiveSize <Unlimited>] [-MaxSendSize
<Unlimited>] [-ModeratedBy <MultiValuedProperty>] [-ModerationEnabled
<$true | $false>] [-Name <String>] [-Password <SecureString>] [-
PrimarySmtpAddress <SmtpAddress>] [-RecipientLimits <Unlimited>] [-
RecoverableItemsQuota <Unlimited>] [-RecoverableItemsWarningQuota
<Unlimited>] [-RejectMessagesFrom <MultiValuedProperty>] [-
RejectMessagesFromDLMembers <MultiValuedProperty>] [-
RejectMessagesFromSendersOrMembers <MultiValuedProperty>] [-
RemoteRoutingAddress <ProxyAddress>] [-RemovePicture <SwitchParameter>] [-
RemoveSpokenName <SwitchParameter>] [-RequireSenderAuthenticationEnabled
<$true | $false>] [-ResetPasswordOnNextLogon <$true | $false>] [-
RetainDeletedItemsFor <EnhancedTimeSpan>] [-RetentionComment <String>] [-
RetentionHoldEnabled <$true | $false>] [-RetentionUrl <String>] [-
SamAccountName <String>] [-SecondaryAddress <String>] [-SecondaryDialPlan
<UMDialPlanIdParameter>] [-SendModerationNotifications <Never | Internal |
Always>] [-SimpleDisplayName <String>] [-SingleItemRecoveryEnabled <$true
| $false>] [-StartDateForRetentionHold <DateTime>] [-Type <Regular | Room
| Equipment>] [-UMDtmfMap <MultiValuedProperty>] [-UserCertificate
<MultiValuedProperty>] [-UserPrincipalName <String>] [-
UserSMimeCertificate <MultiValuedProperty>] [-WhatIf [<SwitchParameter>]]
[-WindowsEmailAddress <SmtpAddress>]
```

Examples

EXAMPLE 1

This example configures the mailbox in the service that's associated with the specified mail-enabled user as a room resource mailbox. This example assumes that directory synchronization has been configured.

```
Set-RemoteMailbox davids -Type Room
```

EXAMPLE 2

This example configures delivery restrictions for the mailbox in the service that's associated with the specified mail-enabled user. This example assumes that directory synchronization has been configured.

```
Set-RemoteMailbox kima -AcceptMessagesOnlyFrom davids,
"Executive Team", bill@contoso.com
```

Detailed Description

The **Set-RemoteMailbox** cmdlet configures Exchange attributes for an on-premises mail-enabled user. The configuration set on the on-premises mail-enabled user is synchronized to its associated mailbox in the service.

Note:
Some attributes on mailboxes in the service can only be configured by connecting to the service and using the Set-Mailbox cmdlet.

Directory synchronization must be configured correctly for changes made to an on-premises mail-enabled user to be applied to a mailbox in the service. Changing the configuration of the mailbox in the service isn't immediate and depends on the directory synchronization schedule.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remote mailboxes" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RemoteMailboxIdParameter	The *Identity* parameter specifies the mail-enabled user. You can use one of the following values: • ADObjectID • GUID • Distinguished name (DN)

			• *Domain \SamAccountName* • User principal name (UPN) • Legacy DN • Email address • User alias
AcceptMessagesOnlyFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFrom* parameter specifies the mailbox users and mail-enabled users that can send email messages to this mail-enabled user. You can also specify Exchange as a valid recipient for this parameter. If you configure a mail-enabled user to accept messages only from the Exchange recipient, the mail-enabled user only receives system-generated messages. You can use one of the following values for the valid senders: • DN • Canonical name • GUID • Name • Display name

			• Alias • Exchange DN • Primary SMTP email address By default, this parameter is blank, which enables the mail-enabled user to accept messages from all senders.
AcceptMessagesOnlyFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromDLMembers* parameter specifies the distribution groups whose members are allowed to send email messages to this mail-enabled user. You can use any of the following values for the allowed distribution groups: • DN • Canonical name • GUID • Name • Display name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank, which enables the mail-enabled user to accept messages from all

			senders.
AcceptMessagesOnlyFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromSendersOrMembers* parameter specifies the recipients who are allowed to send email messages to this mail-enabled user. You can use any of the following values for the allowed recipients: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank, which enables the mail-enabled user to accept messages from all senders.
AggregatedMailboxGuids	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
Alias	Optional	System.String	The *Alias* parameter specifies the alias (mail

			nickname) of the mail-enabled user. The alias can be a combination of characters separated by a period with no intervening spaces. Don't use special characters in the alias.
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
ArchiveGuid	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
ArchiveName	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ArchiveName* parameter specifies the name of the archive mailbox. Use this parameter to change the name of the archive.
ArchiveQuota	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ArchiveWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
BypassModerationFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassModerationFromSendersOrMembers* parameter specifies the recipients whose messages bypass moderation when sending

			to this mail-enabled user. You can use any of the following values: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN • Primary SMTP email address This value makes sure that all messages are moderated when this mail-enabled user is configured for moderation. **Note:** Senders designated as moderators for this mail-enabled user are never moderated.
CalendarVersionStoreDisabled	Optional	System.Boolean	The *CalendarVersionStoreDisabled* parameter specifies whether calendar changes in a user's mailbox are logged. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`. By default, all changes to a

			calendar item are recorded in the mailbox of a user to keep older versions of meeting items.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CreateDTMFMap	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
CustomAttribute1	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute10	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.

CustomAttribute11	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute12	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute13	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute14	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute15	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify

			custom attributes. You can use these attributes to store additional information.
CustomAttribute2	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute3	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute4	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute5	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional

			information.
CustomAttribute6	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute7	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute8	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute9	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
DisplayName	Optional	System.String	The *DisplayName*

			parameter specifies the display name for the user account associated with this mail-enabled user.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EmailAddresses	Optional	Microsoft.Exchange.Data.ProxyAddressCollection	The *EmailAddresses* parameter specifies the email alias of the mail-enabled user. All valid Exchange email address types may be used. You can specify multiple values for the *EmailAddresses* parameter as a comma-delimited list. **Important:** Exchange doesn't validate custom addresses for proper formatting. You must ensure that the custom address you specify complies with the format requirements for that address type. Because X.400 addresses are considered custom addresses in Exchange, they're also not validated, and you must provide the

			correct syntax when specifying an X.400 address.
EmailAddressPolicyEnabled	Optional	System.Boolean	The *EmailAddressPolicyEnabled* parameter specifies whether the email addresses for the mail-enabled user are automatically updated based on the email address policies defined. The two possible values for this parameter are `$true` or `$false`. When this parameter is set to `$true`, you can't change the *PrimarySmtpAddress* or *WindowsEmailAddress* parameters.
EndDateForRetentionHold	Optional	System.DateTime	The *EndDateForRetentionHold* parameter specifies the end date for retention hold for messaging records management (MRM). To use this parameter, the *RetentionHoldEnabled* parameter must be set to `$true`.
ExchangeGuid	Optional	System.Guid	This parameter is reserved

			for internal Microsoft use.
ExtensionCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute*

			parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *ExtensionCustomAttribute*

		y	*1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute 1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information

			about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
GrantSendOnBehalfTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *GrantSendOnBehalfTo* parameter specifies the DN of recipients that can send messages on behalf of this mail-enabled user.
HiddenFromAddressListsEnabled	Optional	System.Boolean	The *HiddenFromAddressListsEnabled* parameter specifies whether the mail-enabled user appears in the address list. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to

			access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
ImmutableId	Optional	System.String	The *ImmutableId* parameter is used by Outlook Live Directory Sync (OLSync) and specifies a unique and immutable identifier in the form of an SMTP address for an Exchange mailbox that's used for federated delegation when requesting Security Assertion Markup Language (SAML) tokens.

			If federation is configured for this mail-enabled user and you don't set this parameter when you create the mailbox, Exchange creates the value for the immutable identifier based upon the mailbox's **ExchangeGUID** and the federated account namespace, for example, 7a78e7c8-620e-4d85-99d3-c90d90f29699@mail.contoso.com. You must set the *ImmutableId* parameter if Active Directory Federation Services (AD FS) is deployed to allow single sign-on into off-premises mailboxes and AD FS is configured to use a different attribute than **ExchangeGUID** for sign-on token requests. Both, Exchange and AD FS must request the same token for the same user to ensure proper functionality for cross-premise Exchange

			organizations.
JournalArchiveAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is reserved for internal Microsoft use.
LitigationHoldDate	Optional	System.DateTime	The *LitigationHoldDate* parameter specifies the date when the mail-enabled user and its associated mailbox in the service are placed on litigation hold. The parameter is populated automatically when placing a mailbox on litigation hold. This can be used for informational or reporting purposes. **Note:** When using the Exchange Management Shell to place the mailbox on litigation hold, you can optionally specify any date as the *LitigationHoldDate*, but the mailbox is placed on litigation hold when the cmdlet is run.
LitigationHoldEnabled	Optional	System.Boolean	The *LitigationHoldEnabled* parameter specifies that the mail-enabled user and its associated mailbox in the service are under litigation hold and that messages can't be deleted

			from the mailbox. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`. After a mailbox is placed on litigation hold, deleted items and all versions of changed items are retained in the Recoverable Items folder. Items that are purged from the dumpster are also retained and the items are held indefinitely. If you enable litigation hold, single-item recovery quotas aren't applied.
LitigationHoldOwner	Optional	System.String	The *LitigationHoldOwner* parameter specifies the user who placed the mail-enabled user and the associated mailbox in the service on litigation hold. The parameter is populated automatically when placing a mailbox on litigation hold. This can be used for informational and reporting purposes. **Note:** When using the Shell to place a mailbox on litigation hold, you can optionally specify a string

			value for this parameter.
MailboxContainerGuid	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
MailTip	Optional	System.String	The *MailTip* parameter specifies the message displayed to senders when they start drafting an email message to this mail-enabled user. The *MailTip* parameter message must be less than or equal to 250 characters.
MailTipTranslations	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *MailTipTranslations* parameter specifies additional languages when you want to provide the *MailTip* parameter information for this recipient in multiple languages. For each language, you must provide the locale, followed by a colon and the specific *MailTip* parameter message in that language. Each *MailTip* parameter message must be less than or equal to 250 characters. Multiple languages can be

			separated by commas.
MaxReceiveSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxReceiveSize* parameter specifies the maximum size of email messages that can be received by the mail-enabled user, from 1 kilobyte (KB) through 2,097,151 KB. If not specified, there are no size restrictions.
MaxSendSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxSendSize* parameter specifies the maximum size of email messages that can be sent by the mail-enabled user, from 1 KB through 2,097,151 KB. If not specified, there are no size restrictions.
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users responsible for moderating the messages sent to the mail-enabled user. To designate more than one user, separate the users with commas. This parameter is required if you set the *ModerationEnabled*

			parameter to `$true`. If you leave this parameter blank and there's a user already specified as the manager of this mail-enabled user, the *ModeratedBy* parameter is automatically set by the *ManagedBy* parameter of the distribution group. Otherwise, an error is returned.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable moderation for the mail-enabled user. The two possible values for this parameter are `$true` or `$false`. To enable moderation, set this parameter to `$true`. To disable moderation, set this parameter to `$false`. The default value is `$false`.
Name	Optional	System.String	The *Name* parameter specifies the name of the mail-enabled user.
Password	Optional	System.Security.SecureString	This parameter is reserved for internal Microsoft use.

PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the primary SMTP address.
RecipientLimits	Optional	Microsoft.Exchange.Data.Unlimited	The *RecipientLimits* parameter specifies the maximum number of recipients for messages from this mail-enabled user.
RecoverableItemsQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *RecoverableItemsQuota* parameter specifies the size limit for the Recoverable Items folder for a remote mailbox or remote archive mailbox in the cloud-based service.
RecoverableItemsWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *RecoverableItemsWarningQuota* parameter specifies the size of the Recoverable Item folder before Exchange sends a warning message to the mailbox owner and logs an event to the application event log.
RejectMessagesFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFrom* parameter specifies the recipients from which to reject messages. You can

			use any of the following values to specify the recipients: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank, which enables the mail-enabled user to accept messages from all senders.
RejectMessagesFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromDLMembers* parameter specifies the distribution list members from which to reject messages. You can use any of the following values to specify the recipients: • DN • Alias • Canonical name • Display name • GUID • Name

			• LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank, which enables the mail-enabled user to accept messages from all senders.
RejectMessagesFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromSendersOrMembers* parameter specifies the recipients who aren't allowed to send email messages to this mail-enabled user. You can use any of the following values to specify the recipients: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank, which enables the mail-enabled user to accept messages from all senders.

RemoteRoutingAddress	Optional	Microsoft.Exchange.Data.ProxyAddress	The *RemoteRoutingAddress* parameter specifies the SMTP address of the mailbox in the service that's associated with this mail-enabled user. You shouldn't have to change the remote routing address if the address was automatically configured by Exchange when the mail-enabled user and its associated mailbox in the service were created.
RemovePicture	Optional	System.Management.Automation.SwitchParameter	The *RemovePicture* parameter specifies whether to remove the picture that a user has added to a mailbox.
RemoveSpokenName	Optional	System.Management.Automation.SwitchParameter	The *RemoveSpokenName* parameter specifies whether to remove the spoken name that a user has added to a mailbox.
RequireSenderAuthenticationEnabled	Optional	System.Boolean	The *RequireSenderAuthenticationEnabled* parameter specifies whether to accept messages only

			from authenticated recipients. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`.
ResetPasswordOnNextLogon	Optional	System.Boolean	The *ResetPasswordOnNextLogon* parameter specifies whether to require the mail-enabled users to change their password the next time they sign in to the cloud-based service. The two possible values for this parameter are `$true` or `$false`. If the *ResetPasswordOnNextLogon* parameter is set to `$true`, the mail-enabled users are required to change their password the next time they sign in to the cloud-based service.
RetainDeletedItemsFor	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *RetainDeletedItemsFor* parameter specifies the length of time to keep deleted items. To specify a value, enter it as a time span: dd.hh:mm:ss where d =

			days, h = hours, m = minutes, and s = seconds. For example, to specify a 15-hour interval, enter 15:00:00.
RetentionComment	Optional	System.String	The *RetentionComment* parameter specifies a comment displayed in Microsoft Outlook regarding the user's retention hold status. This comment can be set only if the *RetentionHoldEnabled* parameter is set to `$true`. This comment should be localized to the user's preferred language.
RetentionHoldEnabled	Optional	System.Boolean	The *RetentionHoldEnabled* parameter specifies whether retention hold is enabled for messaging retention policies. The two possible values for this parameter are `$true` or `$false`. To set the start date for retention hold, use the *StartDateForRetentionHold* parameter.

RetentionUrl	Optional	System.String	The *RetentionUrl* parameter specifies the URL or an external web page with additional details about the organization's messaging retention policies. This URL can be used to expose details regarding retention policies in general, which is usually a customized legal or IT website for the company.
SamAccountName	Optional	System.String	The *SamAccountName* parameter specifies the logon name used to support clients and servers running older versions of the operating system, such as Microsoft Windows NT 4.0, Windows 98, Windows 95, and LAN Manager. This attribute must contain fewer than 20 characters. An account name can contain letters, numbers, and the following punctuation marks and symbols: • ! • #

			• $ • % • ^ • & • - • . • _ • { • } • | • ~
SecondaryAddress	Optional	System.String	The *SecondaryAddress* parameter specifies the secondary address used by the Unified Messaging (UM)-enabled user.
SecondaryDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	This parameter is reserved for internal Microsoft use.
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when they send a message to the moderated mail-enabled user. You can use one of the following values: • `Always` • `Internal` • `Never` If you want notifications

			to be sent to all senders, set this parameter to `Always`. If you want notifications to be sent only to the senders who are internal to your organization, set this parameter to `Internal`. To disable all status notifications, set this parameter to `Never`. **Note:** The sender is always notified if the message is rejected by the moderators, regardless of the value of this parameter. The default value is `Never`.
SimpleDisplayName	Optional	System.String	The *SimpleDisplayName* parameter is used to display an alternative description of the object when only a limited set of characters is permitted. This limited set of characters consists of ASCII characters 26 through 126, inclusively.
SingleItemRecoveryEnabled	Optional	System.Boolean	The *SingleItemRecoveryEnabled* parameter specifies

			whether to prevent the Recovery Items folder from being purged. When this parameter is set to `$true`, it prevents the Recovery Items folder from being purged. It also prevents any items from being removed that have been deleted or edited. The possible values for this parameter are `$true` or `$false`. The default value is `$false`.
StartDateForRetentionHold	Optional	System.DateTime	The *StartDateForRetentionHold* parameter specifies the start date for retention hold for MRM. To use this parameter, the *RetentionHoldEnabled* parameter must be set to `$true`.
Type	Optional	Microsoft.Exchange.Management.RecipientTasks.ConvertibleRemoteMailboxSubType	The *Type* parameter specifies the type for the mailbox in the service. You can use the following values: • `Regular` • `Room` • `Equipment`
UMDtmfMap	Optional	Microsoft.Exchange.Data.MultiValuedPropert	This parameter is reserved for internal Microsoft use.

		y	
UserCertificate	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
UserPrincipalName	Optional	System.String	The *UserPrincipalName* parameter specifies a UPN for the user.
UserSMimeCertificate	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsEmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *WindowsEmailAddress* parameter specifies the Windows email address for this mail-enabled user. This address isn't used by Exchange.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

High availability cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-07*

Database Availability Group (DAG) cmdlets

Get-DatabaseAvailabilityGroup

New-DatabaseAvailabilityGroup

Remove-DatabaseAvailabilityGroup

Restore-DatabaseAvailabilityGroup

Set-DatabaseAvailabilityGroup

Start-DatabaseAvailabilityGroup

Stop-DatabaseAvailabilityGroup

Add-DatabaseAvailabilityGroupServer

Remove-DatabaseAvailabilityGroupServer

DAG network cmdlets

Get-DatabaseAvailabilityGroupNetwork

New-DatabaseAvailabilityGroupNetwork

Remove-DatabaseAvailabilityGroupNetwork

Set-DatabaseAvailabilityGroupNetwork

Mailbox database copy cmdlets

Move-ActiveMailboxDatabase

Add-MailboxDatabaseCopy

Remove-MailboxDatabaseCopy

Resume-MailboxDatabaseCopy

Set-MailboxDatabaseCopy

Suspend-MailboxDatabaseCopy

Update-MailboxDatabaseCopy

Health and status cmdlets

Get-MailboxDatabaseCopyStatus

Test-ReplicationHealth

Move-ActiveMailboxDatabase

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Move-ActiveMailboxDatabase** cmdlet to perform a database or server switchover.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Move-ActiveMailboxDatabase -Identity <DatabaseIdParameter> <COMMON
PARAMETERS>


Move-ActiveMailboxDatabase -Server <MailboxServerIdParameter> <COMMON
PARAMETERS>


COMMON PARAMETERS: [-ActivateOnServer <MailboxServerIdParameter>] [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
MountDialOverride <None | Lossless | GoodAvailability | BestAvailability |
BestEffort>] [-MoveComment <String>] [-SkipActiveCopyChecks
<SwitchParameter>] [-SkipClientExperienceChecks <SwitchParameter>] [-
SkipHealthChecks <SwitchParameter>] [-SkipLagChecks <SwitchParameter>] [-
SkipMaximumActiveDatabasesChecks <SwitchParameter>] [-TerminateOnWarning
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example performs a switchover of the database DB2 to the Mailbox server MBX1. When the command completes, MBX1 hosts the active copy of DB2. Because the *MountDialOverride* parameter is set to *None*, MBX1 mounts the database using its own defined database auto mount dial settings.

```
Move-ActiveMailboxDatabase DB2 -ActivateOnServer MBX1 -
MountDialOverride:None
```

EXAMPLE 2

This example performs a switchover of the database DB1 to the Mailbox server MBX3. When the command completes, MBX3 hosts the active copy of DB1. Because the *MountDialOverride* parameter is specified with a value of *Good Availability*, MBX3 mounts the database using a database auto mount dial setting of *GoodAvailability*.

```
Move-ActiveMailboxDatabase DB1 -ActivateOnServer MBX3 -
MountDialOverride:GoodAvailability
```

EXAMPLE 3

This example performs a switchover of the database DB3 to the Mailbox server MBX4. When the command completes, MBX4 hosts the active copy of DB3. Because the *MountDialOverride* parameter isn't specified, MBX4 mounts the database using a database auto mount dial setting of *Lossless*.

```
Move-ActiveMailboxDatabase DB3 -ActivateOnServer MBX4
```

EXAMPLE 4

This example performs a server switchover for the Mailbox server MBX1. All active mailbox database copies on MBX1 will be activated on one or more other Mailbox servers with healthy copies of the active databases on MBX1.

```
Move-ActiveMailboxDatabase -Server MBX1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Identity* parameter specifies the identity of the mailbox database being activated.
Server	Required	Microsoft.Exchange.Configuration.Tasks.MailboxServerIdParameter	The *Server* parameter specifies the identity of the server from which to move all active mailbox databases.
ActivateOnServer	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxServerIdParameter	The *ActivateOnServer* parameter specifies the name of the Mailbox server on which the mailbox database copy should be activated.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Da	The *DomainController*

		ta.Fqdn	parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
MountDialOverride	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DatabaseMountDialOverride	The *MountDialOverride* parameter is used to override the auto database mount dial (AutoDatabaseMountDial) setting for the target server and specify an alternate setting. The following are possible values: • `None` When using this value, the currently configured auto database mount dial setting on the target server will be used. • `Lossless` This is the default value. When using this value, the database doesn't automatically mount until all log files that were generated on the original active copy have been copied to the passive copy. • `GoodAvailability` If you specify this value, the database automatically mounts immediately after a

			failover if the copy queue length is less than or equal to 6. If the copy queue length is greater than 6, the database doesn't automatically mount. When the copy queue length is less than or equal to 6, Microsoft Exchange attempts to replicate the remaining logs to the passive copy and then mounts the database. • `BestEffort` If you specify this value, the database automatically mounts regardless of the size of the copy queue length. Because the database will mount with any amount of log loss, using this value could result in a large amount of data loss. • `BestAvailability` If you specify this value, the database automatically mounts immediately after a failover if the copy queue length is less than or equal to 12. The copy queue length is the number of logs recognized by the passive copy that needs to be replicated. If the copy queue length is more than 12, the database doesn't automatically mount. When the copy queue

			length is less than or equal to 12, Exchange attempts to replicate the remaining logs to the passive copy and then mounts the database.
MoveComment	Optional	System.String	The *MoveComment* parameter specifies an optional administrative reason for the move operation. The comment is recorded in the Event log.
SkipActiveCopyChecks	Optional	System.Management.Automation.SwitchParameter	The *SkipActiveCopyChecks* parameter specifies whether to skip checking the current active copy to see if it's currently a seeding source for any passive databases. Be aware that when using this parameter, you can move a database that's currently a seeding source, which cancels the seed operation.
SkipClientExperienceChecks	Optional	System.Management.Automation.SwitchParameter	The *SkipClientExperienceChecks* parameter specifies whether to skip the search catalog (content index) state check to see if the

			search catalog is healthy and up to date. If the search catalog for the database copy you're activating is in an unhealthy or unusable state and you use this parameter to skip the search catalog health check and activate the database copy, you will need to either re-crawl or reseed the search catalog.
SkipHealthChecks	Optional	System.Management.Automation.SwitchParameter	The *SkipHealthChecks* parameter specifies whether to bypass passive copy health checks. With the *SkipHealthChecks* parameter, you can move the active copy to a database copy that's in the Failed state. This parameter should be used only if the initial attempt to move the active database has failed. This is because the *SkipHealthChecks* parameter performs additional validation to ensure that the log files are consistent, which can

			take a considerable amount of time.
SkipLagChecks	Optional	System.Management.Automation.SwitchParameter	The *SkipLagChecks* parameter specifies whether to allow a copy to be activated that has replay and copy queues outside of the configured criteria.
SkipMaximumActiveDatabasesChecks	Optional	System.Management.Automation.SwitchParameter	The *SkipMaximumActiveDatabasesChecks* is used to skip checking the value of MaximumPreferredActiveDatabases during the best copy and server selection (BCSS) process. Any configured value for MaximumActiveDatabases will still be honored during the BCSS process and by the Information Store.
TerminateOnWarning	Optional	System.Management.Automation.SwitchParameter	The *TerminateOnWarning* parameter specifies whether to terminate the task and output an error message if a warning is encountered during the switchover operation.
WhatIf	Optional	System.Management.	The *WhatIf* switch

		Automation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DatabaseAvailabilityGroup

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-DatabaseAvailabilityGroup** cmdlet to obtain a variety of configuration settings, status, and other information about a database availability group (DAG).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-DatabaseAvailabilityGroup [-Identity
<DatabaseAvailabilityGroupIdParameter>] [-DomainController <Fqdn>] [-
Status <SwitchParameter>]
```

Examples

EXAMPLE 1

This example displays the basic properties of the DAG DAG1. The output for the command is formatted as a list.

```
Get-DatabaseAvailabilityGroup DAG1 | Format-List
```

EXAMPLE 2

This example displays the properties of the DAG DAG2. Because it includes the *Status* parameter, the task also displays real-time status information for DAG2, such as the current list of operational servers, and the server currently holding the Primary Active Manager role. In addition, several properties of the DAG, such as the witness server and directory configuration information are also displayed. The output for the command is formatted as a list.

```
Get-DatabaseAvailabilityGroup DAG2 -Status | Format-List
```

Detailed Description

In addition to obtaining a list of DAG members, the **Get-DatabaseAvailabilityGroup** cmdlet can also be used to view real-time status information about a DAG, such as:

- **OperationalServers**
- **PrimaryActiveManager**
- **ReplicationPort**
- **NetworkNames**
- **WitnessShareInUse**

Use the *Status* parameter with the command to include the values for these listed properties. Without the *Status* parameter, the values returned for these properties are blank.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability group properties" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupIdParameter	The *Identity* parameter specifies the name of the DAG to query.
Status	Optional	System.Management.Automation.SwitchParameter	The *Status* parameter instructs the command to query Active Directory for additional information, and to include real-time status information in the output.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-DatabaseAvailabilityGroup

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-01*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-DatabaseAvailabilityGroup** cmdlet to create a database availability group (DAG).

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-DatabaseAvailabilityGroup -Name <String> [-Confirm
[<SwitchParameter>]] [-DagConfiguration
<DatabaseAvailabilityGroupConfigurationIdParameter>] [-
DatabaseAvailabilityGroupIpAddresses <IPAddress[]>] [-DomainController
<Fqdn>] [-ThirdPartyReplication <Disabled | Enabled>] [-WhatIf
[<SwitchParameter>]] [-WitnessDirectory <NonRootLocalLongFullPath>] [-
WitnessServer <FileShareWitnessServerName>]
```

Examples

EXAMPLE 1

This example creates the DAG DAG1 that's configured to use a witness server of CAS1, and a local directory of C:\DAG1. DAG1 is also configured to use DHCP for the DAG's IP addresses.

```
New-DatabaseAvailabilityGroup -Name DAG1 -WitnessServer
CAS1 -WitnessDirectory C:\DAG1
```

EXAMPLE 2

This example creates the DAG DAG2. The system automatically selects a Client Access server without the Mailbox server role in the same site as the DAG to use as the witness server. DAG2 is assigned a single static IP address because the MAPI network for DAG2 contains or will contain a single subnet (10.0.0.x).

```
New-DatabaseAvailabilityGroup -Name DAG2 -
DatabaseAvailabilityGroupIpAddresses 10.0.0.8
```

EXAMPLE 3

This example creates the DAG DAG3. DAG3 is configured to use CAS1 for the witness server, and a witness directory on CAS1 of C:\DAG3. DAG3 is assigned multiple static IP addresses because the MAPI network for the DAG contains or will contain multiple subnets (10.0.0.x and 192.168.0.x).

```
New-DatabaseAvailabilityGroup -Name DAG3 -WitnessServer
CAS1 -WitnessDirectory C:\DAG3 -
DatabaseAvailabilityGroupIpAddresses 10.0.0.8,192.168.0.8
```

EXAMPLE 4

This example creates the DAG DAG4 configured to use DHCP. In addition, the witness server is automatically selected by the system and the default witness directory is created.

```
New-DatabaseAvailabilityGroup -Name DAG4
```

EXAMPLE 5

This example creates the DAG DAG5 without a cluster administrative access point (Windows Server 2012 R2 and later DAG members only). DAG5 is configured to use CAS1 for the witness server, and a witness directory on CAS1 of C:\DAG3.

```
New-DatabaseAvailabilityGroup -Name DAG5 -WitnessServer
CAS1 -WitnessDirectory C:\DAG3 -
DatabaseAvailabilityGroupIpAddresses
([System.Net.IPAddress])::None
```

Detailed Description

When creating a DAG, you need to specify a valid computer name for the DAG no longer than 15 characters that's unique within the Active Directory forest. In addition, each DAG is configured with a witness server and witness directory. The witness server and its directory are used only for quorum purposes where there's an even number of members in the DAG. You don't need to create the witness directory in advance. Exchange automatically creates and secures the directory for you on the witness server. The directory shouldn't be used for any purpose other than for the DAG witness server.

The requirements for the witness server are as follows:

- The witness server can't be a member of the DAG.
- The witness server must be in the same Active Directory forest as the DAG.
- The witness server must be running the Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2, Windows Server 2008, Windows Server 2003 R2, or Windows Server 2003 operating system.
- A single server can serve as a witness for multiple DAGs; however, each DAG requires its own witness directory.

We recommend that you use a Client Access server running on Microsoft Exchange Server 2013 in the Active Directory site containing the DAG. This allows the witness server and directory to remain under the control of an Exchange administrator.

The following combinations of options and behaviors are available:

- You can specify only a name for the DAG. In this scenario, the task searches for a Client Access server in the local Active Directory site that doesn't have the Mailbox server role installed, and it automatically creates the default directory and share on that server and uses that Client Access server as the witness server.
- You can specify a name for the DAG, the witness server that you want to use, and the directory you want created and shared on the witness server.
- You can specify a name for the DAG and the witness server that you want to use. In this scenario,

the task creates the default directory on the specified witness server.

- You can specify a name for the DAG and specify the directory you want created and shared on the witness server. In this scenario, the task searches for a Client Access server in the local Active Directory site that doesn't have the Mailbox server role installed, and it automatically creates the specified directory on that server, shares the directory, and uses that Client Access server as the witness server.

Important:
If the witness server you specify isn't an Exchange 2013 server, you must add the Exchange Trusted Subsystem universal security group (USG) to the local Administrators group on the witness server. If the witness server is a directory server, you must add the Exchange Trusted Subsystem USG to the Builtin\Administrators group. These security permissions are necessary to ensure that Exchange can create a directory and share on the witness server as needed.

In addition to providing a name for the DAG, one or more IP addresses must also be assigned to the DAG, unless you are creating a DAG without a cluster administrative access point, which can be done only with DAG members running Windows Server 2012 R2. You can assign static IP addresses to the DAG by using the *DatabaseAvailabilityGroupIpAddresses* parameter. If you omit this parameter or configure the parameter with a value of 0.0.0.0, the task attempts to use Dynamic Host Configuration Protocol (DHCP) to obtain the necessary IP addresses. If you configure the parameter with a value of 255.255.255.255, the task attempts to create a DAG without a cluster administrative access point.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies a unique name for the new DAG of up to 15 characters. The name you use must not conflict with any computer name in the organization.
Confirm	Optional	System.Management.Automation.SwitchPara	The *Confirm* switch causes the command to

		meter	pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DagConfiguration	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupConfigurationIdParameter	This parameter is reserved for internal Microsoft use.
DatabaseAvailabilityGroupIpAddresses	Optional	System.Net.IPAddress[]	The *DatabaseAvailabilityGroupIpAddresses* parameter specifies one or more static IPv4 addresses to the DAG when a Mailbox server is added to a DAG. If you omit the *DatabaseAvailabilityGroupIpAddresses* parameter when creating a DAG, the system attempts to lease one or more IPv4 addresses from a DHCP server in your organization to assign to the DAG. Setting the

			DatabaseAvailabilityGroupIpAddresses parameter to a value of `0.0.0.0` also configures the DAG to use DHCP. Setting the *DatabaseAvailabilityGroupIpAddresses* parameter to a value of `255.255.255.255` creates a DAG without a cluster administrative access point.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ThirdPartyReplication	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ThirdPartyReplicationMode	The *ThirdPartyReplication* parameter specifies to configure and enable a DAG to use third-party replication that leverages the Exchange Third Party Replication API instead of the built-in continuous replication. After this

			mode is enabled, it can't be changed.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WitnessDirectory	Optional	Microsoft.Exchange.Data.NonRootLocalLongFullPath	The *WitnessDirectory* parameter specifies the name of the directory on the witness server used to store file share witness data. The directory and share should be hosted on an Exchange server other than any of the Mailbox servers in the DAG. This allows an Exchange administrator to maintain operational control over the directory. The specified directory must not be

			in use by any other DAGs or used for any purpose other than for the witness server. If you omit this option, the default witness directory is used.
WitnessServer	Optional	Microsoft.Exchange.Data.FileShareWitnessServerName	The *WitnessServer* parameter specifies the name of a server used as a quorum witness when the DAG contains an even number of members. The selected server must not be a member of the DAG that's configured to use it. If you omit the *WitnessServer* parameter, the task tries to automatically select a Client Access server without the Mailbox server role in the same Active Directory site as the DAG to use as the witness server.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-DatabaseAvailabilityGroup

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-DatabaseAvailabilityGroup** cmdlet to delete an empty database availability group (DAG). Before you can delete a DAG, you must first remove all Mailbox servers from the DAG.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-DatabaseAvailabilityGroup -Identity
<DatabaseAvailabilityGroupIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example, deletes the DAG DAG1.

```
Remove-DatabaseAvailabilityGroup -Identity DAG1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupIdParameter	The *Identity* parameter specifies the name of the DAG to be removed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Restore-DatabaseAvailabilityGroup

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Restore-DatabaseAvailabilityGroup** cmdlet as part of a datacenter switchover of a database availability group (DAG).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Restore-DatabaseAvailabilityGroup -Identity
<DatabaseAvailabilityGroupIdParameter> [-ActiveDirectorySite
<AdSiteIdParameter>] [-AlternateWitnessDirectory
<NonRootLocalLongFullPath>] [-AlternateWitnessServer
<FileShareWitnessServerName>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-UsePrimaryWitnessServer <SwitchParameter>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example activates member servers in the DAG DAG1 for the Active Directory site Portland. In

this example, the values for the *AlternateWitnessServer* parameter and the *AlternateWitnessDirectory* parameter had been previously set by using the **Set-DatabaseAvailabilityGroup** cmdlet. Thus, there is no need to specify them here.

```
Restore-DatabaseAvailabilityGroup -Identity DAG1 -
ActiveDirectorySite Portland
```

EXAMPLE 2

This example activates member servers in the DAG DAG1 for the Active Directory site Redmond. In this example, the values for the *AlternateWitnessServer* parameter and the *AlternateWitnessDirectory* parameter are being set as part of the activation process.

```
Restore-DatabaseAvailabilityGroup -Identity DAG1 -
ActiveDirectorySite Redmond -AlternateWitnessServer CAS4 -
AlternateWitnessDirectory D:\DAG1
```

Detailed Description

You can also use this cmdlet for disaster recovery purposes to restore functionality to a DAG that has lost quorum due to one or more DAG members being offline for an extended period. Before running this cmdlet, you must first run the Stop-DatabaseAvailabilityGroup cmdlet.

The **Restore-DatabaseAvailabilityGroup** cmdlet can be run against a DAG only when the DAG is configured with a *DatacenterActivationMode* parameter value of `DagOnly`. For more information about the *DatacenterActivationMode* parameter, see Datacenter Activation Coordination mode.

You can use the **Set-DatabaseAvailabilityGroup** cmdlet to configure the value for the *DatacenterActivationMode* parameter.

The **Restore-DatabaseAvailabilityGroup** cmdlet performs several operations that affect the structure and membership of the DAG's cluster. This task does the following:

- Forcibly evicts the servers listed on the StoppedMailboxServers list from the DAG's cluster, thereby reestablishing quorum for the cluster enabling the surviving DAG members to start and provide service.
- Configures the DAG to use the alternate witness server if there is an even number of surviving DAG members, or a single surviving DAG member.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupIdParameter	The *Identity* parameter specifies the name of the DAG being manipulated.
ActiveDirectorySite	Optional	Microsoft.Exchange.Configuration.Tasks.AdSiteIdParameter	The *ActiveDirectorySite* parameter specifies the site containing the DAG members to be restored.
AlternateWitnessDirectory	Optional	Microsoft.Exchange.Data.NonRootLocalLongFullPath	The *AlternateWitnessDirectory* parameter specifies the name of an alternate directory used to store witness data. The specified directory must not be in use by any other DAGs or used for any other purpose. This value can be populated ahead by using the **Set-DatabaseAvailabilityGroup** cmdlet.
AlternateWitnessServer	Optional	Microsoft.Exchange.Data.FileShareWitnessServerName	The *AlternateWitnessServer* parameter specifies the name of a new witness server for the DAG as part of a site activation process. This value can

			be populated ahead by using the **Set-DatabaseAvailabilityGroup** cmdlet.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
UsePrimaryWitnessServer	Optional	System.Management.Automation.SwitchParameter	The *UsePrimaryWitnessServer* parameter specifies that the DAG's currently configured witness server should be used if a witness is needed by the DAG

			members being activated.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-DatabaseAvailabilityGroup

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-DatabaseAvailabilityGroup** cmdlet to configure properties of a database availability

group (DAG).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-DatabaseAvailabilityGroup -Identity
<DatabaseAvailabilityGroupIdParameter> [-AllowCrossSiteRpcClientAccess
<SwitchParameter>] [-AlternateWitnessDirectory <NonRootLocalLongFullPath>]
[-AlternateWitnessServer <FileShareWitnessServerName>] [-
AutoDagAllServersInstalled <$true | $false>] [-AutoDagAutoReseedEnabled
<$true | $false>] [-AutoDagBitlockerEnabled <$true | $false>] [-
AutoDagDatabaseCopiesPerDatabase <Int32>] [-AutoDagDatabaseCopiesPerVolume
<Int32>] [-AutoDagDatabasesRootFolderPath <NonRootLocalLongFullPath>] [-
AutoDagDiskReclaimerEnabled <$true | $false>] [-AutoDagFIPSCompliant
<$true | $false>] [-AutoDagTotalNumberOfDatabases <Int32>] [-
AutoDagTotalNumberOfServers <Int32>] [-AutoDagVolumesRootFolderPath
<NonRootLocalLongFullPath>] [-Confirm [<SwitchParameter>]] [-
DagConfiguration <DatabaseAvailabilityGroupConfigurationIdParameter>] [-
DatabaseAvailabilityGroupIpAddresses <IPAddress[]>] [-
DatacenterActivationMode <Off | DagOnly>] [-DiscoverNetworks
<SwitchParameter>] [-DomainController <Fqdn>] [-MailboxLoadBalanceEnabled
<$true | $false>] [-MailboxLoadBalanceMaximumEdbFileSize
<ByteQuantifiedSize>] [-MailboxLoadBalanceOverloadedThreshold <Int32>] [-
MailboxLoadBalanceRelativeLoadCapacity <Int32>] [-
MailboxLoadBalanceUnderloadedThreshold <Int32>] [-
ManualDagNetworkConfiguration <$true | $false>] [-NetworkCompression
<Disabled | Enabled | InterSubnetOnly | SeedOnly>] [-NetworkEncryption
<Disabled | Enabled | InterSubnetOnly | SeedOnly>] [-
ReplayLagManagerEnabled <$true | $false>] [-ReplicationPort <UInt16>] [-
SkipDagValidation <SwitchParameter>] [-WhatIf [<SwitchParameter>]] [-
WitnessDirectory <NonRootLocalLongFullPath>] [-WitnessServer
<FileShareWitnessServerName>]
```

Examples

EXAMPLE 1

This example sets the witness directory to C:\DAG1DIR for the DAG DAG1.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
WitnessDirectory C:\DAG1DIR
```

EXAMPLE 2

This example preconfigures an alternate witness server of CAS3 and an alternate witness directory of C:\DAGFileShareWitnesses\DAG1.contoso.com for the DAG DAG1.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
AlternateWitnessDirectory C:\DAGFileShareWitnesses
\DAG1.contoso.com -AlternateWitnessServer CAS3
```

EXAMPLE 3

This example configures the DAG DAG1 to use DHCP to obtain an IP address.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
DatabaseAvailabilityGroupIpAddresses 0.0.0.0
```

EXAMPLE 4

This example configures the DAG DAG1 to use a static IP address of 10.0.0.8.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
DatabaseAvailabilityGroupIpAddresses 10.0.0.8
```

EXAMPLE 5

This example configures the multi-subnet DAG DAG1 with multiple static IP addresses.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
DatabaseAvailabilityGroupIpAddresses 10.0.0.8,10.0.1.8
```

EXAMPLE 6

This example configures TCP port 63132 as the port used by replication for the DAG DAG1.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
ReplicationPort 63132
```

Note:

After changing the default replication port for a DAG, you must manually modify the Windows Firewall exceptions on each member of the DAG to allow communication to occur over the specified port.

EXAMPLE 7

This example configures the DAG DAG1 for DAC mode.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
DatacenterActivationMode DagOnly
```

EXAMPLE 8

This example configures the DAG DAG1 for AutoReseed using custom mount point paths and 4 databases per volume.

```
Set-DatabaseAvailabilityGroup -Identity DAG1 -
AutoDagVolumesRootFolderPath C:\ExchVols -
AutoDagDatabasesRootFolderPath C:\ExchDBs -
AutoDagDatabaseCopiesPerVolume 4
```

Detailed Description

The **Set-DatabaseAvailabilityGroup** cmdlet enables you to manage DAG properties that can't be

managed from the Exchange Administration Center (EAC), such as configuring network discovery, selecting the TCP port used for replication, and enabling datacenter activation coordination (DAC) mode.

DAG property values are stored in both Active Directory and the cluster database. Because some properties are stored in the cluster database, the underlying cluster for the DAG must have quorum to set the properties for:

- ReplicationPort
- NetworkCompression
- NetworkEncryption
- DiscoverNetworks

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the"Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupIdParameter	The *Identity* parameter specifies the name of the DAG to be modified.
AllowCrossSiteRpcClientAccess	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
AlternateWitnessDirectory	Optional	Microsoft.Exchange.Data.NonRootLocalLongFullPath	The *AlternateWitnessDirectory* parameter specifies the name of an alternate directory that's used to store file share witness data. The specified directory must not be in use by any other DAGs or

			used for any other purpose. This parameter is used only as part of a datacenter switchover process. If the DAG is extended across multiple datacenters in a site resilience configuration, we recommend preconfiguring the alternate witness server and directory.
AlternateWitnessServer	Optional	Microsoft.Exchange.Data.FileShareWitnessServerName	The *AlternateWitnessServer* parameter specifies the name of an alternate server that's used to store file share witness data. The specified server must not be a member of the DAG that's configured to use it. This parameter is used only as part of a datacenter switchover process. If the DAG is extended across multiple datacenters in a site resilience configuration, we

			recommend preconfiguring the alternate witness server and directory.
AutoDagAllServersInstalled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
AutoDagAutoReseedEnabled	Optional	System.Boolean	The *AutoDagAutoReseedEnabled* is used to enable or disable Autoreseed. The default value is True (enabled).
AutoDagBitlockerEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
AutoDagDatabaseCopiesPerDatabase	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
AutoDagDatabaseCopiesPerVolume	Optional	System.Int32	The *AutoDagDatabaseCopiesPerVolume* parameter is used to specify the configured number of database copies per volume. This parameter is used only with AutoReseed.
AutoDagDatabasesRootFolderPath	Optional	Microsoft.Exchange.Data.NonRootLocalLongFu	The *AutoDagDatabasesRoo*

		llPath	*tFolderPath* parameter specifies the directory containing the database mount points when using AutoReseed. This parameter is required when using AutoReseed. AutoReseed uses a default path of C:\ExchangeDatabases.
AutoDagDiskReclaimerEnabled	Optional	System.Boolean	The *AutoDagDiskReclaimerEnabled* is used to enable or disable the volume formatting functions used by Autoreseed. The default value is True (enabled). If you set this to False, you will need to manually format the volume before the database(s) can be reseeded.
AutoDagFIPSCompliant	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
AutoDagTotalNumberOfDatabases	Optional	System.Int32	This parameter is reserved for internal

			Microsoft use.
AutoDagTotalNumberOfServers	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
AutoDagVolumesRootFolderPath	Optional	Microsoft.Exchange.Data.NonRootLocalLongFullPath	The *AutoDagVolumesRootFolderPath* parameter specifies the volume containing the mount points for all disks, including spare disks, when using the AutoReseed feature of the DAG. This parameter is required when using AutoReseed. AutoReseed uses a default path of C:\ExchangeVolumes.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DagConfiguration	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupConfigurationIdParameter	This parameter is reserved for internal Microsoft use.
DatabaseAvailabilityGroupIpAddresses	Optional	System.Net.IPAddress[]	The *DatabaseAvailabilityGroupIpAddresses* parameter specifies one or more static IP addresses to the DAG when a Mailbox server is added to a DAG. If you omit the *DatabaseAvailabilityGroupIpAddresses* parameter when creating a DAG, the system attempts to lease one or more IP addresses from a Dynamic Host Configuration Protocol (DHCP) server in your organization to assign to the DAG. You must specify this parameter each time an additional IP address is added to the DAG, such as in the case of multi-subnet DAGs. You must also specify all IP addresses

			previously assigned to the DAG each time the *DatabaseAvailabilityGroupIpAddresses* parameter is used. Setting the *DatabaseAvailabilityGroupIpAddresses* parameter to a value of `0.0.0.0` automatically configures the DAG to use DHCP.
DatacenterActivationMode	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DatacenterActivationModeOption	The *DatacenterActivationMode* parameter specifies whether datacenter activation mode is disabled (`off`) or enabled for the DAG (`DagOnly`).
DiscoverNetworks	Optional	System.Management.Automation.SwitchParameter	The *DiscoverNetworks* parameter specifies whether to force a rediscovery of the network and network interfaces. By default, internal network heartbeats are sent between DAG members on the same subnet. If there's no

			response to the heartbeats, network discovery is performed automatically by the system. If you add or remove networks or change DAG network subnets, you can force rediscovery of all DAG networks by using the *DiscoverNetworks* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
MailboxLoadBalanceEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
MailboxLoadBalanceMaximumEdbFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
MailboxLoadBalanceOverloadedThreshold	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
MailboxLoadBalanceRe	Optional	System.Int32	This parameter is

lativeLoadCapacity			reserved for internal Microsoft use.
MailboxLoadBalanceUnderloadedThreshold	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
ManualDagNetworkConfiguration	Optional	System.Boolean	The *ManualDagNetworkConfiguration* parameter specifies whether DAG networks should be automatically configured. If this parameter is set to False, DAG networks are automatically configured. If this parameter is set to True, you must manually configure DAG networks.
NetworkCompression	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DatabaseAvailabilityGroup+NetworkOption	The *NetworkCompression* parameter specifies whether network compression is disabled on all networks (`Disabled`), enabled on all networks (`Enabled`), enabled for inter-subnet communication

			only (`InterSubnetOnly`), or enabled only for seeding (`SeedOnly`).
NetworkEncryption	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DatabaseAvailabilityGroup+NetworkOption	The *NetworkEncryption* parameter specifies whether network encryption is disabled on all networks (`Disabled`), enabled on all networks (`Enabled`), enabled for inter-subnet communication only (`InterSubnetOnly`), or enabled only for seeding (`SeedOnly`).
ReplayLagManagerEnabled	Optional	System.Boolean	The *ReplayLagManagerEnabled* parameter specifies whether to disable the automatic playdown of log files for a lagged database copy.
ReplicationPort	Optional	System.UInt16	The *ReplicationPort* parameter specifies a Transmission Control Protocol (TCP) port for replication (log shipping and seeding)

			activity. If this parameter isn't specified, the default port for replication is TCP 64327.
SkipDagValidation	Optional	System.Management.Automation.SwitchParameter	The *SkipDagValidation* switch specifies whether to bypass the validation of the DAG's quorum model and the health check on the DAG's witness when configuring the DAG.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WitnessDirectory	Optional	Microsoft.Exchange.Data.NonRootLocalLongFullPath	The *WitnessDirectory* parameter specifies the name of the directory on the server that's used to store file share

			witness data. The specified directory must not be in use by any other DAGs.
WitnessServer	Optional	Microsoft.Exchange.Data.FileShareWitnessServerName	The *WitnessServer* parameter specifies the name of a server that will act as a witness for the DAG. The server specified can't be a member of the DAG.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Start-DatabaseAvailabilityGroup

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-15*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Start-DatabaseAvailabilityGroup** cmdlet to reincorporate one or more previously failed members of a database availability group (DAG).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Start-DatabaseAvailabilityGroup -ActiveDirectorySite <AdSiteIdParameter> -
Identity <DatabaseAvailabilityGroupIdParameter> [-ConfigurationOnly
<SwitchParameter>] [-QuorumOnly <SwitchParameter>] <COMMON PARAMETERS>
```

```
Start-DatabaseAvailabilityGroup -Identity
<DatabaseAvailabilityGroupIdParameter> -MailboxServer
<MailboxServerIdParameter> [-ConfigurationOnly <SwitchParameter>] [-
QuorumOnly <SwitchParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example starts the Mailbox server MBX2 in the DAG DAG1.

```
Start-DatabaseAvailabilityGroup -Identity DAG1 -
MailboxServer MBX2
```

EXAMPLE 2

This example starts the members of the DAG DAG1 in the Active Directory site Redmond.

```
Start-DatabaseAvailabilityGroup -Identity DAG1 -
ActiveDirectorySite Redmond
```

Detailed Description

The **Start-DatabaseAvailabilityGroup** cmdlet is used to activate DAG members in a recovered datacenter after a datacenter switchover, as part of the switchback process to the recovered datacenter. The **Start-DatabaseAvailabilityGroup** cmdlet manipulates configuration and state so that the servers are incorporated into the operating DAG, and joined to the DAG's underlying cluster. The **Move-ActiveMailboxDatabase** cmdlet is then used to activate databases in the primary datacenter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
ActiveDirectorySite	Required	Microsoft.Exchange.Configuration.Tasks.AdSit	The *ActiveDirectorySite* parameter specifies

		eIdParameter	whether to start all DAG members in the specified site.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupIdParameter	The *Identity* parameter specifies the name of the DAG being started.
MailboxServer	Required	Microsoft.Exchange.Configuration.Tasks.MailboxServerIdParameter	The *MailboxServer* parameter specifies whether to start a single DAG member.
ConfigurationOnly	Optional	System.Management.Automation.SwitchParameter	The *ConfigurationOnly* switch specifies whether to update the Active Directory properties with the start action, but doesn't perform a start of the DAG or any members.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
QuorumOnly	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Stop-DatabaseAvailabilityGroup

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-15*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Stop-DatabaseAvailabilityGroup** cmdlet to mark a member of a database availability group (DAG) as failed, or to mark all DAG members in a specific Active Directory site as failed.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Stop-DatabaseAvailabilityGroup -ActiveDirectorySite <AdSiteIdParameter> -
Identity <DatabaseAvailabilityGroupIdParameter> [-ConfigurationOnly
<SwitchParameter>] [-QuorumOnly <SwitchParameter>] <COMMON PARAMETERS>


Stop-DatabaseAvailabilityGroup -Identity
<DatabaseAvailabilityGroupIdParameter> -MailboxServer
<MailboxServerIdParameter> [-ConfigurationOnly <SwitchParameter>] [-
QuorumOnly <SwitchParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example stops the Mailbox server MBX2 in the DAG DAG1.

```
Stop-DatabaseAvailabilityGroup -Identity DAG1 -
MailboxServer MBX2
```

EXAMPLE 2

This example stops all members in the DAG DAG1 in the Active Directory site Redmond.

```
Stop-DatabaseAvailabilityGroup -Identity DAG1 -
ActiveDirectorySite Redmond
```

EXAMPLE 3

This example stops the Mailbox server MBX3, which is currently offline, in the DAG DAG2.

```
Stop-DatabaseAvailabilityGroup -Identity DAG2 -
MailboxServer MBX3 -ConfigurationOnly
```

Detailed Description

The **Stop-DatabaseAvailabilityGroup** cmdlet is used during a datacenter switchover. This cmdlet is used to mark one or members of the DAG as failed (also known as *stopped*).The **Stop-DatabaseAvailabilityGroup** cmdlet can be run against a DAG only when the DAG is configured with a *DatacenterActivationMode* value of `Dagonly`.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
ActiveDirectorySite	Required	Microsoft.Exchange.Configuration.Tasks.AdSiteIdParameter	The *ActiveDirectorySite* parameter specifies the Active Directory site containing the DAG members to stop (for example, stop all DAG members in a particular Active Directory site).
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupIdParameter	The *Identity* parameter specifies the name of the DAG being stopped.
MailboxServer	Required	Microsoft.Exchange.Configuration.Tasks.MailboxServerIdParameter	The *MailboxServer* parameter specifies a single DAG member to stop. If Datacenter

			Activation Coordination mode is enabled for the DAG and all DAG members are in the same Active Directory site, use the *MailboxServer* parameter to stop individual DAG members instead of the *ActiveDirectorySite* parameter when stopping failed DAG members.
ConfigurationOnly	Optional	System.Management.Automation.SwitchParameter	The *ConfigurationOnly* parameter updates the Active Directory properties with the stop action, but doesn't perform a stop of the DAG or any members. This parameter must be used when the DAG member servers are offline, but Active Directory is up and accessible in the primary datacenter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
QuorumOnly	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DatabaseAvailabilityGroupNetwork

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-DatabaseAvailabilityGroupNetwork** cmdlet to display configuration and state information for a database availability group (DAG) network.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-DatabaseAvailabilityGroupNetwork [-Identity
<DatabaseAvailabilityGroupNetworkIdParameter>] [-DomainController <Fqdn>]
[-Server <ServerIdParameter>]
```

Examples

EXAMPLE 1

This example gets basic configuration and status information for all networks in the DAG DAG1.

```
Get-DatabaseAvailabilityGroupNetwork -Identity DAG1
```

EXAMPLE 2

This example gets complete configuration and status information for all networks in the DAG DAG1.

```
Get-DatabaseAvailabilityGroupNetwork -Identity DAG1 |
Format-List
```

EXAMPLE 3

This example gets complete configuration and status information for the network DAGNetwork02 in the DAG DAG2 from the Mailbox server MBX1.

```
Get-DatabaseAvailabilityGroupNetwork -Identity DAG2
\DAGNetwork02 -Server MBX1 | Format-List
```

Detailed Description

State information is returned for subnets and for network interfaces, as described in the following tables.

Valid states for Internet Protocol version 4 (IPv4) subnets

State	Description
Up	All defined network interfaces in the DAG are functional and available for communication. This is the expected and normal operational state.
Down	All defined network interfaces in the DAG are nonfunctional and have lost communication with each other and all external hosts. All connected network interfaces are in a Failed or Unreachable state.
Partitioned	One or more network interfaces in the DAG are in an Unreachable state, but at least two interfaces can communicate with each other or an external host.
Misconfigured	All subnets for a specified DAG network must have the same values for *ReplicationEnabled* and *IgnoreNetwork*. If any one of the subnets isn't configured with the same values for these parameters as all other subnets on the network, all subnets are in a Misconfigured state.

Unavailable	The network isn't enabled for replication or use by the DAG, or all DAG members attached to the network are inactive or unavailable.
Unknown	The system was unable to determine the state of the subnet.

Valid states for network interfaces

State	Description
Up	The network interface is functional and can communicate with all other network interfaces. This is the expected and normal operational state.
Failed	The network interface is unable to communicate with other network interfaces or external hosts, although other network interfaces on the local area network (LAN) are able to communicate with each other and external hosts.
Unreachable	The system was unable to communicate with at least one network interface whose state is Up.
Unavailable	The network interface isn't enabled for replication or use by the DAG, or the DAG member associated with this network interface is inactive or unavailable.
Unknown	The system was unable to determine the state of the network interface.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupNetworkIdParameter	The *Identity* parameter specifies the name of a DAG or a DAG network.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies health information for the DAG network from a specific Mailbox server in the DAG.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-

DatabaseAvailabilityGroupNetwork

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-DatabaseAvailabilityGroupNetwork** cmdlet to create a database availability group (DAG) network.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-DatabaseAvailabilityGroupNetwork -DatabaseAvailabilityGroup
<DatabaseAvailabilityGroupIdParameter> -Name <String> [-Confirm
[<SwitchParameter>]] [-Description <String>] [-DomainController <Fqdn>] [-
IgnoreNetwork <$true | $false>] [-ReplicationEnabled <$true | $false>] [-
Subnets <DatabaseAvailabilityGroupSubnetId[]>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the DAG network DAG1Repl in the DAG DAG1. A subnet of 10.0.0.0 with a bitmask of 8 is assigned to DAG1Repl, and DAG1Repl is also enabled for continuous replication.

```
New-DatabaseAvailabilityGroupNetwork -
DatabaseAvailabilityGroup DAG1 -Name DAG1Repl -Subnets
10.0.0.0/8 -ReplicationEnabled:$true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
DatabaseAvailabilityGroup	Required	Microsoft.Exchange.Configuration.Tasks.Datab	The *DatabaseAvailabilityGr*

		aseAvailabilityGroupId Parameter	*oup* parameter specifies the name of the DAG that'll use the network being created.
Name	Required	System.String	The *Name* parameter specifies the name of the DAG network being created.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies an optional description of up to 256 characters for the DAG network being created.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this

			configuration change to Active Directory.
IgnoreNetwork	Optional	System.Boolean	The *IgnoreNetwork* parameter excludes the DAG network from use by the DAG.
ReplicationEnabled	Optional	System.Boolean	The *ReplicationEnabled* parameter specifies whether the DAG network being created is enabled for continuous replication.
Subnets	Optional	Microsoft.Exchange.Data.DatabaseAvailabilityGroupSubnetId[]	The *Subnets* parameter specifies the subnets for the DAG network being created.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-DatabaseAvailabilityGroupNetwork

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-DatabaseAvailabilityGroupNetwork** cmdlet to remove a database availability group (DAG) network.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-DatabaseAvailabilityGroupNetwork -Identity
<DatabaseAvailabilityGroupNetworkIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the DAG network DAGNetwork04 from the DAG DAG1.

```
Remove-DatabaseAvailabilityGroupNetwork -Identity DAG1
\DAGNetwork04
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database

availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupNetworkIdParameter	The *Identity* parameter specifies the name of the network being removed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on

			the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-DatabaseAvailabilityGroupNetwork

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-DatabaseAvailabilityGroupNetwork** cmdlet to configure a network for a database availability group (DAG).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-DatabaseAvailabilityGroupNetwork -Identity
<DatabaseAvailabilityGroupNetworkIdParameter> [-Confirm
[<SwitchParameter>]] [-Description <String>] [-DomainController <Fqdn>] [-
IgnoreNetwork <$true | $false>] [-Name <String>] [-ReplicationEnabled
<$true | $false>] [-Subnets <DatabaseAvailabilityGroupSubnetId[]>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the DAG network DAGNetwork01 in the DAG DAG1 for replication.

```
Set-DatabaseAvailabilityGroupNetwork -Identity DAG1
\DAGNetwork01 -ReplicationEnabled:$true
```

EXAMPLE 2

This example disables the DAG network DAGNetwork02 in the DAG DAG2 for replication and configures the DAG to ignore the network.

```
Set-DatabaseAvailabilityGroupNetwork -Identity DAG2
\DAGNetwork02 -ReplicationEnabled:$false -
IgnoreNetwork:$true
```

Detailed Description

You can configure a variety of network properties, such as the name for the network, a description of the network, a list of one or more subnets that comprise the network, and whether the network is enabled for replication (log shipping and seeding).

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupNetworkIdParameter	The *Identity* parameter specifies the DAG network being configured.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies an optional description for the DAG network.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreNetwork	Optional	System.Boolean	The *IgnoreNetwork* parameter indicates that the specified network should be ignored and not used by the DAG.
Name	Optional	System.String	The *Name* parameter provides a name for the DAG network.
ReplicationEnabled	Optional	System.Boolean	The *ReplicationEnabled*

			parameter specifies whether the network can be used for replication activity. If this parameter isn't specified, the default behavior is to enable the network for replication.
Subnets	Optional	Microsoft.Exchange.Data.DatabaseAvailabilityGroupSubnetId[]	The *Subnets* parameter specifies one or more subnets that are associated with the DAG network.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-DatabaseAvailabilityGroupServer

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-07*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Add-DatabaseAvailabilityGroupServer** cmdlet to add a Mailbox server to a database availability group (DAG). A DAG is a set of Mailbox servers that use continuous replication and managed availability to provide automatic database-level recovery from database, server, or network failures.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-DatabaseAvailabilityGroupServer -Identity
<DatabaseAvailabilityGroupIdParameter> -MailboxServer <ServerIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
SkipDagValidation <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds the Mailbox server MBX1 to the DAG DAG1.

```
Add-DatabaseAvailabilityGroupServer -Identity DAG1 -
MailboxServer MBX1
```

Detailed Description

A computer object for a DAG is created in Active Directory when the first server is added to the DAG. This object is used to authenticate servers to each other within a DAG.

To add a Mailbox server to a DAG, the Mailbox server must be running the Windows Server 2008 R2 Enterprise or Datacenter operating system, the Windows Server 2012 Standard or Datacenter operating system, or the Windows Server 2012 R2 operating system, and it must not belong to any other DAG. The Mailbox server must be running the same versions of the Windows operating

system and Microsoft Exchange, and be in the same Active Directory domain as all other Mailbox servers in the DAG. In addition, the Mailbox server must not be configured as an Active Directory domain controller or global catalog server.

To add the first server to a DAG and create a computer object for the DAG, the Exchange Windows Permissions security group must have the appropriate rights to add computer accounts to the domain. Alternatively, a computer account can be created and disabled prior to adding the server. Adding the first server to the DAG enables the computer account for the DAG. Thus, the account used for the task doesn't need permissions to add a computer account to the domain. If you're pre-creating the computer account, the name of the account must match the name for the DAG. For example, if the DAG is named DAG1, the computer account must be named DAG1.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupIdParameter	The *Identity* parameter specifies the name of the DAG to which the server is being added.
MailboxServer	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *MailboxServer* parameter specifies the name of the Mailbox server being added to the DAG.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't

			have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
SkipDagValidation	Optional	System.Management.Automation.SwitchParameter	The *SkipDagValidation* switch specifies whether to bypass the validation of the DAG's quorum model and the health check on the DAG's witness when adding members to the DAG.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-DatabaseAvailabilityGroupServer

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-DatabaseAvailabilityGroupServer** cmdlet to remove a Mailbox server from a database availability group (DAG). To remove a Mailbox server from a DAG, the Mailbox server must not host any replicated databases.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-DatabaseAvailabilityGroupServer -Identity
<DatabaseAvailabilityGroupIdParameter> -MailboxServer <ServerIdParameter>
[-ConfigurationOnly <SwitchParameter>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-SkipDagValidation <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the Mailbox server MBX1 from the DAG DAG3.

```
Remove-DatabaseAvailabilityGroupServer -Identity DAG3 -
```

```
MailboxServer MBX1
```

EXAMPLE 2

This example removes the configuration settings for the Mailbox server MBX4 from the DAG DAG2. MBX4 is currently offline and expected to be offline for an extended period, so its configuration is being removed from the DAG to establish quorum for the DAG or to reduce the number of members needed for quorum by the DAG.

```
Remove-DatabaseAvailabilityGroupServer -Identity DAG2 -
MailboxServer MBX4 -ConfigurationOnly
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupIdParameter	The *Identity* parameter specifies the name of the DAG from which you're removing the server.
MailboxServer	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *MailboxServer* parameter specifies the name of the Mailbox server being removed from the DAG.
ConfigurationOnly	Optional	System.Management.Automation.SwitchParameter	The *ConfigurationOnly* switch should only be used if the Mailbox server has been lost and can no longer be

			contacted, or in situations when the Mailbox server can't be restored to operational service before the messaging service is needed. When used, it removes the Mailbox server from the DAG object in Active Directory. If the Mailbox server is offline but the DAG has quorum, the Mailbox server is evicted from the DAG's cluster and removed from the DAG object in Active Directory.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
SkipDagValidation	Optional	System.Management.Automation.SwitchParameter	The *SkipDagValidation* switch specifies whether to bypass the validation of the DAG's quorum model and the health check on the DAG's witness when removing members from the DAG.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-MailboxDatabaseCopy

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Add-MailboxDatabaseCopy** cmdlet to create a passive copy of an existing active mailbox database.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-MailboxDatabaseCopy -Identity <DatabaseIdParameter> -MailboxServer
<MailboxServerIdParameter> [-ActivationPreference <UInt32>] [-
ConfigurationOnly <SwitchParameter>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-ReplayLagTime <EnhancedTimeSpan>] [-
SeedingPostponed <SwitchParameter>] [-TruncationLagTime
<EnhancedTimeSpan>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds a copy of mailbox database DB1 to the Mailbox server MBX3. Replay lag time and truncation lag time are configured with values of 10 minutes and 15 minutes, respectively. The activation preference is configured with a value of 2.

```
Add-MailboxDatabaseCopy -Identity DB1 -MailboxServer MBX3 -
ReplayLagTime 00:10:00 -TruncationLagTime 00:15:00 -
ActivationPreference 2
```

EXAMPLE 2

This example adds a copy of mailbox database DB2 to the Mailbox server MBX1. Replay lag time and truncation lag time are left at the default values of zero, and the activation preference is configured with a value of 3.

```
Add-MailboxDatabaseCopy -Identity DB2 -MailboxServer MBX1 -
```

```
ActivationPreference 3
```

EXAMPLE 3

This example adds a copy of mailbox database DB3 to the Mailbox server MBX4. Replay lag time and truncation lag time are left at the default values of zero, and the activation preference is configured with a value of 5. In addition, seeding is being postponed for this copy so that it can be seeded using a local source server instead of the current active database copy, which is geographically distant from MBX4.

```
Add-MailboxDatabaseCopy -Identity DB3 -MailboxServer MBX4 -
ActivationPreference 5 -SeedingPostponed
```

Detailed Description

To use the **Add-MailboxDatabaseCopy** cmdlet to add a mailbox database copy, the following criteria must be met:

- The specified Mailbox server must be in the same database availability group (DAG), and the DAG must have quorum and be healthy.
- The specified Mailbox server must not already host a copy of the specified mailbox database.
- The database path used by the specified database must also be available on the specified Mailbox server, because all copies of a database must use the same path.
- If you're adding the second copy of a database (for example, adding the first passive copy of the database), circular logging must not be enabled for the specified mailbox database. If circular logging is enabled, you must first disable it. After the mailbox database copy has been added, circular logging can be enabled. After enabling circular logging for a replicated mailbox database, continuous replication circular logging (CRCL) is used instead of JET circular logging. If you're adding the third or subsequent copy of a database, CRCL can remain enabled.

After running the **Add-MailboxDatabaseCopy** cmdlet, the new copy remains in a Suspended state if the *SeedingPostponed* parameter is specified. When the database copy status is set to Suspended, the *SuspendMessage* is set to "Replication is suspended for database copy '{0}' because database needs to be seeded."

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.DatabaseIdParameter	specifies the name of the mailbox database being copied. Database names must be unique within the Exchange organization.
MailboxServer	Required	Microsoft.Exchange.Configuration.Tasks.MailboxServerIdParameter	The *MailboxServer* parameter specifies the name of the server that will host the database copy. This server must be a member of the same DAG and must not already host a copy of the database.
ActivationPreference	Optional	System.UInt32	The *ActivationPreference* parameter value is used as part of Active Manager's best copy selection process and to redistribute active mailbox databases throughout the DAG when using the RedistributeActiveDatabases.ps1 script. The value for the activation preference is a number equal to or greater than 1, where 1 is at the top of the preference order.

			The preference number can't be larger than the number of copies of the mailbox database.
ConfigurationOnly	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ReplayLagTime	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ReplayLagTime* parameter specifies the amount of time that the Microsoft Exchange Replication service

			waits before replaying log files that have been copied to the database copy. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The maximum allowable setting for this value is 14 days. The minimum allowable setting is 0 seconds, and setting this value to 0 seconds eliminates any delay in log replay activity. For example, to specify a 14-day replay lag period, enter 14.00:00:00. The default value is 00.00:00:00, which specifies that there's no replay lag.
SeedingPostponed	Optional	System.Management.Automation.SwitchParameter	The *SeedingPostponed* parameter specifies that the task doesn't seed the database copy. You must then

			explicitly seed the database copy.
TruncationLagTime	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *TruncationLagTime* parameter specifies the amount of time that the Microsoft Exchange Replication service waits before truncating log files that have replayed into a copy of the database. The time period begins after the log has been successfully replayed into the copy of the database. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The maximum allowable setting for this value is 14 days. The minimum allowable setting is 0 seconds, and setting this value to 0 seconds eliminates any delay in log truncation activity. For example, to specify

			a 14-day truncation lag period, enter 14.00:00:00. The default value is 00.00:00:00, which specifies that there's no truncation lag.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailboxDatabaseCopy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-MailboxDatabaseCopy** cmdlet to remove a passive copy of a mailbox database.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MailboxDatabaseCopy -Identity <DatabaseCopyIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes a copy of mailbox database DB1 from the Mailbox server MBX3.

```
Remove-MailboxDatabaseCopy -Identity DB1\MBX3
```

Detailed Description

To use the **Remove-MailboxDatabaseCopy** cmdlet to remove a mailbox database copy, the following criteria must be met:

- The database availability group (DAG) hosting the mailbox database must have quorum and all cluster and network functions must be healthy.
- If you're removing the last passive copy of the database, continuous replication circular logging (CRCL) must not be enabled for the specified mailbox database. If CRCL is enabled, you must first disable it. After the mailbox database copy has been removed, circular logging can be enabled. After enabling circular logging for a non-replicated mailbox database, JET circular logging is used instead of CRCL. If you aren't removing the last passive copy of a database, CRCL can remain enabled.

You can't use this cmdlet to remove the active copy of a mailbox database. To remove the active copy of a mailbox database, you must first remove all passive copies of the database, and then use the **Remove-MailboxDatabase** cmdlet to remove the active copy.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Note:

Running this cmdlet removes the mailbox database copy configuration, but doesn't delete the database copy's files. If necessary, you can manually delete those files.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseCopyIdParameter	The *Identity* parameter specifies the name of the mailbox database whose copy is being removed. When using this parameter, specify a format of *DatabaseName**ServerName*.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.A	The *WhatIf* switch

		utomation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Resume-MailboxDatabaseCopy

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Resume-MailboxDatabaseCopy** cmdlet to unblock activation or resume log copying and replay for a passive mailbox database copy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Resume-MailboxDatabaseCopy -Identity <DatabaseCopyIdParameter> [-
ReplicationOnly <SwitchParameter>] <COMMON PARAMETERS>
```

```
Resume-MailboxDatabaseCopy -DisableReplayLag <SwitchParameter> -Identity
<DatabaseCopyIdParameter> [-DisableReplayLagReason <String>] <COMMON
PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resumes replication and replay activity for the copy of the database DB1 hosted on the Mailbox server MBX3.

```
Resume-MailboxDatabaseCopy -Identity DB1\MBX3
```

EXAMPLE 2

This example resumes replication and replay activity for the copy of the database DB2 hosted on the Mailbox server MBX4. After the copy is resumed, it remains administratively blocked for activation.

```
Resume-MailboxDatabaseCopy -Identity DB2\MBX4 -
ReplicationOnly
```

Detailed Description

The **Resume-MailboxDatabaseCopy** cmdlet resumes replication and replay from a suspended state. If a database copy was suspended without administrator intervention, it's because the database copy is in a bad state. You can use the **Get-MailboxDatabaseCopyStatus** cmdlet to see if there are any messages indicating a failure. If the copy of the database is in a bad state, resuming the copy causes replication to fail and the mailbox database copy to return to a suspended state.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
DisableReplayLag	Required	System.Management.Automation.SwitchPara	The *DisableReplayLag* parameter specifies

		meter	that any configured replay lag time for the database copy should be disabled when the passive copy is resumed.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseCopyIdParameter	The *Identity* parameter specifies the name of the database whose copying is being resumed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisableReplayLagReason	Optional	System.String	The *DisableReplayLagReason* parameter is used with the *DisableReplayLag* parameter to specify an administrative reason for disabling replay lag time for a passive copy.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ReplicationOnly	Optional	System.Management.Automation.SwitchParameter	The *ReplicationOnly* switch specifies whether to resume replication without affecting the activation setting (for example, the *ActivationSuspended* property for the database copy remains set to True).
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxDatabaseCopy

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-MailboxDatabaseCopy** cmdlet to configure the properties of a database copy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxDatabaseCopy -ClearHostServer <SwitchParameter> -Identity
<DatabaseCopyIdParameter> <COMMON PARAMETERS>


Set-MailboxDatabaseCopy -Identity <DatabaseCopyIdParameter> [-
ActivationPreference <UInt32>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-
DatabaseCopyAutoActivationPolicy <Unrestricted | IntrasiteOnly | Blocked>]
[-DomainController <Fqdn>] [-ReplayLagTime <EnhancedTimeSpan>] [-
TruncationLagTime <EnhancedTimeSpan>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures the replay lag time with a value of 3 days for a copy of the database DB2 hosted on the Mailbox server MBX1.

```
Set-MailboxDatabaseCopy -Identity DB2\MBX1 -ReplayLagTime
3.0:0:0
```

EXAMPLE 2

This example configures an activation preference of 3 for the copy of the database DB1 hosted on the Mailbox server MBX2.

```
Set-MailboxDatabaseCopy -Identity DB1\MBX2 -
ActivationPreference 3
```

Detailed Description

With this cmdlet, you can configure the replay lag time, truncation lag time, and activation preference value for a mailbox database copy.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
ClearHostServer	Required	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseCopyIdParameter	The *Identity* parameter specifies the name of the database whose copy is being modified.
ActivationPreference	Optional	System.UInt32	The *ActivationPreference* parameter value is used as part of Active Manager's best copy selection process and to redistribute active mailbox databases throughout the database availability

			group (DAG) when using the RedistributeActiveDatabases.ps1 script. The value for the *ActivationPreference* parameter is a number equal to or greater than 1, where 1 is at the top of the preference order. The position number can't be larger than the number of database copies of the mailbox database.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DatabaseCopyAutoActivationPolicy	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DatabaseCopyAutoActivationPolicyType	This parameter is reserved for internal Microsoft use.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ReplayLagTime	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ReplayLagTime* parameter specifies the amount of time that the Microsoft Exchange Replication service should wait before replaying log files that have been copied to the passive database copy. Setting this parameter to a value greater than 0 creates a *lagged* database copy. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The maximum allowable setting for this value is 14 days. The minimum

			allowable setting is 0 seconds, and setting this value to 0 seconds eliminates any delay in log replay activity. For example, to specify a 14-day replay lag period, enter 14.00:00:00. The default value is 00.00:00:00, which specifies that there's no replay lag.
TruncationLagTime	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *TruncationLagTime* parameter specifies the amount of time that the Microsoft Exchange Replication service should wait before truncating log files that have replayed into the passive copy of the database. The time period begins after the log has been successfully replayed into the copy of the database. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m =

			minutes, and s = seconds. The maximum allowable setting for this value is 14 days. The minimum allowable setting is 0 seconds, and setting this value to 0 seconds eliminates any delay in log truncation activity. For example, to specify a 14-day truncation lag period, enter 14.00:00:00. The default value is 00.00:00:00, which specifies that there's no truncation lag.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Suspend-MailboxDatabaseCopy

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Suspend-MailboxDatabaseCopy** cmdlet to block replication and replay activities (log copying and replay) or activation for a database configured with two or more database copies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Suspend-MailboxDatabaseCopy -Identity <DatabaseCopyIdParameter> [-
ActivationOnly <SwitchParameter>] [-SuspendComment <String>] <COMMON
PARAMETERS>


Suspend-MailboxDatabaseCopy -EnableReplayLag <SwitchParameter> -Identity
<DatabaseCopyIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example suspends replication and replay activity for the copy of the database DB1 hosted on the Mailbox server MBX3. An optional administrative reason for the suspension is specified.

```
Suspend-MailboxDatabaseCopy -Identity DB1\MBX3 -
SuspendComment "Maintenance on MBX3"
```

EXAMPLE 2

This example only suspends activation for the copy of the database DB3 hosted on the Mailbox server MBX2.

```
Suspend-MailboxDatabaseCopy -Identity DB3\MBX2 -
ActivationOnly
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
EnableReplayLag	Required	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseCopyIdParameter	The *Identity* parameter specifies the name of the database copy being suspended.
ActivationOnly	Optional	System.Management.Automation.SwitchParameter	The *ActivationOnly* switch specifies whether to suspend only activation for the mailbox database copy.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run.

			To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies the reason that the database copy is being suspended. This parameter is limited to 512 characters.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-MailboxDatabaseCopy

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Update-MailboxDatabaseCopy** cmdlet to seed or reseed a mailbox database copy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-MailboxDatabaseCopy -Identity <DatabaseCopyIdParameter> [-BeginSeed
<SwitchParameter>] [-CatalogOnly <SwitchParameter>] [-DatabaseOnly
<SwitchParameter>] [-DeleteExistingFiles <SwitchParameter>] [-Force
<SwitchParameter>] [-ManualResume <SwitchParameter>] [-Network
<DatabaseAvailabilityGroupNetworkIdParameter>] [-
NetworkCompressionOverride <UseDagDefault | Off | On>] [-
NetworkEncryptionOverride <UseDagDefault | Off | On>] [-
SafeDeleteExistingFiles <SwitchParameter>] [-SourceServer
<ServerIdParameter>] <COMMON PARAMETERS>


Update-MailboxDatabaseCopy -CancelSeed <SwitchParameter> -Identity
<DatabaseCopyIdParameter> <COMMON PARAMETERS>


Update-MailboxDatabaseCopy -Server <MailboxServerIdParameter> [-
CatalogOnly <SwitchParameter>] [-DatabaseOnly <SwitchParameter>] [-
DeleteExistingFiles <SwitchParameter>] [-ManualResume <SwitchParameter>]
[-MaximumSeedsInParallel <Int32>] [-NetworkCompressionOverride
<UseDagDefault | Off | On>] [-NetworkEncryptionOverride <UseDagDefault |
Off | On>] [-SafeDeleteExistingFiles <SwitchParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
```

```
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example seeds a copy of the database DB1 on the Mailbox server MBX1.

```
Update-MailboxDatabaseCopy -Identity DB1\MBX1
```

EXAMPLE 2

This example seeds a copy of the database DB1 on the Mailbox server MBX1 using MBX2 as the source Mailbox server for the seed.

```
Update-MailboxDatabaseCopy -Identity DB1\MBX1 -SourceServer
MBX2
```

EXAMPLE 3

This example seeds a copy of the database DB1 on the Mailbox server MBX1 without seeding the content index catalog.

```
Update-MailboxDatabaseCopy -Identity DB1\MBX1 -DatabaseOnly
```

EXAMPLE 4

This example seeds the content index catalog for the copy of the database DB1 on the Mailbox server MBX1 without seeding the database file.

```
Update-MailboxDatabaseCopy -Identity DB1\MBX1 -CatalogOnly
```

EXAMPLE 5

This example performs a full server reseed of all of the databases on the Mailbox server MBX1.

```
Update-MailboxDatabaseCopy -Server MBX1
```

Detailed Description

Seeding is the process in which a copy of a mailbox database is added to another Mailbox server. This becomes the database copy into which copied log files and data are replayed.

The **Update-MailboxDatabaseCopy** cmdlet can also be used to seed a content index catalog for a mailbox database copy.

You must suspend a database copy before you can update it using the **Update-**

MailboxDatabaseCopy cmdlet. For detailed steps about how to suspend a database copy, see Suspend or resume a mailbox database copy.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
CancelSeed	Required	System.Management.Automation.SwitchParameter	The *CancelSeed* switch specifies whether to cancel an in-progress seeding operation.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseCopyIdParameter	The *Identity* parameter specifies the name or GUID of the mailbox database whose copy is being seeded.
Server	Required	Microsoft.Exchange.Configuration.Tasks.MailboxServerIdParameter	The *Server* parameter is used as part of a full server reseed operation. It can be used with the *MaximumSeedsInParallel* parameter to start reseeds of database copies in parallel across the specified server in batches of up to the value of the *MaximumSeedsInParallel* parameter copies at

			a time.
BeginSeed	Optional	System.Management.Automation.SwitchParameter	The *BeginSeed* parameter is useful for scripting reseeds, because with this parameter, the task asynchronously starts the seeding operation and then exits the cmdlet.
CatalogOnly	Optional	System.Management.Automation.SwitchParameter	The *CatalogOnly* parameter specifies that only the content index catalog for the database copy should be seeded.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DatabaseOnly	Optional	System.Management.Automation.SwitchParameter	The *DatabaseOnly* parameter specifies that only the database copy should be seeded.

			The content index catalog isn't seeded.
DeleteExistingFiles	Optional	System.Management.Automation.SwitchParameter	The *DeleteExistingFiles* switch specifies whether to remove the existing log files at the target location. This parameter removes only the files that it checks for and fails if other files are present. No action is taken on other files at the target location. Therefore, if database-related files are present, you must manually remove them.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run

			programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
ManualResume	Optional	System.Management.Automation.SwitchParameter	The *ManualResume* switch specifies whether to automatically resume replication on the database copy. With this parameter, you can manually resume replication to the database copy.
MaximumSeedsInParallel	Optional	System.Int32	The *MaximumSeedsInParallel* parameter is used with the *Server* parameter to specify the maximum number of parallel seeding operations that should occur across the specified server during

			a full server reseed operation. The default value is 10.
Network	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseAvailabilityGroupNetworkIdParameter	The *Network* parameter specifies which DAG network should be used for seeding.
NetworkCompressionOverride	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.UseDagDefaultOnOff	The *NetworkCompressionOverride* parameter specifies whether to override the current DAG network compression settings.
NetworkEncryptionOverride	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.UseDagDefaultOnOff	The *NetworkEncryptionOverride* parameter specifies whether to override the current DAG encryption settings.
SafeDeleteExistingFiles	Optional	System.Management.Automation.SwitchParameter	The *SafeDeleteExistingFiles* parameter is used to perform a seeding operation with a single copy redundancy pre-check prior to the seed. Because this parameter includes the

			redundancy safety check, it requires a lower level of permissions than the *DeleteExistingFiles* parameter, enabling a limited permission administrator to perform the seeding operation.
SourceServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *SourceServer* parameter specifies the name of a Mailbox server with a passive copy of the mailbox database to be used as the source for the seed operation.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxDatabaseCopyStatus

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxDatabaseCopyStatus** cmdlet to view health and status information about one or more mailbox database copies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxDatabaseCopyStatus -Server <MailboxServerIdParameter> <COMMON
PARAMETERS>


Get-MailboxDatabaseCopyStatus [-Identity <DatabaseCopyIdParameter>] [-
Local <SwitchParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Active <SwitchParameter>] [-ConnectionStatus
<SwitchParameter>] [-DomainController <Fqdn>] [-ExtendedErrorInfo
<SwitchParameter>] [-UseServerCache <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns status information for all copies of the database DB1. The status results are displayed in a list format.

```
Get-MailboxDatabaseCopyStatus -Identity DB1 | Format-List
```

EXAMPLE 2

This example returns the status for all database copies on the Mailbox server MBX1. The status results are also displayed in a list format.

```
Get-MailboxDatabaseCopyStatus -Server MBX1 | Format-List
```

EXAMPLE 3

This example returns the status for the copy of database DB1 on the Mailbox server MBX2. The status results are also displayed in a list format.

```
Get-MailboxDatabaseCopyStatus -Identity DB1\MBX2 | Format-List
```

Detailed Description

If a database is specified by using the *Identity* parameter with the command, the status of all copies of the database is returned. If a server is specified by using the *Server* parameter with the command, information about all database copies on the server is returned. If neither parameter is specified with the command, information about all database copies in the organization is returned.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.MailboxServerIdParameter	The *Server* parameter specifies that a Mailbox server returns status information for all of its mailbox database copies. This parameter can't be combined with the *Identity* parameter.
Active	Optional	System.Management.Automation.SwitchParameter	The *Active* switch specifies whether to return mailbox

			database copy status for the active mailbox database copy only.
ConnectionStatus	Optional	System.Management.Automation.SwitchParameter	The *ConnectionStatus* switch is obsolete and in the process of being deprecated. Use of this switch will be ignored by the task. The information previously provided by this switch is now provided through an internal caching mechanism and, as such, the switch is no longer needed.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
ExtendedErrorInfo	Optional	System.Management.Automation.SwitchParameter	The *ExtendedErrorInfo* switch specifies whether to return an output object containing any exception details.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.Datab	The *Identity* parameter specifies the name of

		aseCopyIdParameter	the database copy for which the command should gather information. The Identity parameter can be specified in the form of <*database*> \<*server*>. Specifying just <*database*> returns information for all copies of the database. This parameter can't be combined with the *Server* parameter.
Local	Optional	System.Management.Automation.SwitchParameter	The *Local* switch specifies whether to return mailbox database copy status information from only the local Mailbox server.
UseServerCache	Optional	System.Management.Automation.SwitchParameter	The *UseServerCache* switch specifies whether to enable a server-side remote procedure call (RPC) caching of status information for 5 seconds.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-ReplicationHealth

Exchange Management Shell > Exchange 2013 cmdlets > High availability cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-ReplicationHealth** cmdlet to check all aspects of replication and replay, or to provide status for a specific Mailbox server in a database availability group (DAG).

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-ReplicationHealth [-Identity <ServerIdParameter>] [-
ActiveDirectoryTimeout <Int32>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-MonitoringContext <$true | $false>] [-
OutputObjects <SwitchParameter>] [-TransientEventSuppressionWindow
<UInt32>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the health of replication for the Mailbox server MBX1.

```
Test-ReplicationHealth -Identity MBX1
```

Detailed Description

The **Test-ReplicationHealth** cmdlet is designed for the proactive monitoring of continuous replication and the continuous replication pipeline, the availability of Active Manager, and the health and status of the underlying cluster service, quorum, and network components. The **Test-ReplicationHealth** cmdlet can be run locally or remotely against any Mailbox server in a DAG.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Database availability groups" entry in the High availability and site resilience permissions topic.

Parameters

Parameter	Required	Type	Description
ActiveDirectoryTimeout	Optional	System.Int32	The *ActiveDirectoryTimeout* parameter specifies the amount of time, in seconds, allowed for each directory service operation to complete before the operation times out. The default value is 15 seconds.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the name of the Mailbox server that you want to test.
MonitoringContext	Optional	System.Boolean	The *MonitoringContext* parameter includes or excludes the associated monitoring events and performance counters in the results. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you specify the value `$true`, the monitoring events and performance counters are included in the command results. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.

OutputObjects	Optional	System.Management.Automation.SwitchParameter	The *OutputObjects* switch specifies whether to output an array of information regarding failures.
TransientEventSuppressionWindow	Optional	System.UInt32	The *TransientEventSuppressionWindow* parameter specifies the number of minutes that the queue lengths can be exceeded before the queue length tests are considered to have failed. This parameter is used to reduce the number of failures due to transient load generation.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Mail flow cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-08*

Mail flow configuration cmdlets

Get-AdSite

Set-AdSite

Get-AdSiteLink

Set-AdSiteLink

Get-FrontendTransportService

Set-FrontendTransportService

Get-MailboxTransportService

Set-MailboxTransportService

Get-TransportConfig

Set-TransportConfig

Get-TransportServer

Set-TransportServer

Note:
The ***-TransportServer** cmdlets will be removed in a future version of Exchange. You should use the ***-TransportService** cmdlets instead.

Get-TransportService

Set-TransportService

Mail flow troubleshooting and reporting cmdlets

Get-AgentLog

Test-Mailflow

Get-MessageTrackingLog

Get-MessageTrackingReport

Search-MessageTrackingReport

Get-NetworkConnectionInfo

Test-SmtpConnectivity

Get-TransportPipeline

Address rewriting cmdlets on Edge Transport servers

Get-AddressRewriteEntry

New-AddressRewriteEntry

Remove-AddressRewriteEntry

Set-AddressRewriteEntry

Connector cmdlets

Receive connector cmdlets

Get-ReceiveConnector

New-ReceiveConnector

Remove-ReceiveConnector

Set-ReceiveConnector

Send connector cmdlets

Get-SendConnector

New-SendConnector

Remove-SendConnector

Set-SendConnector

Delivery agent connector cmdlets

Get-DeliveryAgentConnector

New-DeliveryAgentConnector

Remove-DeliveryAgentConnector

Set-DeliveryAgentConnector

Foreign connector cmdlets

Get-ForeignConnector

New-ForeignConnector

Remove-ForeignConnector

Set-ForeignConnector

Delivery status notification (DSN) cmdlets

Get-SystemMessage

New-SystemMessage

Remove-SystemMessage

Set-SystemMessage

Domain management cmdlets

Accepted domain cmdlets

Get-AcceptedDomain

New-AcceptedDomain

Remove-AcceptedDomain

Set-AcceptedDomain

Remote domain cmdlets

Get-RemoteDomain

New-RemoteDomain

Remove-RemoteDomain

Set-RemoteDomain

X.400 authoritative domain cmdlets

Get-X400AuthoritativeDomain

New-X400AuthoritativeDomain

Remove-X400AuthoritativeDomain

Set-X400AuthoritativeDomain

Edge subscription and EdgeSync cmdlets

Get-EdgeSubscription

New-EdgeSubscription

Remove-EdgeSubscription

Start-EdgeSynchronization

Test-EdgeSynchronization

Get-EdgeSyncServiceConfig

New-EdgeSyncServiceConfig

Set-EdgeSyncServiceConfig

Queue and message cmdlets

Export-Message

Get-Message

Redirect-Message

Remove-Message

Resume-Message

Suspend-Message

Get-Queue

Resume-Queue

Retry-Queue

Suspend-Queue

Add-ResubmitRequest

Get-ResubmitRequest

Remove-ResubmitRequest

Set-ResubmitRequest

Get-QueueDigest

Transport agent cmdlets

Disable-TransportAgent

Enable-TransportAgent

Get-TransportAgent

Install-TransportAgent

Set-TransportAgent

Uninstall-TransportAgent

Get-AcceptedDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-AcceptedDomain** cmdlet to view the configuration information for the accepted domains in your organization.

```
Get-AcceptedDomain [-Identity <AcceptedDomainIdParameter>] [-
AccountPartition <AccountPartitionIdParameter>] [-DomainController <Fqdn>]
[-Filter <String>] [-Organization <OrganizationIdParameter>] [-
UsnForReconciliationSearch <Int64>]
```

Examples

EXAMPLE 1

This example lists all the accepted domains in your organization.

```
Get-AcceptedDomain
```

EXAMPLE 2

This example lists all the authoritative accepted domains in your organization.

```
Get-AcceptedDomain | Where{$_.DomainType -eq
'Authoritative'}
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Accepted domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An

			Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Filter	Optional	System.String	The *Filter* parameter specifies a set of attributes used to filter the list of accepted domains.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.AcceptedDomainIdParameter	The *Identity* parameter specifies a string value for the accepted domain. Enter either the GUID or the name of the accepted domain.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
UsnForReconciliationSearch	Optional	System.Int64	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-AcceptedDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-AcceptedDomain** cmdlet to create an accepted domain in your organization. An *accepted domain* is any SMTP namespace for which an Exchange organization sends and receives email.

```
New-AcceptedDomain -DomainName <SmtpDomainWithSubdomains> -Name <String>
[-AuthenticationType <Managed | Federated>] [-CatchAllRecipient
<RecipientIdParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-DomainType <Authoritative | ExternalRelay | InternalRelay>] [-
InitialDomain <$true | $false>] [-LiveIdInstanceType <Consumer |
Business>] [-MailFlowPartner <MailFlowPartnerIdParameter>] [-
MatchSubDomains <$true | $false>] [-Organization
<OrganizationIdParameter>] [-OutboundOnly <$true | $false>] [-
SkipDnsProvisioning <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the new authoritative accepted domain Contoso.

```
New-AcceptedDomain -DomainName Contoso.com -DomainType
Authoritative -Name Contoso
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Accepted domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description

DomainName	Required	Microsoft.Exchange.Data.SmtpDomainWithSubdomains	The *DomainName* parameter specifies the SMTP domain that you want to establish as an accepted domain. Valid input for the *DomainName* parameter is an SMTP domain. You can use a wildcard character to specify all subdomains of a specified domain, as shown in the following example: `*.contoso.com`. However, you can't embed a wildcard character, as shown in the following example: `domain.*.contoso.com`. The domain name string may not contain more than 256 characters.
Name	Required	System.String	The *Name* parameter specifies a unique name for the accepted domain object.
AuthenticationType	Optional	Microsoft.Exchange.Data.Directory.AuthenticationType	This parameter is reserved for internal Microsoft use.

CatchAllRecipient	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

DomainType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AcceptedDomainType	The *DomainType* parameter specifies the type of accepted domain that you want to configure. Valid values are `Authoritative`, `InternalRelay`, or `ExternalRelay`. You must set at least one value. In an authoritative domain, messages are delivered to a recipient that has a domain account in your Exchange organization. In an internal relay domain, messages are relayed to a server outside your Exchange organization, but still under the authority of your company or IT department. Use the internal relay domain if you want to treat the messages to this domain as internal messages. In an external relay domain, messages are relayed to an email server,

			outside your organization, which you don't control. The default value is `Authoritative`.
InitialDomain	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LiveIdInstanceType	Optional	Microsoft.Exchange.Data.Directory.LiveIdInstanceType	This parameter is reserved for internal Microsoft use.
MailFlowPartner	Optional	Microsoft.Exchange.Configuration.Tasks.MailFlowPartnerIdParameter	This parameter is reserved for internal Microsoft use.
MatchSubDomains	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OutboundOnly	Optional	System.Boolean	The *OutboundOnly* parameter specifies whether this accepted domain is an internal relay domain for the on-premises deployment for organizations that have coexistence with a

			cloud-based organization. The authoritative accepted domain for the on-premises deployment is configured as an internal relay accepted domain on the cloud side. If the on-premises deployment is using Microsoft Forefront Online Protection for Exchange, you must set this parameter to `$true` for the accepted domain that represents your on-premises deployment. This parameter is used only if the *DomainType* parameter is set to `Authoritative` or `InternalRelay`. The default value is `$false`.
SkipDnsProvisioning	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on

			the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-AcceptedDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-AcceptedDomain** cmdlet to remove an accepted domain. When you remove an accepted domain, the accepted domain object is deleted.

```
Remove-AcceptedDomain -Identity <AcceptedDomainIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the accepted domain Contoso.

```
Remove-AcceptedDomain Contoso
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Accepted domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AcceptedDomainIdParameter	The *Identity* parameter specifies the accepted domain you want to remove. Enter either the GUID or the name of the remote domain.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AcceptedDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-07-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-AcceptedDomain** cmdlet to configure an existing accepted domain in your organization. An *accepted domain* is any SMTP namespace for which an Exchange organization sends and receives email.

```
Set-AcceptedDomain -Identity <AcceptedDomainIdParameter> [-
AddressBookEnabled <$true | $false>] [-AuthenticationType <Managed |
Federated>] [-CatchAllRecipient <RecipientIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-DomainType
<Authoritative | ExternalRelay | InternalRelay>] [-DualProvisioningEnabled
<$true | $false>] [-EnableNego2Authentication <$true | $false>] [-
InitialDomain <$true | $false>] [-IsCoexistenceDomain <$true | $false>] [-
LiveIdInstanceType <Consumer | Business>] [-MailFlowPartner
<MailFlowPartnerIdParameter>] [-MakeDefault <$true | $false>] [-
MatchSubDomains <$true | $false>] [-Name <String>] [-OutboundOnly <$true |
$false>] [-PendingCompletion <$true | $false>] [-PendingRemoval <$true |
$false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example makes the accepted domain Contoso the default accepted domain.

```
Set-AcceptedDomain -Identity Contoso -MakeDefault $true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Accepted domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AcceptedDomainIdParameter	The *Identity* parameter specifies the accepted domain you want to modify. You can use any value that uniquely identifies the accepted domain object. For example, you can use the name, GUID or distinguished name (DN) of the accepted domain.
AddressBookEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *AddressBookEnabled* parameter specifies whether to enable recipient filtering on the server that accepts mail for this accepted domain. The default values for this parameter are as follows: • For authoritative domains `$true` • For internal relay domains `$false` • For external relay domains `$false`
AuthenticationType	Optional	Microsoft.Exchange.Da	This parameter is reserved

		ta.Directory.AuthenticationType	for internal Microsoft use.
CatchAllRecipient	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the

			local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
DomainType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AcceptedDomainType	The *DomainType* parameter specifies the type of accepted domain that you want to configure. Valid values are `Authoritative`, `InternalRelay`, and `ExternalRelay`. You must set at least one value. In an authoritative domain, messages are delivered to a recipient that has a domain account in your Exchange organization. In an internal relay domain, messages are relayed to a server outside your Exchange organization, but still under the authority of your company or IT department. Use the internal relay domain if you want to treat messages to this domain as internal messages. In

			an external relay domain, messages are relayed to an email server outside your organization, which you don't control. The default value is `Authoritative`. `ExternalRelay` is only available in on-premises Exchange 2013.
DualProvisioningEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
EnableNego2Authentication	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
InitialDomain	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
IsCoexistenceDomain	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LiveIdInstanceType	Optional	Microsoft.Exchange.Data.Directory.LiveIdInstanceType	This parameter is reserved for internal Microsoft use.
MailFlowPartner	Optional	Microsoft.Exchange.Configuration.Tasks.MailFlowPartnerIdParameter	This parameter is reserved for internal Microsoft use.
MakeDefault	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *MakeDefault*

			parameter specifies whether the accepted domain is the default domain. The default accepted domain is the domain name associated with outbound messages that have encapsulated addresses, such as IMCEANOTES-user +40OtherSystem@contoso.com, for non-Exchange email system interoperability. If you don't interoperate with a non-Exchange email system in your organization, you don't have to set this parameter. For the first accepted domain created in the organization, the default value is `$true`. For subsequent accepted domains, the default value is `$false`.
MatchSubDomains	Optional	System.Boolean	The *MatchSubDomains* parameter enables mail to be sent by and received from users on any subdomain of this accepted domain. The

			default value is `$false`.
Name	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *Name* parameter specifies a unique name for the accepted domain object.
OutboundOnly	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *OutboundOnly* parameter specifies whether this accepted domain is an internal relay domain for the on-premises deployment for organizations that have coexistence with a cloud-based organization. The authoritative accepted domain for the on-premises deployment is configured as an internal relay accepted domain on the cloud side. If the on-premises deployment is using Microsoft Forefront Online Protection for Exchange, you must set this parameter to `$true`

			for the accepted domain that represents your on-premises deployment. This parameter is used only if the *DomainType* parameter is set to `Authoritative` or `InternalRelay`. The default value is `$false`.
PendingCompletion	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
PendingRemoval	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AddressRewriteEntry

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Get-AddressRewriteEntry** cmdlet to view an existing address rewrite entry that rewrites sender and recipient email addresses in messages sent to or sent from your organization through an Edge Transport server.

```
Get-Addressrewriteentry [-Identity <AddressRewriteEntryIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns a summary listing of all address rewrite entries.

```
Get-AddressRewriteEntry
```

EXAMPLE 2

This example returns the detailed configuration of a single address rewrite entry by piping the results to the **Format-List** command.

```
Get-AddressRewriteEntry "Address rewrite entry for
contoso.com" | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address Rewriting - Edge Transport" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Management.MessagingPolicies.AddressRewrite.AddressRewriteEntryIdParameter	The *Identity* parameter specifies the address rewrite entry to be retrieved. The *Identity* parameter accepts a GUID or the unique address rewrite name. You can omit the *Identity* parameter label.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-AddressRewriteEntry

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **New-AddressRewriteEntry** cmdlet to create an address rewrite entry that rewrites sender and recipient email addresses in messages sent to or sent from your organization through an Edge Transport server.

```
New-AddressRewriteEntry -ExternalAddress <String> -InternalAddress
<String> -Name <String> [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-ExceptionList <MultiValuedProperty>] [-OutboundOnly <$true |
$false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates an address rewrite entry that rewrites the email address david@contoso.com to david@northwindtraders.com in outbound mail. Because the *OutboundOnly* parameter is not set to `$true`, inbound mail sent to david@northwindtraders.com is rewritten back to david@contoso.com.

```
New-AddressRewriteEntry -Name "Address rewrite entry for
david@contoso.com" -InternalAddress david@contoso.com -
ExternalAddress david@northwindtraders.com
```

EXAMPLE 2

This example creates an address rewrite entry that rewrites all email addresses in the contoso.com domain to northwindtraders.com in outbound mail. Because the *OutboundOnly* parameter is not set

to $true, inbound mail sent to northwindtraders.com recipients is rewritten back to contoso.com.

```
New-AddressRewriteEntry -Name "Address rewrite entry for
all contoso.com email addresses" -InternalAddress
contoso.com -ExternalAddress northwindtraders.com
```

EXAMPLE 3

This example creates an address rewrite entry that rewrites all email addresses in the contoso.com domain and all subdomains to northwindtraders.com. However, email addresses in research.contoso.com and corp.contoso.com are not rewritten. Because this address rewrite entry affects a domain and all subdomains (*.contoso.com), address rewriting occurs on outbound mail only.

```
New-AddressRewriteEntry -Name "Address rewrite entry for
contoso.com and all subdomain email addresses" -
InternalAddress *.contoso.com -ExternalAddress
northwindtraders.com -ExceptionList
research.contoso.com,corp.contoso.com -OutboundOnly $true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address Rewriting - Edge Transport" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
ExternalAddress	Required	System.String	The *ExternalAddress* parameter specifies the final email addresses that you want. If the *InternalAddress* parameter specifies a single email address (chris@contoso.com), the *ExternalAddress*

			parameter must also specify a single email address (support@contoso.com). If the *InternalAddress* parameter specifies a single domain (contoso.com) or a domain and all subdomains (*.contoso.com), the *ExternalAddress* parameter must specify a single domain (fabrikam.com). **Note:** You can't use the wildcard character (*) with the *ExternalAddress* parameter.
InternalAddress	Required	System.String	The *InternalAddress* parameter specifies the original email addresses that you want to change. You can use the following values: • **Single email address** david@contoso.com • **Single domain** contoso.com or sales.contoso.com • **Domain and all subdomains** *.contoso.com

Name	Required	System.String	The *Name* parameter specifies a unique name for this address rewrite entry.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

ExceptionList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptionList* parameter specifies the email address domains that shouldn't be rewritten when the *InternalAddress* parameter contains the wildcard character to rewrite addresses in a domain and all subdomains (`*.contoso.com`). You can enter multiple domain values in the *ExceptionList* parameter separated by commas.
OutboundOnly	Optional	System.Boolean	The *OutboundOnly* parameter enables or disables outbound-only address rewriting. Valid input for this parameter is `$true` or `$false`. The value `$true` means address rewriting occurs in outbound mail only. The value `$false` means address rewriting occurs on outbound mail and also on inbound mail (rewritten email addresses are changed back to the original email addresses in inbound mail). The

			default value is `$false`. **Note:** You must set this parameter to `$true` if the *InternalAddress* parameter contains the wildcard character to rewrite addresses in a domain and all subdomains (*.contoso.com). Also, when you configure outbound-only address rewriting, you need to configure the rewritten email address as a proxy address on the affected recipients. For example, if laura@sales.contoso.com is rewritten to laura@contoso.com, the proxy address laura@contoso.com must be configured on Laura's mailbox. This allows replies and inbound messages to be delivered correctly.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-AddressRewriteEntry

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Remove-AddressRewriteEntry** cmdlet to remove an existing address rewrite entry that's no longer needed on an Edge Transport server.

```
Remove-Addressrewriteentry -Identity <AddressRewriteEntryIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes a specific address rewrite entry.

```
Remove-AddressRewriteEntry "Address rewrite entry for
contoso.com"
```

EXAMPLE 2

This example removes all address rewrite entries that include contoso.com in the domain name. It accomplishes the following:

- Retrieves all address rewrite entries.
- Filters the result for entries that have contoso.com or its subdomains as the internal address.

- Removes the filtered entries.

```
Get-AddressRewriteEntry | Where {$_.InternalAddress -like
'*contoso.com'} | Remove-AddressRewriteEntry
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address Rewriting - Edge Transport" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.MessagingPolicies.AddressRewrite.AddressRewriteEntryIdParameter	The *Identity* parameter specifies the address rewrite entry you want to remove. The *Identity* parameter accepts a GUID or the unique address rewrite name. You can omit the *Identity* parameter label.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon

			(：) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AddressRewriteEntry

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available or effective only on Edge Transport servers in on-premises Exchange Server 2013.

Use the **Set-AddressRewriteEntry** cmdlet to modify an existing address rewrite entry that rewrites sender and recipient email addresses in messages sent to or sent from your organization through an Edge Transport server.

```
Set-Addressrewriteentry -Identity <AddressRewriteEntryIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-ExceptionList
<MultiValuedProperty>] [-ExternalAddress <String>] [-InternalAddress
<String>] [-Name <String>] [-OutboundOnly <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the existing address rewrite entry named "Address rewrite entry for contoso.com" with the following settings:

- Changes the original email addresses that are affected by the address rewrite entry to all addresses in the northwindtraders.com domain.
- Changes the Name value to "Address rewrite entry for northwindtraders.com".

```
Set-AddressRewriteEntry "Address rewrite entry for
contoso.com" -Name "Address rewrite entry for
```

```
northwindtraders.com" -InternalAddress northwindtraders.com
```

EXAMPLE 2

This example changes the existing address rewrite entry named "Address entry for all contoso.com email addresses" from inbound and outbound to outbound only. You need to configure a proxy address that matches the rewritten email address for all affected recipients.

```
Set-AddressRewriteEntry "Address rewrite entry for all
contoso.com email addresses" -OutboundOnly $true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Address Rewriting - Edge Transport" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.Messaging Policies.AddressRewrite.AddressRewriteEntryIdParameter	The *Identity* parameter specifies the address rewrite entry you want to modify. You can specify the name or GUID of the address rewrite entry.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
ExceptionList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptionList* parameter specifies the domain names that should be excluded from address rewriting when the *InternalAddress* parameter contains a value that specifies multiple domain names. You can separate multiple domain names included with the *ExceptionList* parameter with commas. For more information

			about how to add values to or remove values from multivalued properties, see Modifying multivalued properties.
ExternalAddress	Optional	System.String	The *ExternalAddress* parameter specifies the final email addresses that you want. If the *InternalAddress* parameter specifies a single email address (chris@contoso.com), the *ExternalAddress* parameter must also specify a single email address (support@contoso.com). If the *InternalAddress* parameter specifies a single domain (contoso.com) or a domain and all subdomains (*.contoso.com), the *ExternalAddress* parameter must specify a single domain (fabrikam.com). **Note:** You can't use the wildcard character (*) with the *ExternalAddress* parameter.

InternalAddress	Optional	System.String	The *InternalAddress* parameter specifies the original email addresses that you want to change. You can use the following values: • **Single email address** david@contoso.com • **Single domain** contoso.com or sales.contoso.com • **Domain and all subdomains** *.contoso.com
Name	Optional	System.String	The *Name* parameter specifies a unique name for this address rewrite entry.
OutboundOnly	Optional	System.Boolean	The *OutboundOnly* parameter enables or disables outbound-only address rewriting. Valid input for this parameter is `$true` or `$false`. The value `$true` means address rewriting occurs in outbound mail only. The value `$false` means address rewriting occurs on outbound mail and also on inbound mail (rewritten email addresses

			are changed back to the original email addresses in inbound mail). The default value is `$false`. **Note:** You must set this parameter to `$true` if the *InternalAddress* parameter contains the wildcard character to rewrite addresses in a domain and all subdomains (*.contoso.com). Also, when you configure outbound-only address rewriting, you need to configure the rewritten email address as a proxy address on the affected recipients. For example, if laura@sales.contoso.com is rewritten to laura@contoso.com, the proxy address laura@contoso.com must be configured on Laura's mailbox. This allows replies and inbound messages to be delivered correctly.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DeliveryAgentConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-DeliveryAgentConnector** cmdlet to retrieve information about a specific delivery agent connector in your organization.

```
Get-DeliveryAgentConnector [-Identity <DeliveryAgentConnectorIdParameter>]
[-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example reads the configuration of the delivery agent connector named Contoso X.400 Connector from Active Directory and displays all of its properties in a list format.

```
Get-DeliveryAgentConnector "Contoso X.400 Connector" |
Format-List
```

EXAMPLE 2

This example retrieves a list of all delivery agent connectors in your organization and displays their names and delivery protocols in a table format.

```
Get-DeliveryAgentConnector | Format-Table
Name,DeliveryProtocol
```

Detailed Description

Delivery agent connectors are used to route messages addressed to foreign systems that don't use the SMTP protocol. When a message is routed to a delivery agent connector, the associated delivery agent performs the content conversion and message delivery. Delivery agent connectors allow queue management of foreign connectors, thereby eliminating the need for storing messages on the file system in Drop and Pickup directories. For more information, see Delivery agents and Delivery Agent connectors.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Delivery agent connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory

			Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.DeliveryAgentConnectorIdParameter	The *Identity* parameter specifies the GUID or name of the delivery agent connector.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-DeliveryAgentConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

The **New-DeliveryAgentConnector** cmdlet creates a delivery agent connector in your organization.

```
New-DeliveryAgentConnector -AddressSpaces <MultiValuedProperty> -
DeliveryProtocol <String> -Name <String> [-Comment <String>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-IsScopedConnector <$true | $false>] [-MaxConcurrentConnections
<Int32>] [-MaxMessageSize <Unlimited>] [-MaxMessagesPerConnection <Int32>]
[-SourceTransportServers <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example creates a delivery agent connector named Contoso X.400 Connector with the following configuration:

- The delivery agent connector is hosted on the following servers:
  - Hub01
  - Hub02
  - Hub05
- The delivery agent connector is designed to handle X.400 connections to a company called Contoso that uses the carrier Fabrikam.
- The address space for the connector is c=US;a=Fabrikam;p=Contoso.

```
New-DeliveryAgentConnector -Name "Contoso X.400 Connector"
-AddressSpaces "X400:c=US;a=Fabrikam;p=Contoso;1" -
DeliveryProtocol "X.400" -SourceTransportServers
Hub01,Hub02,Hub05
```

Detailed Description

Delivery agent connectors are used to route messages addressed to foreign systems that don't utilize the SMTP protocol. When a message is routed to a delivery agent connector, the associated delivery agent performs the content conversion and message delivery. Delivery agent connectors allow queue management of foreign connectors, thereby eliminating the need for storing messages on the file system in the Drop and Pickup directories. For more information, see Delivery agents and Delivery Agent connectors.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Delivery agent connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
AddressSpaces	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *AddessSpaces* parameter specifies the domain names for which this delivery agent connector is responsible. The syntax for entering an address space is as

			follows: *<AddressSpaceType>:<AddressSpace>;<AddressSpaceCost>*. You must enclose each address space in quotation marks (").
DeliveryProtocol	Required	System.String	The *DeliveryProtocol* parameter specifies the communication protocol that determines which delivery agents are responsible for servicing the connector.
Name	Required	System.String	The *Name* parameter specifies the name of this delivery agent connector. The value for the *Name* parameter can't exceed 64 characters.
Comment	Optional	System.String	The *Comment* parameter specifies an optional comment. If you specify a value that contains spaces, enclose the value in quotation marks ("), for example: `"This is an admin note"`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the delivery agent connector is enabled. The default value is `$true`.
IsScopedConnector	Optional	System.Boolean	The *IsScopedConnector*

			parameter specifies the availability of the connector to other Mailbox servers. If the value of this parameter is `$false`, the connector can be used by all Mailbox servers in your organization. If the value of this parameter is `$true`, the connector can only be used by Mailbox servers in the same Active Directory site. The default value is `$false`.
MaxConcurrentConnections	Optional	System.Int32	The *MaxConcurrentConnections* parameter specifies the maximum number of concurrent connections this connector accepts from a specific IP address. The default value is 5.
MaxMessageSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxMessageSize* parameter specifies the maximum size of a message that's allowed to pass through this connector. When you enter a value, qualify the value with one of the following units:

			• B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 65536 through 2147483647 bytes. The default value is `unlimited`.
MaxMessagesPerConnection	Optional	System.Int32	The *MaxMessagesPerConnection* parameter specifies the maximum number of messages this connector accepts per connection. The connector terminates the connection after this limit is reached, and the sending server has to initiate a new connection to send more messages. The default value is 20.
SourceTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SourceTransportServers* parameter specifies the list of Mailbox servers that host this connector. You can specify more than one server by separating their names with commas. By default, only the local

			server on which the command is executed is added to this parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-DeliveryAgentConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

The **Remove-DeliveryAgentConnector** cmdlet removes a specific delivery agent connector

configured in your organization.

```
Remove-DeliveryAgentConnector -Identity
<DeliveryAgentConnectorIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the delivery agent connector named Contoso Delivery Agent Connector.

```
Remove-DeliveryAgentConnector "Contoso Delivery Agent
Connector"
```

Detailed Description

Delivery agent connectors are used to route messages addressed to foreign systems that don't use the SMTP protocol. When a message is routed to a delivery agent connector, the associated delivery agent performs the content conversion and message delivery. Delivery agent connectors allow queue management of foreign connectors, thereby eliminating the need for storing messages on the file system in Drop and Pickup directories. For more information, see Delivery agents and Delivery Agent connectors.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Delivery agent connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DeliveryAgentConnectorIdParameter	The *Identity* parameter specifies the GUID or name of the delivery agent connector.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default

			when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes

			would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-DeliveryAgentConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-DeliveryAgentConnector** cmdlet to configure a specific delivery agent connector in your organization.

```
Set-DeliveryAgentConnector -Identity <DeliveryAgentConnectorIdParameter>
[-AddressSpaces <MultiValuedProperty>] [-Comment <String>] [-Confirm
[<SwitchParameter>]] [-DeliveryProtocol <String>] [-DomainController
<Fqdn>] [-Enabled <$true | $false>] [-Force <SwitchParameter>] [-
IsScopedConnector <$true | $false>] [-MaxConcurrentConnections <Int32>] [-
MaxMessageSize <Unlimited>] [-MaxMessagesPerConnection <Int32>] [-Name
<String>] [-SourceTransportServers <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures restrictions on the delivery agent connector Contoso X.400 Connector. It

makes the following configuration changes:

- Sets the maximum message size allowed through the connector to 10 MB.
- Sets the maximum number of messages allowed per connection to 100.
- Sets the maximum concurrent connections to 10.

```
Set-DeliveryAgentConnector "Contoso X.400 Connector" -
MaxMessageSize 10MB -MaxMessagesPerConnection 100 -
MaxConcurrentConnections 10
```

EXAMPLE 2

This example uses the temporary variable *$ConnectorConfig* to add the address space c=US;p=Fabrikam;a=Contoso;o=Sales to the delivery agent connector Contoso X.400 Connector and also adds the server Hub04 to the list of servers that host the connector.

```
$ConnectorConfig = Get-DeliveryAgentConnector "Contoso
X.400 Connector"
$ConnectorConfig.AddressSpaces +=
"X400:c=US;p=Fabrikam;a=Contoso;o=Sales;1"
$ConnectorConfig.SourceTransportServers += Hub04
Set-DeliveryAgentConnector "Contoso X.400 Connector" -
AddressSpaces $ConnectorConfig.AddressSpaces -
SourceTransportServers
$ConnectorConfig.SourceTransportServers
```

Detailed Description

Delivery agent connectors are used to route messages addressed to foreign systems that don't use the SMTP protocol. When a message is routed to a delivery agent connector, the associated delivery agent performs the content conversion and message delivery. Delivery agent connectors allow queue management of foreign connectors, thereby eliminating the need for storing messages on the file system in Drop and Pickup directories. For more information, see Delivery agents and Delivery Agent connectors.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Delivery agent connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.DeliveryAgentConnectorIdParameter	The *Identity* parameter specifies the GUID or name of the delivery agent connector.
AddressSpaces	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AddessSpaces* parameter specifies the domain names for which this delivery agent connector is responsible. The syntax for entering an address space is as follows: *<AddressSpaceType>:<AddressSpace>;<AddressSpaceCost>*. You must enclose each address space in quotation marks (").
Comment	Optional	System.String	The *Comment* parameter specifies an optional comment. If you specify a value that contains spaces, enclose the value in quotation marks ("), for example: `"This is an admin note"`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DeliveryProtocol	Optional	System.String	The *DeliveryProtocol* parameter specifies the communication protocol that determines which delivery agents are responsible for servicing the connector.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter

			specifies whether the delivery agent connector is enabled. The default value is `$true`.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
IsScopedConnector	Optional	System.Boolean	The *IsScopedConnector* parameter specifies the availability of the connector to other Mailbox servers. If the value of this parameter is `$false`, the connector can be used by all Mailbox servers in your organization. If the value of this parameter is `$true`, the connector can only be used by Mailbox servers

			in the same Active Directory site. The default value is `$false`.
MaxConcurrentConnections	Optional	System.Int32	The *MaxConcurrentConnections* parameter specifies the maximum number of concurrent connections this connector accepts from a specific IP address. The default value is 5.
MaxMessageSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxMessageSize* parameter specifies the maximum size of a message that's allowed to pass through this connector. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 65536 through 2147483647 bytes. The default value is

			`unlimited`.
MaxMessagesPerConnection	Optional	System.Int32	The *MaxMessagesPerConnection* parameter specifies the maximum number of messages this connector accepts per connection. The connector terminates the connection after this limit is reached and the sending server has to initiate a new connection to send more messages. The default value is 20.
Name	Optional	System.String	The *Name* parameter specifies the name of this delivery agent connector. The value for the *Name* parameter can't exceed 64 characters.
SourceTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SourceTransportServers* parameter specifies the list of Mailbox servers that host this connector. You can specify more than one server by separating their names with commas. By default, only the local server on which the command is executed is

			added to this parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-EdgeSubscription

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-EdgeSubscription** cmdlet to retrieve information about Edge Subscriptions in your organization.

```
Get-EdgeSubscription [-Identity <TransportServerIdParameter>] [-
```

```
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves detailed Edge Subscription information for all Edge Transport servers subscribed to your Exchange organization.

```
Get-EdgeSubscription | Format-List
```

EXAMPLE 2

This example retrieves the Edge Subscription information for the Edge Transport server name Edge1 from the domain controller named DC1.

```
Get-EdgeSubscription Edge1 -DomainController
DC1.contoso.com
```

Detailed Description

Run the **Get-EdgeSubscription** cmdlet on an Exchange server in your organization. This cmdlet retrieves the list of Edge Subscriptions. Each Edge Transport server that's subscribed to the Exchange organization has a separate Edge Subscription. You can use this cmdlet to view the Edge Subscription information for a specific Edge Transport server. You can also use this cmdlet to view the Edge Subscription information for all Edge Transport servers subscribed to Active Directory sites.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "EdgeSync" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from

			Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.TransportServerIdParameter	The *Identity* parameter specifies the name of the Edge Transport server for which you want to retrieve Edge Subscription information. The identity is expressed as the host name of the Edge Transport server. If no identity is specified, all Edge Subscriptions are returned.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-EdgeSubscription

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-EdgeSubscription** cmdlet to export an Edge Subscription file from an Edge Transport server and to import the Edge Subscription file to a Mailbox server.

```
New-EdgeSubscription [-AccountExpiryDuration <TimeSpan>] [-Confirm
[<SwitchParameter>]] [-CreateInboundSendConnector <$true | $false>] [-
CreateInternetSendConnector <$true | $false>] [-DomainController <Fqdn>]
[-FileData <Byte[]>] [-FileName <LongPath>] [-Force <SwitchParameter>] [-
Site <AdSiteIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the Edge Subscription file. It should be run on your Edge Transport server.

```
New-EdgeSubscription -FileName "c:
\EdgeServerSubscription.xml"
```

EXAMPLE 2

This example imports the Edge Subscription file generated in EXAMPLE 1 to the Active Directory site `Default-First-Site-Name`. Importing the Edge Subscription file completes the Edge Subscription process. You must run this command on the Mailbox server.

The first command reads the data from the Edge Subscription file and stores it in a temporary variable as a byte-encoded data object. The second command completes the Edge subscription process.

```
[byte[]]$Temp = Get-Content -Path "C:
\EdgeServerSubscription.xml" -Encoding Byte -ReadCount 0
New-EdgeSubscription -FileData $Temp -Site "Default-First-
Site-Name"
```

EXAMPLE 3

This example also imports the Edge Subscription file generated in EXAMPLE 1 to the Active Directory site `Default-First-Site-Name`; however, the end result is accomplished in a single line of code. You must run this command on the Mailbox server.

```
New-EdgeSubscription -FileData ([byte[]]$(Get-Content -Path
"C:\EdgeServerSubscription.xml" -Encoding Byte -ReadCount
0)) -Site "Default-First-Site-Name"
```

Detailed Description

The Edge Transport server doesn't have access to Active Directory. All configuration and recipient information is stored in the Active Directory Lightweight Directory Services (AD LDS) instance. The **New-EdgeSubscription** cmdlet creates the Edge Subscription file that will be imported on a Mailbox server in the Active Directory site to which you want to subscribe this Edge Transport server..

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "EdgeSync" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
AccountExpiryDuration	Optional	System.TimeSpan	The *AccountExpiryDuration* parameter specifies how soon the bootstrap account created by this command will expire. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.

			The value for this parameter must be a minimum of `00:02:00` or 2 minutes.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CreateInboundSendConnector	Optional	System.Boolean	The *CreateInboundSendConnector* parameter specifies whether to create the Send connector to connect the Edge Transport server and the Hub Transport servers. The default value is `$true`. The Send connector address space is set to "--", the smart hosts are set to "--", the Edge Transport server is set as the source server, and Domain Name

			System (DNS) routing is disabled. This parameter is only used when you run the command on the Hub Transport server.
CreateInternetSendConnector	Optional	System.Boolean	The *CreateInternetSendConnector* parameter specifies whether to create the Send connector to connect to the Internet. The default value is `$true`. The Send connector address space is set to all domains (*), the Edge Transport server is set as the source server, and DNS routing is enabled. This parameter is only used when you run the command on the Hub Transport server.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change

			to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
FileData	Optional	System.Byte[]	The *FileData* parameter specifies the byte-encoded data object that contains the Edge Subscription file information. For more information about the syntax required to use this parameter, see Exchange Management Shell quick reference for Exchange 2013. You can only use this parameter when you're running this command on a Mailbox server.
FileName	Optional	Microsoft.Exchange.Data.LongPath	The *FileName* parameter specifies the full path of the Edge

			Subscription file. You can only use this parameter when you're running this command on an Edge Transport server.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter. This switch is useful when you use a script with the Edge Subscription command because it bypasses confirmation. Another scenario in which this switch is useful is when

			you have to subscribe an Edge Transport server again, and you want to overwrite the existing configuration information.
Site	Optional	Microsoft.Exchange.Configuration.Tasks.AdSiteIdParameter	The *Site* parameter specifies the name of the Active Directory site that contains the Mailbox servers with which the Edge Transport servers are associated. This parameter is used and required only when you run the command on a Mailbox server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-EdgeSubscription

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-EdgeSubscription** cmdlet to remove Edge Subscription from the Exchange organization and from the subscribed Edge Transport server.

```
Remove-EdgeSubscription -Identity <TransportServerIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes Edge Subscription for the Edge Transport server.Edge1.

```
Remove-EdgeSubscription -Identity Edge1
```

Detailed Description

The **Remove-EdgeSubscription** cmdlet removes Edge Subscription. After you remove Edge Subscription, synchronization of information from Active Directory to the Active Directory Lightweight Directory Services (AD LDS) instance stops. All the accounts stored in AD LDS are removed, and the Edge Transport server is removed from the source server list of any Send connector.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "EdgeSync" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.TransportServerIdParameter	The *Identity* parameter specifies the identity of the Edge Transport server for which you want to remove Edge Subscription. The identity is expressed as the host name of the Edge Transport server.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this

			configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter. If you don't specify the `Force` switch, the

			command will inform you that the removal of the replicated recipient data from AD LDS can take a long time and will give you the option to cancel the operation.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Start-EdgeSynchronization

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Start-EdgeSynchronization** cmdlet to immediately start synchronization of configuration data from Active Directory to the subscribed Edge Transport servers.

```
Start-EdgeSynchronization [-Confirm [<SwitchParameter>]] [-ForceFullSync
<SwitchParameter>] [-ForceUpdateCookie <SwitchParameter>] [-Server
<ServerIdParameter>] [-TargetServer <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example starts edge synchronization on the Mailbox server named Mailbox01.

```
Start-EdgeSynchronization -Server Mailbox01
```

Detailed Description

The Microsoft Exchange EdgeSync service that runs on Mailbox servers replicates data stored in Active Directory to the local Active Directory Lightweight Directory Services (AD LDS) store on the Edge Transport server. After the initial replication, one-way synchronization of changed data in Active Directory to AD LDS keeps this data up to date.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "EdgeSync" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before

			processing continues. You don't have to specify a value with the *Confirm* switch.
ForceFullSync	Optional	System.Management.Automation.SwitchParameter	The *ForceFullSync* switch specifies whether to initiate a full edge synchronization. If you run the command without this switch, only changes since the last replication are synchronized. If you use this switch, the entire configuration information and recipient data are synchronized.
ForceUpdateCookie	Optional	System.Management.Automation.SwitchParameter	The *ForceUpdateCookie* switch specifies whether to force the Microsoft Exchange EdgeSync service to update the replication cookie even if it encounters an error. The cookie maintains the changes in Active Directory since the previous EdgeSync replication. Normally, the Microsoft Exchange EdgeSync service doesn't update the cookie if it encounters any errors

			during replication.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.
TargetServer	Optional	System.String	The *TargetServer* parameter specifies an Edge Transport server to initiate edge synchronization with. If omitted, all Edge Transport servers are synchronized. You may want to use this parameter to specify a single Edge Transport server for synchronization if a new Edge Transport server has been installed or if that Edge Transport

			server has been unavailable for some time.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-EdgeSynchronization

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-EdgeSynchronization** cmdlet to diagnose whether the subscribed Edge Transport servers have a current and accurate synchronization status.

```
Test-EdgeSynchronization [-ExcludeRecipientTest <SwitchParameter>] [-
FullCompareMode <SwitchParameter>] [-MaxReportSize <Unlimited>] [-
MonitoringContext <$true | $false>] [-TargetServer <String>] <COMMON
PARAMETERS>


Test-EdgeSynchronization -VerifyRecipient <ProxyAddress> <COMMON
PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example diagnoses the synchronization status of subscribed Edge Transport servers, outputs only the first 500 data inconsistencies, and generates events and performance counters for use by System Center Operations Manager 2007.

```
Test-EdgeSynchronization -MaxReportSize 500 -
MonitoringContext $true
```

EXAMPLE 2

This example verifies the synchronization status of the single recipient kate@contoso.com.

```
Test-EdgeSynchronization -VerifyRecipient kate@contoso.com
```

Detailed Description

The **Test-EdgeSynchronization** cmdlet is a diagnostic cmdlet that provides a report of the synchronization status of subscribed Edge Transport servers. You can use the *VerifyRecipient* parameter with this cmdlet to verify that a single recipient has been synchronized to the Active Directory Lightweight Directory Services (AD LDS) instance. The Edge Subscription process establishes one-way replication of recipient and configuration information from Active Directory to AD LDS.

This cmdlet compares the data stored in Active Directory and the data stored in AD LDS. Any inconsistencies in data are reported in the results output by this cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "EdgeSync" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
VerifyRecipient	Required	Microsoft.Exchange.Data.ProxyAddress	The *VerifyRecipient* parameter specifies a single recipient with which to verify the synchronization status. You identify the recipient by specifying a proxy address assigned to the recipient. The proxy address is the recipient's email address. The recipient verification test is mutually exclusive of the test that verifies synchronization of configuration data.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
ExcludeRecipientTest	Optional	System.Management.Automation.SwitchParameter	The *ExcludeRecipientTest* switch specifies whether to exclude validation of recipient data synchronization. If you include this switch, only the synchronization of configuration objects is validated. Validating that recipient data is synchronized takes longer than validating only configuration data. You don't have to include a value with this switch.
FullCompareMode	Optional	System.Management.Automation.SwitchParameter	The *FullCompareMode* switch specifies whether a full comparison of the configuration data between Active Directory and AD LDS instance on the target Edge Transport server

			is performed. If you don't use this switch, a full comparison of replicated configuration data is skipped and the command only tests the Edge synchronization by verifying the replication cookie.
MaxReportSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxReportSize* parameter specifies the total number of objects and properties listed in the results. The results output by this command include a list of all out-of-sync objects and properties in both AD LDS and Active Directory. If the directory services aren't synchronized, a large amount of data can result. If you don't specify a value for this parameter, the default value of 1,000 is used. The minimum value for this parameter is 1. The maximum value for this

			parameter is `unlimited`.
MonitoringContext	Optional	System.Boolean	The *MonitoringContext* parameter includes or excludes the associated monitoring events and performance counters in the results. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you specify the value `$true`, the monitoring events and performance counters are included in the command results. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
TargetServer	Optional	System.String	The *TargetServer* parameter specifies an Edge Transport server to initiate edge

			synchronization with. If omitted, all Edge Transport servers are synchronized. You may want to use this parameter to specify a single Edge Transport server for synchronization if a new Edge Transport server has been installed or if that Edge Transport server has been unavailable for some time.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-EdgeSyncServiceConfig

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-EdgeSyncServiceConfig** cmdlet to retrieve the edge synchronization services settings that control the general synchronization behavior shared by all Microsoft Exchange EdgeSync services.

```
Get-EdgeSyncServiceConfig [-Identity <EdgeSyncServiceConfigIdParameter>]
[-DomainController <Fqdn>] [-Site <AdSiteIdParameter>]
```

Examples

EXAMPLE 1

This example reads the configuration of the Microsoft Exchange EdgeSync service settings named Primary EdgeSync Settings from Active Directory and displays all its properties in a list format.

```
Get-EdgeSyncServiceConfig "Primary EdgeSync Settings" |
Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "EdgeSync" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.EdgeSyncServiceConfigIdParameter	The *Identity* parameter specifies the name of the Microsoft Exchange EdgeSync service configuration you want to view.
Site	Optional	Microsoft.Exchange.Configuration.Tasks.AdSiteIdParameter	The *Site* parameter specifies the Active Directory site that EdgeSync connects to for synchronizing configuration and recipient data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-EdgeSyncServiceConfig

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-EdgeSyncServiceConfig** cmdlet to create edge synchronization service settings that control the general synchronization behavior shared by all EdgeSync services.

```
New-EdgeSyncServiceConfig [-ConfigurationSyncInterval <EnhancedTimeSpan>]
[-Confirm [<SwitchParameter>]] [-CookieValidDuration <EnhancedTimeSpan>]
[-DomainController <Fqdn>] [-FailoverDCInterval <EnhancedTimeSpan>] [-
LockDuration <EnhancedTimeSpan>] [-LockRenewalDuration <EnhancedTimeSpan>]
[-LogEnabled <$true | $false>] [-LogLevel <None | Low | Medium | High>] [-
LogMaxAge <EnhancedTimeSpan>] [-LogMaxDirectorySize <Unlimited>] [-
LogMaxFileSize <Unlimited>] [-LogPath <String>] [-OptionDuration
<EnhancedTimeSpan>] [-RecipientSyncInterval <EnhancedTimeSpan>] [-Site
<AdSiteIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates EdgeSync service settings with the following configuration:

- EdgeSync logging is enabled.
- The log files are stored in the EdgeSyncLog share on Server01.
- The maximum individual log file size is 5 megabytes (MB).
- The log files are kept for 3 days.

```
New-EdgeSyncServiceConfig -LogEnabled $true -LogPath "\
\Server01\EdgeSyncLog" -LogMaxFileSize 5MB -LogMaxAge 3
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "EdgeSync" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
ConfigurationSyncInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConfigurationSyncInterval* parameter specifies how

			frequently the EdgeSync service synchronizes configuration data. The default value is 3 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CookieValidDuration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *CookieValidDuration* parameter specifies how long a cookie record is valid. The default value is 21 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
FailoverDCInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *FailoverDCInterval* parameter specifies how long EdgeSync waits before failing over to another domain controller if it can't read configuration data from Active Directory. The default value is 5 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
LockDuration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *LockDuration* parameter specifies how long an instance of the EdgeSync service can maintain an exclusive lock on the synchronization rights. While an EdgeSync service maintains an exclusive lock on synchronization rights, no other EdgeSync service can take over

			synchronization. The default value is 6 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
LockRenewalDuration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *LockRenewalDuration* parameter specifies how long before the expiry of an exclusive lock an EdgeSync service can renew the lock. The default value is 4 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
LogEnabled	Optional	System.Boolean	The *LogEnabled* parameter enables or disables the EdgeSync log. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
LogLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.EdgeSyncLoggingLevel	The *LogLevel* parameter specifies the EdgeSync logging level. Valid values for this parameter are `None`, `Low`, `Medium` and

			`High`. The default value is `None`.
LogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *LogMaxAge* parameter specifies the maximum duration in days to keep the EdgeSyncLog files. Log files older than the specified value can be overwritten. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
LogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *LogMaxDirectorySize* parameter specifies the maximum amount of disk space the EdgeSyncLog directory can use. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes.

			The value of the *LogMaxFileSize* parameter must be less than or equal to the value of the *LogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the EdgeSyncLLog directory.
LogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *LogMaxFileSize* parameter specifies the maximum log file size for the EdgeSyncLog files. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *LogMaxFileSize* parameter must be less than or equal to the value of the

			LogMaxDirectorySize parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the EdgeSyncLog files.
LogPath	Optional	System.String	The *LogPath* parameter specifies the default location for the EdgeSyncLog files. The default value is `TransportRoles\Logs\EdgeSync\`.
OptionDuration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *OptionDuration* parameter specifies how long an instance of the EdgeSync service can maintain an optional lock on synchronization rights. While an EdgeSync service maintains an optional lock on synchronization rights, another EdgeSync service can take over synchronization after the optional lock has expired if it's initiated using the **Start-**

			EdgeSynchronization command. The default value is 30 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
RecipientSyncInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *RecipientSyncInterval* parameter specifies how frequently the EdgeSync service synchronizes recipient data from the global catalog. The default value is 5 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
Site	Optional	Microsoft.Exchange.Configuration.Tasks.AdSiteIdParameter	The *Site* parameter specifies the Active Directory site that EdgeSync connects to for synchronizing configuration and recipient data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-EdgeSyncServiceConfig

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-EdgeSyncServiceConfig** cmdlet to modify the configuration of edge synchronization service settings that control the general synchronization behavior shared by all EdgeSync services.

```
Set-EdgeSyncServiceConfig -Identity <EdgeSyncServiceConfigIdParameter> [-
ConfigurationSyncInterval <EnhancedTimeSpan>] [-Confirm
[<SwitchParameter>]] [-CookieValidDuration <EnhancedTimeSpan>] [-
DomainController <Fqdn>] [-FailoverDCInterval <EnhancedTimeSpan>] [-
LockDuration <EnhancedTimeSpan>] [-LockRenewalDuration <EnhancedTimeSpan>]
[-LogEnabled <$true | $false>] [-LogLevel <None | Low | Medium | High>] [-
LogMaxAge <EnhancedTimeSpan>] [-LogMaxDirectorySize <Unlimited>] [-
LogMaxFileSize <Unlimited>] [-LogPath <String>] [-Name <String>] [-
OptionDuration <EnhancedTimeSpan>] [-RecipientSyncInterval
<EnhancedTimeSpan>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures the Microsoft Exchange EdgeSync service settings named Primary EdgeSync Settings with the following values:

- EdgeSync logging is enabled and set to the medium detail level.
- The log files are stored in the EdgeSyncLog share on the server named Server01.
- The maximum individual log file size is 5 megabytes (MB).
- The log files are kept for 3 days.

```
Set-EdgeSyncServiceConfig "Primary EdgeSync Settings" -
LogEnabled $true -LogLevel Medium -LogPath "\\Server01
\EdgeSyncLog" -LogMaxFileSize 5MB -LogMaxAge 3
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "EdgeSync" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.EdgeSyncServiceConfigIdParameter	The *Identity* parameter specifies the name of the Microsoft Exchange EdgeSync service you want to configure.
ConfigurationSyncInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConfigurationSyncInterval* parameter specifies how frequently the Microsoft Exchange EdgeSync service synchronizes configuration data. The

			default value is 3 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CookieValidDuration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *CookieValidDuration* parameter specifies how long a cookie record is valid. The default value is 21 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			writes this configuration change to Active Directory.
FailoverDCInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *FailoverDCInterval* parameter specifies how long EdgeSync waits before failing over to another domain controller if it can't read configuration data from Active Directory. The default value is 5 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
LockDuration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *LockDuration* parameter specifies how long an instance of the Microsoft Exchange EdgeSync service can maintain an exclusive lock on the synchronization rights. While an EdgeSync service maintains an exclusive lock on synchronization rights, no other EdgeSync service can take over synchronization. The default value is 6 minutes.

			To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
LockRenewalDuration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *LockRenewalDuration* parameter specifies how long before the expiry of an exclusive lock an EdgeSync service can renew the lock. The default value is 4 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
LogEnabled	Optional	System.Boolean	The *LogEnabled* parameter specifies enables or disables the EdgeSyncLog. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
LogLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.EdgeSyncLoggingLevel	The *LogLevel* parameter specifies the EdgeSync logging level. Valid values for this parameter are `None`, `Low`, `Medium` and `High`. The default value is `None`.

LogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *LogMaxAge* parameter specifies the maximum duration in days to keep the EdgeSyncLog files. Log files older than the specified value can be overwritten. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
LogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *LogMaxDirectorySize* specifies the maximum amount of disk space the EdgeSyncLog directory can use. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *LogMaxFileSize* parameter must be less than or equal

			to the value of the *LogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the EdgeSyncLog directory.
LogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *LogMaxFileSize* parameter specifies the maximum log file size for the EdgeSyncLog files. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *LogMaxFileSize* parameter must be less than or equal to the value of the *LogMaxDirectorySize* parameter. The valid input range for either

			parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the EdgeSyncLog files.
LogPath	Optional	System.String	The *LogPath* parameter specifies the disk location to store the EdgeSyncLog files. The default value is `TransportRoles\Logs\EdgeSync\`.
Name	Optional	System.String	The *Name* parameter specifies a unique name for the EdgeSync service configuration.
OptionDuration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *OptionDuration* parameter specifies how long an instance of the Microsoft Exchange EdgeSync service can maintain an optional lock on the synchronization rights. While an EdgeSync service maintains an optional lock on synchronization rights, another EdgeSync service can take over synchronization after the optional lock has expired

			if it's initiated using the **Start-EdgeSynchronization** command. The default value is 30 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
RecipientSyncInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *RecipientSyncInterval* parameter specifies how frequently the Microsoft Exchange EdgeSync service synchronizes recipient data from the global catalog. The default value is 5 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply

			any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ForeignConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ForeignConnector** cmdlet to view the configuration information for a Foreign connector in the Transport service of a Mailbox server.

```
Get-ForeignConnector [-Identity <ForeignConnectorIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example lists all Foreign connectors in your organization.

```
Get-ForeignConnector
```

EXAMPLE 2

This example displays detailed configuration information for the Foreign connector named Fax

Connector.

```
Get-ForeignConnector "Fax Connector" | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Foreign connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ForeignConnectorIdParameter	The *Identity* parameter specifies the Foreign connector that you want to examine. The *Identity*

			parameter can take any of the following values for the Foreign connector object: • GUID • Connector name • *ServerName* *\ConnectorName*

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ForeignConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-ForeignConnector** cmdlet to create a new Foreign connector in the Transport service of a Mailbox server.

```
New-ForeignConnector -AddressSpaces <MultiValuedProperty> -Name <String>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
IsScopedConnector <$true | $false>] [-SourceTransportServers
<MultiValuedProperty>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a Foreign connector with the following properties:

- Connector name: Contoso Foreign Connector
- Address space: "c=US;a=Fabrikam;P=Contoso"
- Address space type: X.400
- Address space cost: 5
- Source transport servers: Hub01 and Hub02

```
New-ForeignConnector -Name "Contoso Foreign Connector" -
AddressSpaces "X400:c=US;a=Fabrikam;P=Contoso;5" -
SourceTransportServers Hub01,Hub02
```

Detailed Description

A *Foreign connector* uses a Drop directory in the Transport service of a Mailbox server to send messages to a local messaging server that doesn't use SMTP as its primary transport mechanism. These messaging servers are known as foreign gateway servers. Third-party fax gateway servers are examples of foreign gateway servers. The address spaces assigned to a Foreign connector can be SMTP or non-SMTP.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Foreign connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
AddressSpaces	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *AddressSpaces* parameter specifies the domain names to which the Foreign connector sends messages. The complete syntax for entering each address space is as follows: *<AddressSpaceType>:<AddressSpace>;<AddressSpaceCost>* • **AddressSpaceType:** The address space type

			may be SMTP, X400, or any other text string. If you omit the address space type, an SMTP address space type is assumed. • **AddressSpace:** For SMTP address space types, the address space that you enter must be RFC 1035-compliant. For example, *, *.com, and *.contoso.com are permitted, but *contoso.com isn't permitted. For X.400 address space types, the address space that you enter must be RFC 1685-compliant, such as **o=MySite;p=MyOrg;a=adatum;c=us**. For all other values of an address type, you can enter any text for the address space. • **AddressSpaceCost :** The valid input range for the cost is from 1 through 100. A lower cost indicates a better route. If you omit the address space cost, a

			cost of 1 is assumed. If you enter a non-SMTP address space that contains the semicolon character (;), you must specify the address space cost. If you specify the address space type or the address space cost, you must enclose the address space in quotation marks ("). For example, the following address space entries are equivalent: • `"SMTP:contoso.com;1"` • `"contoso.com;1"` • `"SMTP:contoso.com"` • `contoso.com` You may specify multiple address spaces by separating the address spaces with commas, for example: `contoso.com,fabrikam.com`. If you specify the address space type or the address space cost, you must enclose the address space in quotation marks ("), for example: `"contoso.com;2","fabrikam.com;3"`. To add or remove one or more address space

			values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
Name	Required	System.String	The *Name* parameter specifies the name for the Foreign connector.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge

			Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
IsScopedConnector	Optional	System.Boolean	The *IsScopedConnector* parameter specifies the availability of the connector to other Mailbox servers. When the value of this parameter is `$false`, the connector can be used by all Mailbox servers in the Exchange organization. When the value of this parameter is `$true`, the connector can be used only by Mailbox servers in the same Active Directory site. The default value is `$false`.
SourceTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SourceTransportServers* parameter specifies the names of the Mailbox servers that use this Foreign connector. Having a single Foreign connector homed on multiple servers provides fault tolerance and high

			availability if one of the Mailbox servers fails. The default value of this parameter is the name of the server on which this Foreign connector is first installed. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what

			changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ForeignConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-ForeignConnector** cmdlet to delete a Foreign connector in the Transport service of a Mailbox server.

```
Remove-ForeignConnector -Identity <ForeignConnectorIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the Foreign connector named Fax Connector.

```
Remove-ForeignConnector "Fax Connector"
```

Detailed Description

The **Remove-ForeignConnector** cmdlet deletes the object and the configuration information for a Foreign connector.

Caution:
Although a Foreign connector is configured on a local Mailbox server, if you delete a Foreign connector, you may affect mail flow throughout the organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Foreign connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ForeignConnectorIdParameter	The *Identity* parameter specifies the GUID or connector name of a specific Foreign connector. You can also include the server name by using the format *ServerName**ConnectorName*.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon

			(：) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ForeignConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ForeignConnector** cmdlet to modify an existing Foreign connector in the Transport service of a Mailbox server.

```
Set-ForeignConnector -Identity <ForeignConnectorIdParameter> [-
AddressSpaces <MultiValuedProperty>] [-Comment <String>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-DropDirectory <String>]
[-DropDirectoryQuota <Unlimited>] [-Enabled <$true | $false>] [-Force
<SwitchParameter>] [-IsScopedConnector <$true | $false>] [-MaxMessageSize
<Unlimited>] [-Name <String>] [-RelayDsnRequired <$true | $false>] [-
SourceTransportServers <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures a 10 MB message size limit on the existing Foreign connector named Fax Connector.

```
Set-ForeignConnector "Fax Connector" -MaxMessageSize 10MB
```

Detailed Description

A *Foreign connector* uses a Drop directory in the Transport service of a Mailbox server to send messages to a local messaging server that doesn't use SMTP as its primary transport mechanism.

These messaging servers are known as foreign gateway servers. Third-party fax gateway servers are examples of foreign gateway servers. The address spaces assigned to a Foreign connector can be SMTP or non-SMTP.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Foreign connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ForeignConnectorIdParameter	The *Identity* parameter specifies the Foreign connector that you want to modify. The *Identity* parameter can take any of the following values for the Foreign connector object: • GUID • Connector name • *ServerName* *\ConnectorName*
AddressSpaces	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AddressSpaces* parameter specifies the domain names to which the Foreign connector sends messages. The complete syntax for entering each address space is as follows: *<AddressSpaceType>:<AddressSpace>;<AddressSpaceCost>* • **AddressSpaceType:**

			The address space type may be `SMTP`, `X400`, or any other text string. If you omit the address space type, an SMTP address space type is assumed. • **AddressSpace:** For SMTP address space types, the address space that you enter must be RFC 1035-compliant. For example, `*`, `*.com`, and `*.contoso.com` are permitted, but `*contoso.com` isn't permitted. For X.400 address space types, the address space that you enter must be RFC 1685-compliant, such as **o=MySite;p=MyOrg;a=adatum;c=us**. For all other values of address type, you can enter any text for the address space. • **AddressSpaceCost :** The valid input range for the cost is from 1 through 100. A lower cost indicates a better route. If you omit the

			address space cost, a cost of 1 is assumed. If you enter a non-SMTP address space that contains a semicolon (;), you must specify the address space cost. If you specify the address space type or the address space cost, you must enclose the address space in quotation marks ("). For example, the following address space entries are equivalent: • `"SMTP:contoso.com;1"` • `"contoso.com;1"` • `"SMTP:contoso.com"` • `contoso.com` You may specify multiple address spaces by separating the address spaces with commas, for example: `contoso.com,fabrikam.com`. If you specify the address space type or the address space cost, you must enclose the address space in quotation marks ("), for example: `"contoso.com;2","fabrikam.com;3"`. To add or remove one or more address space

			values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
Comment	Optional	System.String	The *Comment* parameter specifies an optional comment. If you specify a value that contains spaces, enclose the value in quotation marks ("), for example: `"This is an admin note"`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

			The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
DropDirectory	Optional	System.String	The *DropDirectory* parameter specifies the name of the Drop directory used by this Foreign connector. All outbound messages sent to address spaces defined by this Foreign connector are put in the specified Drop directory. The location of the Drop directory for each Foreign connector is controlled by the following two items: • **RootDropDirectoryPath parameter in the Set-TransportService cmdlet:** This option is used for all Foreign connectors that exist on the Mailbox server. The value of the *RootDropDirectoryPath*

			parameter may be a local path or a Universal Naming Convention (UNC) path to a remote server. • **DropDirectory parameter in the Set-ForeignConnector cmdlet:** This value is set for each Foreign Connector that exists on the server. By default, the *RootDropDirectoryPath* parameter is blank. This indicates the value of *RootDropDirectoryPath* is the Exchange 2010 installation folder. The default Exchange 2010 installation folder is C:\Program Files\Microsoft\Exchange Server\. By default, the value of the *DropDirectory* parameter is the name of the Foreign connector. If the value of the *DropDirectory* parameter doesn't contain absolute path information, the location of the Drop directory is defined by the

			combination of the *DropDirectory* parameter and the *RootDropDirectoryPath* parameter. If the value of the *DropDirectory* parameter contains absolute path information, the value of the *RootDropDirectoryPath* must be unspecified. The location of the Drop directory is defined only by the value of the *DropDirectory* parameter. The Drop directory isn't created for you. Therefore, you have to manually create each Drop directory folder. The Drop directory must have the following permissions assigned to it: • Network Service: Full Control • System: Full Control • Administrators: Full Control
DropDirectoryQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *DropDirectoryQuota* parameter specifies the maximum size of all message files in the Drop

			directory. When the specified value is reached, no new message files can be copied into the Drop directory until the existing messages are delivered and deleted. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 1 through 2147483647 bytes. If you enter a value of `unlimited`, no message size limit is imposed on the Drop directory. The default value is `unlimited`.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether to enable the Foreign connector. The valid values are `$true` or `$false`. The default value is `$true`.

Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
IsScopedConnector	Optional	System.Boolean	The *IsScopedConnector* parameter specifies the availability of the connector to other Mailbox servers. When the value of this parameter is `$false`, the connector can be used by all Mailbox servers in the Exchange organization. When the value of this parameter is `$true`, the connector can be used only by Mailbox servers in the same Active Directory site. The default value is `$false`.
MaxMessageSize	Optional	Microsoft.Exchange.Da	The *MaxMessageSize*

		ta.Unlimited	parameter specifies the maximum size of a message that can pass through this Foreign connector. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. If you enter a value of `unlimited`, no message size limit is imposed on this Foreign connector. The default value is `unlimited`. The valid input range for this parameter is from 0 through 2147483647 kilobytes. If you set the value of the *MaxMessageSize* parameter to 0, you effectively disable the Foreign connector. However, if you set the value of the *MaxMessageSize* parameter to 0 when the

			value of the *Enabled* attribute is `$true`, you generate event log errors. The preferred method to disable the Foreign connector is to use the *Enabled* parameter.
Name	Optional	System.String	The *Name* parameter specifies a descriptive name for the Foreign connector.
RelayDsnRequired	Optional	System.Boolean	The *RelayDsnRequired* parameter specifies whether a Relay delivery status notification (DSN) is required by the Foreign connector when messages are written to the Drop directory. The valid input values for this parameter are `$true` or `$false`. The default value is `$false`.
SourceTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SourceTransportServers* parameter specifies the names of the Mailbox servers that use this Foreign connector. Having a single Foreign connector homed on multiple Mailbox servers running the Transport service

			provides fault tolerance and high availability if one of the servers fails. The default value of this parameter is the name of the Mailbox server on which this Foreign connector was first installed. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-FrontendTransportService

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-FrontEndTransportService** cmdlet to view the transport configuration information for the Front End Transport service on Client Access servers.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-FrontendTransportService [-Identity
<FrontendTransportServerIdParameter>] [-DomainController <Fqdn>]
```

Examples

Example 1

This example displays a list of all Client Access servers in your organization.

```
Get-FrontEndTransportService
```

Example 2

This example retrieves the detailed configuration information for the Front End Transport service on the Client Access server named CAS01.

```
Get-FrontEndTransportService CAS01 | Format-List
```

Detailed Description

The Front End Transport service runs on all Client Access servers and acts as a stateless proxy for all inbound and outbound external SMTP traffic for the Exchange organization. The Front End Transport service only communicates with the Transport service on a Mailbox server, and doesn't queue any messages locally.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Front End Transport service" entries in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.FrontEndTransportServerIdParameter	The *Identity* parameter specifies the server that you want to view.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-FrontendTransportService

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-18*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-FrontEndTransportService** cmdlet to set the transport configuration options for the Front End Transport service on Client Access servers.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-FrontendTransportService -Identity
<FrontendTransportServerIdParameter> [-AgentLogEnabled <$true | $false>]
[-AgentLogMaxAge <EnhancedTimeSpan>] [-AgentLogMaxDirectorySize
<Unlimited>] [-AgentLogMaxFileSize <Unlimited>] [-AgentLogPath
<LocalLongFullPath>] [-AntispamAgentsEnabled <$true | $false>] [-
AttributionLogEnabled <$true | $false>] [-AttributionLogMaxAge
<EnhancedTimeSpan>] [-AttributionLogMaxDirectorySize <Unlimited>] [-
AttributionLogMaxFileSize <Unlimited>] [-AttributionLogPath
<LocalLongFullPath>] [-Confirm [<SwitchParameter>]] [-
ConnectivityLogEnabled <$true | $false>] [-ConnectivityLogMaxAge
<EnhancedTimeSpan>] [-ConnectivityLogMaxDirectorySize <Unlimited>] [-
ConnectivityLogMaxFileSize <Unlimited>] [-ConnectivityLogPath
<LocalLongFullPath>] [-DnsLogEnabled <$true | $false>] [-DnsLogMaxAge
<EnhancedTimeSpan>] [-DnsLogMaxDirectorySize <Unlimited>] [-
DnsLogMaxFileSize <Unlimited>] [-DnsLogPath <LocalLongFullPath>] [-
DomainController <Fqdn>] [-ExternalDNSAdapterEnabled <$true | $false>] [-
ExternalDNSAdapterGuid <Guid>] [-ExternalDNSProtocolOption <Any |
UseUdpOnly | UseTcpOnly>] [-ExternalDNSServers <MultiValuedProperty>] [-
ExternalIPAddress <IPAddress>] [-InternalDNSAdapterEnabled <$true |
$false>] [-InternalDNSAdapterGuid <Guid>] [-InternalDNSProtocolOption <Any
| UseUdpOnly | UseTcpOnly>] [-InternalDNSServers <MultiValuedProperty>] [-
IntraOrgConnectorProtocolLoggingLevel <None | Verbose>] [-
MaxConnectionRatePerMinute <Int32>] [-MaxReceiveTlsRatePerMinute <Int32>]
[-ReceiveProtocolLogMaxAge <EnhancedTimeSpan>] [-
ReceiveProtocolLogMaxDirectorySize <Unlimited>] [-
ReceiveProtocolLogMaxFileSize <Unlimited>] [-ReceiveProtocolLogPath
<LocalLongFullPath>] [-ResourceLogEnabled <$true | $false>] [-
ResourceLogMaxAge <EnhancedTimeSpan>] [-ResourceLogMaxDirectorySize
<Unlimited>] [-ResourceLogMaxFileSize <Unlimited>] [-ResourceLogPath
<LocalLongFullPath>] [-SendProtocolLogMaxAge <EnhancedTimeSpan>] [-
SendProtocolLogMaxDirectorySize <Unlimited>] [-SendProtocolLogMaxFileSize
```

```
<Unlimited>] [-SendProtocolLogPath <LocalLongFullPath>] [-
SmtpEnableAllTlsVersions <$true | $false>] [-TransientFailureRetryCount
<Int32>] [-TransientFailureRetryInterval <EnhancedTimeSpan>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example sets the *TransientFailureRetryCount* parameter to 3 and sets the *TransientFailureRetryInterval* parameter to 30 seconds for the Front End Transport service on the Client Access server named CAS01.

```
Set-FrontEndTransportService CAS01 -
TransientFailureRetryCount 3 -TransientFailureRetryInterval
00:00:30
```

Example 2

This example sets the *ReceiveProtocolLogPath* parameter to C:\SMTP Protocol Logs\Receive.log for the Front End Transport service on the Client Access server named CAS01.

```
Set-FrontEndTransportService CAS01 -ReceiveProtocolLogPath
"C:\SMTP Protocol Logs\Receive.log"
```

Detailed Description

The Front End Transport service runs on all Client Access servers and acts as a stateless proxy for all inbound and outbound external SMTP traffic for the Exchange organization. The Front End Transport service only communicates with the Transport service on a Mailbox server, and doesn't queue any messages locally.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Front End Transport service" entries in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.FrontendTransportServerId	The *Identity* parameter specifies the server that you want to modify.

		Parameter	
AgentLogEnabled	Optional	System.Boolean	The *AgentLogEnabled* parameter specifies whether the agent log is enabled. The default value is `$true`.
AgentLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *AgentLogMaxAge* parameter specifies the maximum age for the agent log file. Log files older than the specified value are deleted. The default value is `7.00:00:00` or 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Setting the value of the *AgentLogMaxAge* parameter to 00:00:00 prevents the automatic removal of agent log files because of their age.
AgentLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *AgentLogMaxDirectorySize* parameter specifies the maximum size of all agent logs in the agent log directory. When a

			directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *AgentLogMaxFileSize* parameter must be less than or equal to the value of the *AgentLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the agent log directory.
AgentLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *AgentLogMaxFileSize* parameter specifies the maximum size of each agent log file. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB.

			When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *AgentLogMaxFileSize* parameter must be less than or equal to the value of the *AgentLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the agent log files.
AgentLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *AgentLogPath* parameter specifies the default agent log directory location. The default location is `%ExchangeInstallPath%TransportRoles\Logs\FrontEnd\AgentLog`. Setting the value of this parameter to `$null` disables agent logging. However, setting this parameter to `$null` when the value of the

			AgentLogEnabled attribute is `$true` generates event log errors.
AntispamAgentsEnabled	Optional	System.Boolean	The *AntispamAgentsEnabled* parameter specifies whether anti-spam agents are installed on the server specified with the *Identity* parameter. The default value is `$false` for the Front End Transport service on Client Access servers. **Important:** You set this parameter by using a script. You shouldn't modify this parameter manually.
AttributionLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
AttributionLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
AttributionLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
AttributionLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
AttributionLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ConnectivityLogEnabled	Optional	System.Boolean	The *ConnectivityLogEnabled* parameter specifies whether the connectivity log is enabled. The default value is `$true`.
ConnectivityLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConnectivityLogMaxAge* parameter specifies the maximum age for the connectivity log file. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 25

			days for this parameter, use 25.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *ConnectivityLogMaxAge* parameter to 00:00:00 prevents the automatic removal of connectivity log files because of their age.
ConnectivityLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *ConnectivityLogMaxDirectorySize* parameter specifies the maximum size of all connectivity logs in the connectivity log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 1000 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes)

			Unqualified values are treated as bytes. The value of the *ConnectivityLogMaxFileSize* parameter must be less than or equal to the value of the *ConnectivityLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the connectivity log directory.
ConnectivityLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ConnectivityLogMaxFileSize* parameter specifies the maximum size of each connectivity log file. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes)

			• `TB` (terabytes) Unqualified values are treated as bytes. The value of the *ConnectivityLogMaxFileSize* parameter must be less than or equal to the value of the *ConnectivityLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the connectivity log files.
ConnectivityLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ConnectivityLogPath* parameter specifies the default connectivity log directory location. The default location is `%ExchangeInstallPath%TransportRoles\Logs\FrontEnd\Connectivity`. Setting the value of this parameter to `$null` disables connectivity logging. However, setting this parameter to `$null` when the value of the *ConnectivityLogEnabled*

			attribute is `$true` generates event log errors.
DnsLogEnabled	Optional	System.Boolean	The *DnsLogEnabled* parameter specifies whether the DNS log is enabled. The default value is `$false`.
DnsLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *DnsLogMaxAge* parameter specifies the maximum age for the DNS log file. Log files older than the specified value are deleted. The default value is `7.00:00:00` or 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Setting the value of the *DnsLogMaxAge* parameter to `00:00:00` prevents the automatic removal of DNS log files because of their age.
DnsLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *DnsLogMaxDirectorySize* parameter specifies the maximum size of all DNS

			logs in the DNS log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 100 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *DnsLogMaxFileSize* parameter must be less than or equal to the value of the *DnsLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the DNS log directory.
DnsLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *DnsLogMaxFileSize* parameter specifies the maximum size of each DNS log file. When a log file reaches its maximum file size, a new log file is

			created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *DnsLogMaxFileSize* parameter must be less than or equal to the value of the *DnsLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the DNS log files.
DnsLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *DnsLogPath* parameter specifies the DNS log directory location. The default value is blank (`$null`), which indicates no location is configured. If you enable DNS logging, you need to specify a local file path for the DNS log files by using this parameter. If the path

			contains spaces, enclose the entire path value in quotation marks (").
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExternalDNSAdapterEnabled	Optional	System.Boolean	The *ExternalDNSAdapterEnabled* parameter specifies one or more Domain Name System (DNS) servers that Exchange uses for external DNS lookups. When the *ExternalDNSAdapterEnabled* parameter is set to `$true`, DNS lookups of destinations outside the Exchange organization are performed by using the DNS settings of the external network adapter specified by the value of the *ExternalDNSAdapterGuid* parameter. If you want to specify a custom list of

			DNS servers used for external Exchange DNS lookups only, you must specify the DNS servers by using the *ExternalDNSServers* parameter, and you must also set the value of the *ExternalDNSAdapterEnabled* parameter to `$false`. The default value of the *ExternalDNSAdapterEnabled* parameter is `$true`.
ExternalDNSAdapterGuid	Optional	System.Guid	The *ExternalDNSAdapterGuid* parameter specifies the network adapter that has the DNS settings used for DNS lookups of destinations that exist outside the Exchange organization. The concept of an external network adapter and an internal network adapter is only applicable in a multi-homed Exchange server environment. When no particular network adapter is specified as the network adapter for external DNS lookups, the value of the

			ExternalDNSAdapterGuid parameter is `00000000-0000-0000-0000-000000000000`, and external DNS lookups are performed by using the DNS settings of any available network adapter. You may enter the GUID of a specific network adapter to use for external DNS lookups. The default value of the *ExternalDNSAdapterGuid* parameter is `00000000-0000-0000-0000-000000000000`. **Note:** If the value of the *ExternalDNSAdapterEnabled* parameter is set to `$false`, the value of the *ExternalDNSAdapterGuid* parameter is ignored, and the list of DNS servers from the *ExternalDNSServers* parameter is used.
ExternalDNSProtocolOption	Optional	Microsoft.Exchange.Data.ProtocolOption	The *ExternalDNSProtocolOption* parameter specifies which protocol to use when querying external DNS servers. The valid options for this parameter are `Any`, `UseTcpOnly`, and `UseUdpOnly`. The default

			value is `Any`.
ExternalDNSServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalDNSServers* parameter specifies the list of external DNS servers that the server queries when resolving a remote domain. You must separate IP addresses by using commas. The default value is an empty list ({}). **Note:** If the value of the *ExternalDNSAdapterEnabled* parameter is set to `$true`, the *ExternalDNSServers* parameter and its list of DNS servers isn't used.
ExternalIPAddress	Optional	System.Net.IPAddress	The *ExternalIPAddress* parameter specifies the IP address used in the `Received` message header field for every message that travels through the Front End Transport service on a Client Access server. The IP address in the `Received` header field is used for hop count and routing loop detection. The IP address specified by the *ExternalIPAddress* parameter overrides the

			external network adapter's actual IP address. Typically, you would want to set the value of the *ExternalIPAddress* parameter to match the value of your domain's public MX record. The default value of the *ExternalIPAddress* parameter is blank. This means the actual IP address of the external network adapter is used in the `Received` header field.
InternalDNSAdapterEnabled	Optional	System.Boolean	The *InternalDNSAdapterEnabled* parameter specifies one or more DNS servers that Exchange uses for internal DNS lookups. When the *InternalDNSAdapterEnabled* parameter is set to `$true`, DNS lookups of destinations inside the Exchange organization are performed by using the DNS settings of the internal network adapter specified by the value of

			the *InternalDNSAdapterGuid* parameter. If you want to specify a custom list of DNS servers used for internal Exchange DNS lookups only, you must specify the DNS servers by using the *InternalDNSServers* parameter, and you must also set the value of the *InternalDNSAdapterEnabled* parameter to `$false`. The default value of the *InternalDNSAdapterEnabled* parameter is `$true`.
InternalDNSAdapterGuid	Optional	System.Guid	The *InternalDNSAdapterGuid* parameter specifies the network adapter that has the DNS settings used for DNS lookups of servers that exist inside the Exchange organization. The concept of an internal network adapter and an external network adapter is only applicable in a multi-homed Exchange server environment. When no particular network adapter is

			specified as the network adapter for external DNS lookups, the value of the *InternalDNSAdapterGuid* parameter is `00000000-0000-0000-0000-000000000000`, and internal DNS lookups are performed by using the DNS settings of any available network adapter. You may enter the GUID of a specific network adapter to use for internal DNS lookups. The default value of the *InternalDNSAdapterGuid* parameter is `00000000-0000-0000-0000-000000000000`. **Note:** If the value of the *InternalDNSAdapterEnabled* parameter is set to `$false`, the value of the *InternalDNSAdapterGuid* parameter is ignored, and the list of DNS servers from the *InternalDNSServers* parameter is used.
InternalDNSProtocolOption	Optional	Microsoft.Exchange.Data.ProtocolOption	The *InternalDNSProtocolOption* parameter specifies which protocol to use when you query internal DNS servers. Valid

			options for this parameter are `Any`, `UseTcpOnly`, or `UseUdpOnly`. The default value is `Any`.
InternalDNSServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *InternalDNSServers* parameter specifies the list of DNS servers that should be used when resolving a domain name. DNS servers are specified by IP address and are separated by commas. The default value is any empty list ({}). **Note:** If the *InternalDNSAdapterGuid* parameter is set, and the value of the *InternalDNSAdapterEnabled* parameter is set to `$true`, the *InternalDNSServers* parameter and its list of DNS servers isn't used.
IntraOrgConnectorProtocolLoggingLevel	Optional	Microsoft.Exchange.Data.ProtocolLoggingLevel	The *IntraOrgConnectorProtocolLoggingLevel* parameter enables or disables SMTP protocol logging on the implicit and invisible intra-organization Send connectors that are used to transmit messages between Exchange servers

			in the Exchange organization. Valid values for this parameter are `None` or `Verbose`. The value `Verbose` enables protocol logging for the connector. The value `None` disables protocol logging for the connector. The default value is `None`. When the *IntraOrgConnectorProtocolLoggingLevel* parameter is set to `Verbose`, the information is written to the Send connector protocol log specified by the *SendProtocolLog* parameters.
MaxConnectionRatePerMinute	Optional	System.Int32	The *MaxConnectionRatePerMinute* parameter specifies the maximum rate that connections are allowed to be opened with the transport service. If many connections are attempted with the transport service at the same time, the *MaxConnectionRatePerMinute* parameter limits the

			rate that the connections are opened so that the server's resources aren't overwhelmed. The default value is 1200 connections per minute. The valid input range for this parameter is from 1 through 2147483647.
MaxReceiveTlsRatePerMinute	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
ReceiveProtocolLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ReceiveProtocolLogMaxAge* parameter specifies the maximum age of the Receive connector protocol log file. Log files that are older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 20 days for this parameter, use 20.00:00:00. The valid input range for this parameter is from

			00:00:00 through 24855.03:14:07. Setting the value of the *ReceiveProtocolLogMaxAge* parameter to 00:00:00 prevents the automatic removal of Receive connector protocol log files because of their age.
ReceiveProtocolLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *ReceiveProtocolLogMaxDirectorySize* parameter specifies the maximum size of the Receive connector protocol log directory shared by all the Receive connectors that exist on the server. When the maximum directory size is reached, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes.

			The value of the *ReceiveProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *ReceiveProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Receive connector protocol log directory.
ReceiveProtocolLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ReceiveProtocolLogMaxFileSize* parameter specifies the maximum size of the Receive connector protocol log files shared by all the Receive connectors that exist on the server. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes)

			• KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ReceiveProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *ReceiveProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Receive connector protocol log files.
ReceiveProtocolLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ReceiveProtocolLogPath* parameter specifies the path of the protocol log directory for all the Receive connectors that exist on the server. The default location is `%ExchangeInstallPath%TransportRoles\Logs\FrontEnd\ProtocolLog\SmtpReceive`. Setting the

			value of this parameter to `$null` disables protocol logging for all Receive connectors on the server. However, setting this parameter to `$null` when the value of the *ProtocolLoggingLevel* attribute for any Receive connector on the server is `Verbose` generates event log errors. The preferred method of disabling protocol logging is to use the **Set-ReceiveConnector** cmdlet to set the *ProtocolLoggingLevel* to `None` on each Receive connector.
ResourceLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ResourceLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
ResourceLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ResourceLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ResourceLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.

SendProtocolLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *SendProtocolLogMaxAge* parameter specifies the Send connector protocol log file maximum age. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 25 days for this parameter, use 25.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *SendProtocolLogMaxAge* parameter to 00:00:00 prevents the automatic removal of Send connector protocol log files because of their age.
SendProtocolLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *SendProtocolLogMaxDirectorySize* parameter specifies the maximum

			size of the Send connector protocol log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The minimum value is 1 MB. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *SendProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *SendProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Send

			connector protocol log directory.
SendProtocolLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *SendProtocolLogMaxFileSize* parameter specifies the maximum size of the Send connector protocol log files shared by all the Send connectors that exist on a server. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *SendProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *SendProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1

			through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Send connector protocol log files.
SendProtocolLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *SendProtocolLogPath* parameter specifies the location of protocol log storage for the Send connectors. The default location is `%ExchangeInstallPath%TransportRoles\Logs\FrontEnd\ProtocolLog\SmtpSend`. Setting the value of this parameter to `$null` disables protocol logging for all Send connectors on the server. However, setting this parameter to `$null` when the value of the *ProtocolLoggingLevel* or *IntraOrgConnectorProtocolLoggingLevel* attribute for any Send connector on the server is `verbose` generates event log errors. The preferred method of disabling protocol logging is to use the **Set-SendConnector**

			cmdlet to set the *ProtocolLoggingLevel* parameter to None on each Send connector and to set the *IntraOrgConnectorProtocolLoggingLevel* parameter to None.
SmtpEnableAllTlsVersions	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransientFailureRetryCount	Optional	System.Int32	The *TransientFailureRetryCount* parameter specifies the maximum number of immediate connection retries attempted when the server encounters a connection failure with a remote server. The default value is 6. The valid input range for this parameter is from 0 through 15. When the value of this parameter is set to 0, the server doesn't immediately attempt to retry an unsuccessful connection.
TransientFailureRetryInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *TransientFailureRetryInterval* parameter controls the connection interval

			between each connection attempt specified by the *TransientFailureRetryCount* parameter. For the Front End Transport service on a Client Access server, the default value of the *TransientFailureRetryInterval* parameter is 5 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 8 minutes for this parameter, use 00:08:00. The valid input range for this parameter is from 00:00:01 through 12:00:00.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxTransportService

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxTransportService** cmdlet to view the transport configuration information for the Mailbox Transport service on Mailbox servers.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxTransportService [-Identity
<MailboxTransportServerIdParameter>] [-DomainController <Fqdn>]
```

Examples

Example 1

This example displays a list of all Mailbox servers in your organization.

```
Get-MailboxTransportService
```

Example 2

This example retrieves the detailed transport configuration information for the Mailbox Transport

service on the Mailbox server named Mailbox01.

```
Get-MailboxTransportService Mailbox01 | Format-List
```

Detailed Description

The Mailbox Transport service runs on all Mailbox servers and is responsible for delivering messages to and accepting messages from local mailbox databases using remote procedure calls (RPC). The Mailbox Transport service also uses SMTP to send messages to and from the Transport service that runs on all Mailbox servers for routing their ultimate destinations.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Transport service" entries in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxTransportServerIdParameter	The *Identity* parameter specifies the server that you want to view.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxTransportService

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-17*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-MailboxTransportService** cmdlet to view the transport configuration information for the Mailbox Transport service on Mailbox servers.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxTransportService -Identity <MailboxTransportServerIdParameter>
[-Confirm [<SwitchParameter>]] [-ConnectivityLogEnabled <$true | $false>]
[-ConnectivityLogMaxAge <EnhancedTimeSpan>] [-
ConnectivityLogMaxDirectorySize <Unlimited>] [-ConnectivityLogMaxFileSize
<Unlimited>] [-ConnectivityLogPath <LocalLongFullPath>] [-
ContentConversionTracingEnabled <$true | $false>] [-DomainController
<Fqdn>] [-MailboxDeliveryAgentLogEnabled <$true | $false>] [-
MailboxDeliveryAgentLogMaxAge <EnhancedTimeSpan>] [-
MailboxDeliveryAgentLogMaxDirectorySize <Unlimited>] [-
MailboxDeliveryAgentLogMaxFileSize <Unlimited>] [-
MailboxDeliveryAgentLogPath <LocalLongFullPath>] [-
MailboxDeliveryConnectorProtocolLoggingLevel <None | Verbose>] [-
MailboxDeliveryConnectorSmtpUtf8Enabled <$true | $false>] [-
MailboxDeliveryThrottlingLogEnabled <$true | $false>] [-
MailboxDeliveryThrottlingLogMaxAge <EnhancedTimeSpan>] [-
MailboxDeliveryThrottlingLogMaxDirectorySize <Unlimited>] [-
MailboxDeliveryThrottlingLogMaxFileSize <Unlimited>] [-
MailboxDeliveryThrottlingLogPath <LocalLongFullPath>] [-
MailboxSubmissionAgentLogEnabled <$true | $false>] [-
MailboxSubmissionAgentLogMaxAge <EnhancedTimeSpan>] [-
MailboxSubmissionAgentLogMaxDirectorySize <Unlimited>] [-
MailboxSubmissionAgentLogMaxFileSize <Unlimited>] [-
MailboxSubmissionAgentLogPath <LocalLongFullPath>] [-
MaxConcurrentMailboxDeliveries <Int32>] [-MaxConcurrentMailboxSubmissions
<Int32>] [-PipelineTracingEnabled <$true | $false>] [-PipelineTracingPath
<LocalLongFullPath>] [-PipelineTracingSenderAddress <SmtpAddress>] [-
ReceiveProtocolLogMaxAge <EnhancedTimeSpan>] [-
ReceiveProtocolLogMaxDirectorySize <Unlimited>] [-
ReceiveProtocolLogMaxFileSize <Unlimited>] [-ReceiveProtocolLogPath
<LocalLongFullPath>] [-SendProtocolLogMaxAge <EnhancedTimeSpan>] [-
SendProtocolLogMaxDirectorySize <Unlimited>] [-SendProtocolLogMaxFileSize
<Unlimited>] [-SendProtocolLogPath <LocalLongFullPath>] [-
SmtpEnableAllTlsVersions <$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example sets the *ReceiveProtocolLogPath* parameter to C:\SMTP Protocol Logs\Receive.log for the Mailbox Transport service on server Mailbox01.

```
Set-MailboxTransportService Mailbox01 -
ReceiveProtocolLogPath "C:\SMTP Protocol Logs\Receive.log"
```

Detailed Description

The Mailbox Transport service runs on all Mailbox servers and is responsible for delivering messages to and accepting messages from local mailbox databases using a remote procedure call (RPC). The Mailbox Transport service also uses SMTP to send messages to and from the Transport service that runs on all Mailbox servers for routing messages to their ultimate destinations.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Transport service" entries in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxTransportServerIdParameter	The *Identity* parameter specifies the server that you want to modify.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ConnectivityLogEnabled	Optional	System.Boolean	The *ConnectivityLogEnabled* parameter specifies

			whether the connectivity log is enabled. The default value is `$true`.
ConnectivityLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConnectivityLogMaxAge* parameter specifies the maximum age for the connectivity log file. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 25 days for this parameter, use 25.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *ConnectivityLogMaxAge* parameter to 00:00:00 prevents the automatic removal of connectivity log files because of their age.

ConnectivityLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *ConnectivityLogMaxDirectorySize* parameter specifies the maximum size of all connectivity logs in the connectivity log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 1000 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ConnectivityLogMaxFileSize* parameter must be less than or equal to the value of the *ConnectivityLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807

			bytes. If you enter a value of `unlimited`, no size limit is imposed on the connectivity log directory.
ConnectivityLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ConnectivityLogMaxFileSize* parameter specifies the maximum size of each connectivity log file. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *ConnectivityLogMaxFileSize* parameter must be less than or equal to the value of the *ConnectivityLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through

			9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the connectivity log files.
ConnectivityLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ConnectivityLogPath* parameter specifies the default connectivity log directory location. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Mailbox\Connectivity`. Setting the value of this parameter to `$null` disables connectivity logging. However, setting this parameter to `$null` when the value of the *ConnectivityLogEnabled* attribute is `$true` generates event log errors.
ContentConversionTracingEnabled	Optional	System.Boolean	The *ContentConversionTracingEnabled* parameter specifies whether content conversion tracing is enabled. Content conversion tracing captures content conversion failures that occur in the Transport

			service or in the Mailbox Transport service on the Mailbox server. The default value is `$false`. Content conversion tracing captures a maximum of 128 MB of content conversion failures. When the 128 MB limit is reached, no more content conversion failures are captured. Content conversion tracing captures the complete contents of email messages to the path specified by the *PipelineTracingPath* parameter. Make sure that you restrict access to this directory. The permissions required on the directory specified by the *PipelineTracingPath* parameter are as follows: • Administrators: Full Control • Network Service: Full Control • System: Full Control
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory.
MailboxDeliveryAgentLogEnabled	Optional	System.Boolean	The *MailboxDeliveryAgentLogEnabled* parameter specifies whether the agent log for the Mailbox Transport Delivery service is enabled. The default value is `$true`.
MailboxDeliveryAgentLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MailboxDeliveryAgentLogMaxAge* parameter specifies the maximum age for the agent log file of the Mailbox Transport Delivery service. Log files older than the specified value are deleted. The default value is `7.00:00:00` or 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Setting the value of the *MailboxDeliveryAgentLog*

			MaxAge parameter to 00:00:00 prevents the automatic removal of agent log files because of their age.
MailboxDeliveryAgentLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *MailboxDeliveryAgentLogMaxDirectorySize* parameter specifies the maximum size of all Mailbox Transport Delivery service agent logs in the agent log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *MailboxDeliveryAgentLogMaxFileSize* parameter must be less than or equal

			to the value of the *MailboxDeliveryAgentLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the agent log directory.
MailboxDeliveryAgentLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MailboxDeliveryAgentLogMaxFileSize* parameter specifies the maximum size of each agent log file for the Mailbox Transport Delivery service. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *MailboxDeliveryAgentLogMaxFileSize* parameter must be less than or equal to the value of the

			MailboxDeliveryAgentLogMaxDirectorySize parameter. If you enter a value of `unlimited`, no size limit is imposed on the agent log files.
MailboxDeliveryAgentLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *MailboxDeliveryAgentLogPath* parameter specifies the default agent log directory location for the Mailbox Transport Delivery service. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Mailbox\AgentLog\Delivery`. Setting the value of this parameter to `$null` disables agent logging. However, setting this parameter to `$null` when the value of the *MailboxDeliveryAgentLogEnabled* attribute is `$true` generates event log errors.
MailboxDeliveryConnectorProtocolLoggingLevel	Optional	Microsoft.Exchange.Data.ProtocolLoggingLevel	The *MailboxDeliveryConnectorProtocolLoggingLevel* parameter sets the protocol logging level for messages transferred

			from the Transport service to the Mailbox Transport Delivery service on Mailbox servers using SMTP. Valid values for this parameter are `None` and `verbose`. The default value is `None`.
MailboxDeliveryConnectorSmtpUtf8Enabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
MailboxDeliveryThrottlingLogEnabled	Optional	System.Boolean	The *MailboxDeliveryThrottlingLogEnabled* parameter specifies whether the mailbox delivery throttling log is enabled. The default value is `$true`.
MailboxDeliveryThrottlingLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MailboxDeliveryThrottlingLogMaxAge* parameter specifies the maximum age for the mailbox delivery throttling log file. Log files older than the specified value are deleted. The default value is `7.00:00:00` or 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d =

			days, h = hours, m = minutes, and s = seconds. Setting the value of the *MailboxDeliveryThrottlingLogMaxAge* parameter to `00:00:00` prevents the automatic removal of mailbox delivery throttling log files because of their age.
MailboxDeliveryThrottlingLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *MailboxDeliveryThrottlingLogMaxDirectorySize* parameter specifies the maximum size of all mailbox delivery throttling logs in the mailbox delivery throttling log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 200 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are

			treated as bytes. The value of the *MailboxDeliveryThrottlingLogMaxFileSize* parameter must be less than or equal to the value of the *MailboxDeliveryThrottlingLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the mailbox delivery throttling log directory.
MailboxDeliveryThrottlingLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MailboxDeliveryThrottlingLogMaxFileSize* parameter specifies the maximum size of each mailbox delivery throttling log file. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes.

			The value of the *MailboxDeliveryThrottlingLogMaxFileSize* parameter must be less than or equal to the value of the *MailboxDeliveryThrottlingLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the mailbox delivery throttling log files.
MailboxDeliveryThrottlingLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *MailboxDeliveryThrottlingLogPath* parameter specifies the default mailbox delivery throttling log directory location. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Throttling\Delivery`. Setting the value of this parameter to `$null` disables mailbox delivery throttling logging. However, setting this parameter to `$null` when the value of the *MailboxDeliveryThrottlingLogEnabled* attribute is `$true` generates event log errors.

MailboxSubmissionAgentLogEnabled	Optional	System.Boolean	The *MailboxSubmissionAgentLogEnabled* parameter specifies whether the agent log is enabled for the Mailbox Transport Submission service. The default value is `$true`.
MailboxSubmissionAgentLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MailboxSubmissionAgentLogMaxAge* parameter specifies the maximum age for the agent log file of the Mailbox Transport Submission service. Log files older than the specified value are deleted. The default value is `7.00:00:00` or 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Setting the value of the *MailboxSubmissionAgentLogMaxAge* parameter to 00:00:00 prevents the automatic removal of agent log files because of their age.

MailboxSubmissionAgentLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *MailboxSubmissionAgentLogMaxDirectorySize* parameter specifies the maximum size of all Mailbox Transport Submission service agent logs in the agent log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *MailboxSubmissionAgentLogMaxFileSize* parameter must be less than or equal to the value of the *MailboxSubmissionAgentLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no

			size limit is imposed on the agent log directory.
MailboxSubmissionAgentLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MailboxSubmissionAgentLogMaxFileSize* parameter specifies the maximum size of each agent log file for the Mailbox Transport Submission service. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *MailboxSubmissionAgentLogMaxFileSize* parameter must be less than or equal to the value of the *MailboxSubmissionAgentLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on

			the agent log files.
MailboxSubmissionAgentLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *MailboxSubmissionAgentLogPath* parameter specifies the default agent log directory location for the Mailbox Transport Submission service. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Mailbox\AgentLog\Submission`. Setting the value of this parameter to `$null` disables agent logging. However, setting this parameter to `$null` when the value of the *MailboxSubmissionAgentLogEnabled* attribute is `$true` generates event log errors.
MaxConcurrentMailboxDeliveries	Optional	System.Int32	The *MaxConcurrentMailboxDeliveries* parameter specifies the maximum number of delivery threads that the transport service can have open at the same time to deliver messages to mailboxes. The default value is 20. The valid input range for

			this parameter is from 1 through 256. We recommend that you don't modify the default value unless Microsoft Customer Service and Support advises you to do this.
MaxConcurrentMailboxSubmissions	Optional	System.Int32	The *MaxConcurrentMailboxSubmissions* parameter specifies the maximum number of submission threads that the transport service can have open at the same time to send messages from mailboxes. The default value is 20. The valid input range for this parameter is from 1 through 256.
PipelineTracingEnabled	Optional	System.Boolean	The *PipelineTracingEnabled* parameter specifies whether to enable pipeline tracing. Pipeline tracing captures message snapshot files that record the changes made to the message by each transport agent configured in the

			transport service on the server. Pipeline tracing creates verbose log files that accumulate quickly. Pipeline tracing should only be enabled for a short time to provide in-depth diagnostic information that enables you to troubleshoot problems. In addition to troubleshooting, you can use pipeline tracing to validate changes that you make to the configuration of the transport service where you enable pipeline tracing. The default value is `$false`.
PipelineTracingPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *PipelineTracingPath* parameter specifies the location of the pipeline tracing logs. The default location is `%ExchangeInstallPath%TransportRoles\Mailbox\Hub\PipelineTracing`. The path must be local to the Exchange computer. Setting the value of this parameter to `$null` disables pipeline tracing. However, setting this parameter to `$null` when

			the value of the *PipelineTracingEnabled* attribute is `$true` generates event log errors. The preferred method to disable pipeline tracing is to use the *PipelineTracingEnabled* parameter. Pipeline tracing captures the complete contents of email messages to the path specified by the *PipelineTracingPath* parameter. Make sure that you restrict access to this directory. The permissions required on the directory specified by the *PipelineTracingPath* parameter are as follows: • Administrators: Full Control • Network Service: Full Control • System: Full Control
PipelineTracingSenderAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PipelineTracingSenderAddress* parameter specifies the sender email address that invokes pipeline tracing. Only messages

			from this address generate pipeline tracing output. The address can be either inside or outside the Exchange organization. Depending on your requirements, you may have to set this parameter to different sender addresses and send new messages to start the transport agents or routes that you want to test. The default value of this parameter is `$null`.
ReceiveProtocolLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ReceiveProtocolLogMaxAge* parameter specifies the maximum age of the Receive connector protocol log file. Log files that are older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 20 days for this parameter,

			use 20.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *ReceiveProtocolLogMaxAge* parameter to 00:00:00 prevents the automatic removal of Receive connector protocol log files because of their age.
ReceiveProtocolLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *ReceiveProtocolLogMaxDirectorySize* parameter specifies the maximum size of the Receive connector protocol log directory shared by all the Receive connectors that exist on the server. When the maximum directory size is reached, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes)

			Unqualified values are treated as bytes. The value of the *ReceiveProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *ReceiveProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Receive connector protocol log directory.
ReceiveProtocolLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ReceiveProtocolLogMaxFileSize* parameter specifies the maximum size of the Receive connector protocol log files shared by all the Receive connectors that exist on the server. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value,

			qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ReceiveProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *ReceiveProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Receive connector protocol log files.
ReceiveProtocolLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ReceiveProtocolLogPath* parameter specifies the path of the protocol log directory for all the Receive connectors that exist on the server. The default location is %

			`ExchangeInstallPath%TransportRoles\Logs\Mailbox\ProtocolLog\SmtpReceive`. Setting the value of this parameter to `$null` disables protocol logging for all Receive connectors on the server. However, setting this parameter to `$null` when the value of the *ProtocolLoggingLevel* attribute for any Receive connector on the server is `Verbose` generates event log errors. The preferred method of disabling protocol logging is to use the **Set-ReceiveConnector** cmdlet to set the *ProtocolLoggingLevel* to `None` on each Receive connector.
SendProtocolLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *SendProtocolLogMaxAge* parameter specifies the Send connector protocol log file maximum age. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it

			as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 25 days for this parameter, use 25.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *SendProtocolLogMaxAge* parameter to 00:00:00 prevents the automatic removal of Send connector protocol log files because of their age.
SendProtocolLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *SendProtocolLogMaxDirectorySize* parameter specifies the maximum size of the Send connector protocol log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The minimum value is 1 MB. The default value is 250 MB. When you enter a value,

			qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *SendProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *SendProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Send connector protocol log directory.
SendProtocolLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *SendProtocolLogMaxFileSize* parameter specifies the maximum size of the Send connector protocol log files shared by all the Send connectors that exist on a server. When a log

			file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following: When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *SendProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *SendProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Send connector protocol log files.
SendProtocolLogPath	Optional	Microsoft.Exchange.Da	The *SendProtocolLogPath*

		ta.LocalLongFullPath	parameter specifies the location of protocol log storage for the Send connectors. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Mailbox\ProtocolLog\SmtpSend`. Setting the value of this parameter to `$null` disables protocol logging for all Send connectors on the server. However, setting this parameter to `$null` when the value of the *ProtocolLoggingLevel* or *IntraOrgConnectorProtocolLoggingLevel* attribute for any Send connector on the server is `Verbose` generates event log errors. The preferred method of disabling protocol logging is to use the **Set-SendConnector** cmdlet to set the *ProtocolLoggingLevel* parameter to `None` on each Send connector and to set the *IntraOrgConnectorProtocolLoggingLevel* parameter to `None`.

SmtpEnableAllTlsVersions	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-Mailflow

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-09*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-Mailflow** cmdlet to diagnose whether mail can be successfully sent from and delivered to the system mailbox on a Mailbox server. You can also use this cmdlet to verify that

email is sent between Mailbox servers within a defined latency threshold.

```
Test-Mailflow -AutoDiscoverTargetMailboxServer <SwitchParameter> [-
Identity <ServerIdParameter>] <COMMON PARAMETERS>


Test-Mailflow -TargetEmailAddress <String> [-Identity <ServerIdParameter>]
[-TargetEmailAddressDisplayName <String>] <COMMON PARAMETERS>


Test-Mailflow [-Identity <ServerIdParameter>] <COMMON PARAMETERS>


Test-Mailflow -TargetDatabase <DatabaseIdParameter> [-Identity
<ServerIdParameter>] <COMMON PARAMETERS>


Test-Mailflow -TargetMailboxServer <ServerIdParameter> [-Identity
<ServerIdParameter>] <COMMON PARAMETERS>


Test-Mailflow -CrossPremises <$true | $false> [-
CrossPremisesExpirationTimeout <EnhancedTimeSpan>] [-
CrossPremisesPendingErrorCount <Int32>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-ActiveDirectoryTimeout <Int32>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ErrorLatency <Int32>] [-
ExecutionTimeout <Int32>] [-MonitoringContext <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests message flow from the server name Mailbox1 to the server named Mailbox2. Note that you need to run this command while connected to Mailbox1.

```
Test-Mailflow Mailbox1 -TargetMailboxServer Mailbox2
```

EXAMPLE 2

This example tests message flow from the local Mailbox server where you're running this command to the email address john@contoso.com.

```
Test-Mailflow -TargetEmailAddress john@contoso.com
```

Detailed Description

The **Test-Mailflow** cmdlet tests mail submission, transport, and delivery. The cmdlet verifies that each Mailbox server can successfully send itself a message. You can also use this cmdlet to verify that the system mailbox on one Mailbox server can successfully send a message to the system mailbox on another Mailbox server. A system mailbox is required on all servers that are involved in the test.

The test messages are available in the target user or system mailbox. The message subject is `Test-`

`Mailflow <GUID>`, and the message body contains the text `This is a Test-Mailflow probe message.`

The **Test-Mailflow** results are displayed on-screen. The interesting values in the results are:

- **TestMailflowResult** The values returned are typically `Success` or `*FAILURE*`.
- **MessageLatencyTime** The time required to complete the test (deliver the test message). The value uses the syntax *hh:mm:ss.ffff* where *hh* = hours, *mm* = minutes, *ss* = seconds and *ffff* = fractions of a second.

You can write the **Test-Mailflow** results to a file by piping the output to **ConvertTo-Html** or **ConvertTo-Csv** and adding "> <filename>" to the command. For example:

```
Test-Mailflow -AutoDiscoverTargetMailboxServer | ConvertTo-Csv > "C:\My Documents\test-mailflow 2014-05-01.csv"
```

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Testing mail flow" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
AutoDiscoverTargetMailboxServer	Required	System.Management.Automation.SwitchParameter	The *AutoDiscoverTargetMailboxServer* switch specifies whether to automatically populate a list of target Mailbox servers to which to send a test message. The task queries Active Directory to discover all Mailbox servers and then sends each server a test message. When you use this switch, you can't use the *CrossPremises*,

			TargetDatabase, *TargetEmailAddress* or *TargetMailboxServer* parameters.
CrossPremises	Required	System.Boolean	The *CrossPremises* parameter specifies whether the mail flow test will be conducted in cross-premises mode. Set this parameter to `$true` if your organization is using a cross-premises deployment and you want to verify cross-premises mail flow. When you use this parameter, you can't use the *AutoDiscoverTargetMailboxServer*, *TargetDatabase*, *TargetEmailAddress* or *TargetMailboxServer* parameters.
TargetDatabase	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *TargetDatabase* parameter specifies the mailbox database to which test messages are sent.

			When you use this parameter, you can't use the *AutoDiscoverTargetMailboxServer*, *CrossPremises*, *TargetEmailAddress* or *TargetMailboxServer* parameters.
TargetEmailAddress	Required	System.String	The *TargetEmailAddress* parameter specifies the SMTP address of the mailbox to which test messages are sent. Use this parameter to send test messages to a Mailbox server in a remote forest. If this parameter is used, the test is always a remote test. When you use this parameter, you can't use the *AutoDiscoverTargetMailboxServer*, *CrossPremises*, *TargetDatabase* or *TargetMailboxServer* parameters.
TargetMailboxServer	Required	Microsoft.Exchange.Co	The

		nfiguration.Tasks.ServerIdParameter	*TargetMailboxServer* parameter specifies one or more Mailbox servers in the local Exchange organization to which test messages are sent. When you use this parameter, you can't use the *AutoDiscoverTargetMailboxServer*, *CrossPremises*, *TargetDatabase* or *TargetEmailAddress* parameters.
ActiveDirectoryTimeout	Optional	System.Int32	The *ActiveDirectoryTimeout* parameter specifies the number of seconds that elapse before the task provides an informational message about the delay. The default value is 15 seconds.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do

			before processing continues. You don't have to specify a value with the *Confirm* switch.
CrossPremisesExpirationTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *CrossPremisesExpirationTimeout* parameter is used when this cmdlet is run by Microsoft System Center Operations Manager 2007 agents for asynchronous monitoring. We don't recommend using this parameter when running this cmdlet manually.
CrossPremisesPendingErrorCount	Optional	System.Int32	The *CrossPremisesPendingErrorCount* parameter is used when this cmdlet is run by System Center Operations Manager 2007 agents for asynchronous monitoring. We don't recommend using this parameter when running this cmdlet manually.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
ErrorLatency	Optional	System.Int32	The *ErrorLatency* parameter specifies how long to wait for a test message to be delivered before an error event is logged in Microsoft System Center Operations Manager 2007. The default value when a test message is sent to the local Mailbox server is 15 seconds and 180 seconds when a test message is sent to a remote Mailbox server.
ExecutionTimeout	Optional	System.Int32	The *ExecutionTimeout* parameter specifies the maximum time that this task can run before the test is determined to be a failure. If no test message or delivery report arrives before this time expires, the

			task ends and an error is reported. When the task is run in the Exchange Management Shell, the default setting is 240 seconds. When the *MonitoringContext* parameter is used, the default setting is 15 seconds.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the source Mailbox server name from which a test message is sent. If you don't use this parameter, the local Mailbox server is used.
MonitoringContext	Optional	System.Boolean	The *MonitoringContext* parameter includes or excludes the associated monitoring events and performance counters in the results. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you specify the value `$true`, the monitoring events and performance

			counters are included in the command results. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
TargetEmailAddressDisplayName	Optional	System.String	The *TargetEmailAddressDisplayName* parameter specifies a custom display name that's used on events and reports in Microsoft System Center Operations Manager 2007 when the *TargetEmailAddress* parameter is used. If you don't use the *TargetEmailAddressDisplayName* parameter, the events and reports use the email address value specified by the *TargetEmailAddress* parameter.

			This parameter is available only with the *TargetEmailAddress* parameter, and has no effect on the output of the cmdlet outside of Microsoft System Center Operations Manager.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-Message

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Export-Message** cmdlet to copy a message from a queue on a Mailbox server or an Edge Transport server to a specified file path in your organization.

```
Export-Message -Identity <MessageIdentity> [-Confirm [<SwitchParameter>]]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example exports a single message to the specified file path. Because the **Export-Message** cmdlet returns a binary object, you must use the **AssembleMessage** filter to be able to save the message content into a specified location.

```
Export-Message ExchSrv1\contoso.com\1234 | AssembleMessage
-Path "c:\exportfolder\filename.eml"
```

EXAMPLE 2

This example retrieves all messages from the specified queue. The query results are then piped to the **Export-Message** command, and all the messages are copied to individual .eml files. The Internet Message IDs of each message are used as the file names. To accomplish this, the command does the following:

- Retrieves all messages in a specific queue using the **Get-Message** cmdlet.
- The result is pipelined into the **ForEach-Object** cmdlet, which prepares a file name including full path using the temporary variable `$Temp` that consists of the Internet Message ID with .eml extension. The Internet Message ID field contains angled brackets (">" and "<") which need to be removed as they are invalid file names. This is done using the **Replace** method of the temporary variable.
- The **ForEach-Object** cmdlet also exports the message using the file name prepared.

```
Get-Message -Queue "Server1\contoso.com" | ForEach-Object
{$Temp="C:\ExportFolder\"+$_.InternetMessageID
+".eml";$Temp=$Temp.Replace("<","_");$Temp=$Temp.Replace(">
```

```
","_");Export-Message $_.Identity | AssembleMessage -Path
$Temp}
```

Detailed Description

The **Export-Message** cmdlet copies messages from the Delivery queue, the Unreachable queue, or the poison message queue on Mailbox server or an Edge Transport server to a specified file path. Before you export a message, you must first suspend the message. Messages in the poison message queue are already suspended. You can use the **Export-Message** cmdlet to copy messages to the Replay directory of another Mailbox server for delivery.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Data.QueueViewer.MessageIdentity	The *Identity* parameter specifies the message. Valid input for this parameter uses the syntax *Server**Queue**MessageInteger* or *Queue**MessageInteger* or *MessageInteger*, for example, `Mailbox01\contoso.com\5` or `10`. For details about message identity, see the "Message identity" section in Use the Exchange Management Shell to manage queues.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-Message

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-Message** cmdlet to view the details of one or more messages in queues on Mailbox servers or Edge Transport servers.

```
Get-Message [-Filter <String>] [-Server <ServerIdParameter>] <COMMON
PARAMETERS>

Get-Message [-Identity <MessageIdentity>] <COMMON PARAMETERS>

Get-Message [-Queue <QueueIdentity>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-BookmarkIndex <Int32>] [-BookmarkObject
<ExtensibleMessageInfo>] [-IncludeBookmark <$true | $false>] [-
IncludeComponentLatencyInfo <SwitchParameter>] [-IncludeRecipientInfo
<SwitchParameter>] [-ResultSize <Unlimited>] [-ReturnPageInfo <$true |
$false>] [-SearchForward <$true | $false>] [-SortOrder
<QueueViewerSortOrderEntry[]>]
```

Examples

EXAMPLE 1

This example displays detailed information about all messages queued on the local server and received from any sender at the contoso.com domain.

```
Get-Message -Filter {FromAddress -like "*@contoso.com"} |
Format-List
```

EXAMPLE 2

This example lists all messages queued on the local server, received from any sender at the contoso.com domain, and that have an SCL value greater than 3.

```
Get-Message -Filter {FromAddress -like "*@contoso.com" -and
SCL -gt 3}
```

EXAMPLE 3

This example displays all messages queued on the server named Server01. The results are sorted first in ascending order by sender address and then in descending order of size.

```
Get-Message -Server Server01.contoso.com -SortOrder:
+FromAddress,-Size
```

Detailed Description

You can display messages by including the server name as part of the *Identity* parameter or the *Queue* parameter or by including the *Server* parameter with a filter query. The *Identity* parameter, *Queue* parameter, and *Filter* parameter settings are mutually exclusive.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
BookmarkIndex	Optional	System.Int32	The *BookmarkIndex* parameter specifies the position in the result set where the displayed results start. The value of this parameter is a 1-based index in the total result set. The *BookmarkIndex* parameter can't be used with the *BookmarkObject* parameter.
BookmarkObject	Optional	Microsoft.Exchange.Data.QueueViewer.ExtensibleMessageInfo	The *BookmarkObject* parameter specifies the object in the result set where the displayed results start. The *BookmarkObject*

			parameter can't be used with the *BookmarkIndex* parameter.
Filter	Optional	System.String	The *Filter* parameter specifies one or more messages by using OPath filter syntax. The OPath filter includes a message property name followed by a comparison operator and value, for example, `{FromAddress -like "*@contoso.com"}`. For details about filterable message properties and comparison operators, see Message filters and Use the Exchange Management Shell to manage queues. You can specify multiple criteria by using the **and** comparison operator. Property values that aren't expressed as an integer must be enclosed in quotation marks (").

Identity	Optional	Microsoft.Exchange.Data.QueueViewer.MessageIdentity	The *Identity* parameter specifies the message. Valid input for this parameter uses the syntax *Server**Queue**MessageInteger* or *Queue**MessageInteger* or *MessageInteger*, for example, `Mailbox01\contoso.com\5` or `10`. For details about message identity, see the "Message identity" section in Use the Exchange Management Shell to manage queues.
IncludeBookmark	Optional	System.Boolean	The *IncludeBookmark* parameter specifies whether to include the bookmark object when the query results are displayed. The *IncludeBookmark* parameter is valid when it's used with the *BookmarkObject* or *BookmarkIndex* parameters. If you don't specify a value for the *IncludeBookmark*

			parameter, the default value of `$true` is used.
IncludeComponentLatencyInfo	Optional	System.Management.Automation.SwitchParameter	The *IncludeComponentLatencyInfo* switch specifies whether the information about component latency is included in the message properties. If you include this switch, the message objects returned will include latency measurements for each Transport component that has contributed to the local server latency for each queued message.
IncludeRecipientInfo	Optional	System.Management.Automation.SwitchParameter	The *IncludeRecipientInfo* switch specifies whether to display the message recipients in the **Recipients** field. If you don't include the *IncludeRecipientInfo* switch, the **Recipients** field is blank. Storing the results of a `Get-Message -IncludeRecipientInfo` command in a variable

			allows you to display additional properties for the message recipients. The following list describes the available recipient properties: • **Address:** The email address of the recipient. • **Type:** The recipient type, which may be External, Mailbox, or Distribution Group. Distribution Group is used when the destination is an expansion server. • **FinalDestination:** The distinguished name (DN) of the object used to route the message. • **Status:** The recipient status may be `Complete`, `Ready`, or `Retry`. • **LastError:** The SMTP response after the last delivery attempt or a localized error message if the message is placed in

			the unreachable queue.
			For example, to store the recipient information of a message in the contoso.com remote delivery queue that has the **MessageIdentity** value of `1234` to a variable named `$x`, use the following command.
			`$x=Get-Message -I`
			To display the extended recipient properties that are now stored in the `$x` variable, use the following command.
			`$x.Recipients`

Queue	Optional	Microsoft.Exchange.Data.QueueViewer.QueueIdentity	The *Queue* parameter specifies the identity of the queue that contains the messages that you want to display. Valid input for this parameter uses the syntax <*Server*>\<*Queue*> or <*Queue*>, for example, `Mailbox01\contoso.com` or

			`Unreachable`. For details about queue identity, see the "Queue identity" section in Use the Exchange Management Shell to manage queues. If you use the *Queue* parameter, you can't use the *Identity*, *Filter* or *Server* parameters.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
ReturnPageInfo	Optional	System.Boolean	The *ReturnPageInfo* parameter is a hidden parameter. Use it to return information about the total number of results and the index of the first object of the current page. The default value is `$false`.
SearchForward	Optional	System.Boolean	The *SearchForward*

			parameter specifies whether to search forward or backward in the result set. The default value is `$true`. This value causes the result page to be calculated forward from either the start of the result set or forward from a bookmark if specified.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can use the *Server* parameter and the *Filter* parameter in the

			same command. You can't use the *Server* parameter and the *Identity* parameter in the same command.
SortOrder	Optional	Microsoft.Exchange.Data.QueueViewerSortOrderEntry[]	The *SortOrder* parameter specifies an array of message properties used to control the sort order of the result set. Separate each property by using a comma. Prepend a plus sign (+) symbol to the beginning of the property name to display the results in ascending order. Prepend a minus sign (-) symbol to the beginning of the property name to display the results in descending order. If you don't specify a sort order, the result set is displayed in ascending order by **MessageIdentity** integer.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Redirect-Message

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Redirect-Message** cmdlet to drain the active messages from all the delivery queues on a Mailbox server, and transfer those messages to another Mailbox server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Redirect-Message -Server <ServerIdParameter> -Target <MultiValuedProperty>
[-Confirm [<SwitchParameter>]] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example drains the active messages from the delivery queues on the Mailbox server named Mailbox01, and transfers the messages to the server named Mailbox02.

```
Redirect-Message -Server Mailbox01 -Target Mailbox02
```

Detailed Description

When a message queue is drained, the active messages in the queues on the source Mailbox server are routed to the target Mailbox server. After the messages are received and queued by the target Mailbox server, the messages are made redundant. Other considerations include the following:

- Only active messages are drained. Shadow queues aren't drained.

- Messages in the poison message queue aren't drained.
- The source server won't accept new messages while the queues are drained.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.
Target	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *Target* parameter specifies the target Mailbox server where you want to transfer the messages from the drained delivery queues. Enter the server name as a fully qualified domain name (FQDN).

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn’t accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn’t return data.

Remove-Message

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-Message** cmdlet to delete a message from a queue on a Mailbox server or an Edge Transport server.

```
Remove-Message -Filter <String> [-Server <ServerIdParameter>] <COMMON
PARAMETERS>

Remove-Message -Identity <MessageIdentity> <COMMON PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]] [-WithNDR <$true | $false>]
```

Examples

EXAMPLE 1

This example removes all messages that meet the following criteria without generating NDRs:

- The messages are sent by the sender Kweku@contoso.com.
- The messages are queued on the server Server1.

```
Remove-Message -Server Server1 -Filter {FromAddress -eq
"Kweku@contoso.com"} -WithNDR $false
```

Detailed Description

A message being transmitted to multiple recipients might be located in multiple queues. If you specify an *Identity* parameter, the message is removed from a single queue if that identity matches only a single message. If the identity matches more than one message, you receive an error. To remove a message from more than one queue in a single operation, you must use the *Filter* parameter. If you try to remove a message currently being delivered, the message status changes to `PendingDelete`. Message delivery isn't interrupted, but if the delivery fails and causes the message to re-enter the queue, it's then removed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues"

entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Filter	Required	System.String	The *Filter* parameter specifies one or more messages by using OPath filter syntax. The OPath filter includes a message property name followed by a comparison operator and value, for example, `{FromAddress -like "*@contoso.com"}`. For details about filterable message properties and comparison operators, see Message filters and Use the Exchange Management Shell to manage queues. You can specify multiple criteria by using the **and** comparison operator. Property values that aren't expressed as an integer must be enclosed in quotation marks (").
Identity	Required	Microsoft.Exchange.Data.QueueViewer.MessageIdentity	The *Identity* parameter specifies the message. Valid input for this parameter uses the

			syntax *Server**Queue**MessageInteger* or *Queue**MessageInteger* or *MessageInteger*, for example, `Mailbox01\contoso.com\5` or `10`. For details about message identity, see the "Message identity" section in Use the Exchange Management Shell to manage queues.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name

			(DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can use the *Server* parameter and the *Filter* parameter in the same command. You can't use the *Server* parameter and the *Identity* parameter in the same command.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WithNDR	Optional	System.Boolean	The *WithNDR* parameter specifies whether a non-delivery report (NDR) is returned to the sender of a message. The default value is `$true`. This parameter can be used with both the *Identity*

			parameter and *Filter* parameter sets.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Resume-Message

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Resume-Message** cmdlet to enable delivery of a previously suspended message in a queue on a Mailbox server or Edge Transport server.

```
Resume-Message -Filter <String> [-Server <ServerIdParameter>] <COMMON
PARAMETERS>

Resume-Message -Identity <MessageIdentity> <COMMON PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resumes delivery of all messages in a suspended state and for which the following conditions are true:

- The messages were sent by the sender kweku@contoso.com.
- The messages are queued on the server Server1.
- The messages will expire before 15:00 on January 5, 2013.

```
Resume-Message -Server Server1 -Filter {FromAddress -eq
"kweku@contoso.com" -and ExpirationTime -lt "1/5/2013 3:00
PM"}
```

Detailed Description

A message being sent to multiple recipients might be located in multiple queues. If you specify an *Identity* parameter, the message is resumed in a single queue if that identity matches only a single message. If the identity matches more than one message, you receive an error. To resume a message in more than one queue in a single operation, you must use the *Filter* parameter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Filter	Required	System.String	The *Filter* parameter specifies one or more messages by using OPath filter syntax. The OPath filter includes a message property name followed by a comparison operator and value, for example, `{FromAddress -like "*@contoso.com"}`. For details about filterable message properties and comparison operators, see Message filters and Use the Exchange Management Shell to manage queues. You can specify multiple

			criteria by using the **and** comparison operator. Property values that aren't expressed as an integer must be enclosed in quotation marks (").
Identity	Required	Microsoft.Exchange.Data.QueueViewer.MessageIdentity	The *Identity* parameter specifies the message. Valid input for this parameter uses the syntax *Server**Queue**MessageInteger* or *Queue**MessageInteger* or *MessageInteger*, for example, `Mailbox01\contoso.com\5` or `10`. For details about message identity, see the "Message identity" section in Use the Exchange Management Shell to manage queues.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can use the *Server* parameter and the *Filter* parameter in the same command. You can't use the *Server* parameter and the *Identity* parameter in the same command.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Suspend-Message

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Suspend-Message** cmdlet to prevent delivery of a particular message in a queue on a Mailbox server or an Edge Transport server.

```
Suspend-Message -Filter <String> [-Server <ServerIdParameter>] <COMMON
PARAMETERS>

Suspend-Message -Identity <MessageIdentity> <COMMON PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example prevents delivery of all messages for which the following conditions are true:

- The messages are sent by the sender kweku@contoso.com.
- The messages are queued on the server Server1.

```
Suspend-Message -Server Server1 -Filter {FromAddress -eq
```

```
"kweku@contoso.com"}
```

Detailed Description

A message already in delivery won't be suspended. Delivery will continue and the message status will be `PendingSuspend`. If the delivery fails, the message will re-enter the queue and it will then be suspended. You can't suspend a message that's in the Submission queue or poison message queue.

A message being sent to multiple recipients might be located in multiple queues. If you specify an *Identity* parameter, the message is suspended in a single queue if that identity matches only a single message. If the identity matches more than one message, you receive an error. To suspend a message in more than one queue in a single operation, you must use the *Filter* parameter.

For instructions on how to resume a suspended message, see Resume-Message.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Filter	Required	System.String	The *Filter* parameter specifies one or more messages by using OPath filter syntax. The OPath filter includes a message property name followed by a comparison operator and value, for example, `{FromAddress -like "*@contoso.com"}`. For details about filterable message properties and comparison operators, see Message filters and Use the Exchange Management Shell to manage queues.

			You can specify multiple criteria by using the **and** comparison operator. Property values that aren't expressed as an integer must be enclosed in quotation marks (").
Identity	Required	Microsoft.Exchange.Data.QueueViewer.MessageIdentity	The *Identity* parameter specifies the message. Valid input for this parameter uses the syntax *Server**Queue**MessageInteger* or *Queue**MessageInteger* or *MessageInteger*, for example, `Mailbox01\contoso.com\5` or `10`. For details about message identity, see the "Message identity" section in Use the Exchange Management Shell to manage queues.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`)

			in the syntax.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can use the *Server* parameter and the *Filter* parameter in the same command. You can't use the *Server* parameter and the *Identity* parameter in the same command.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply

			any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MessageTrackingLog

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-09*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MessageTrackingLog** cmdlet to search for message delivery information stored in the message tracking log.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MessageTrackingLog [-EventId <String>] [-InternalMessageId <String>]
[-MessageId <String>] [-MessageSubject <String>] [-Recipients <String[]>]
[-Reference <String>] [-Sender <String>] [-DomainController <Fqdn>] [-End
<DateTime>] [-ResultSize <Unlimited>] [-Server <ServerIdParameter>] [-
Start <DateTime>]
```

Examples

EXAMPLE 1

This example searches the message tracking logs on the Mailbox server named Mailbox01 for information about all messages sent from March 13, 2013, 09:00 to March 15, 2013, 17:00 by the sender john@contoso.com.

```
Get-MessageTrackingLog -Server Mailbox01 -Start "03/13/2013
09:00:00" -End "03/15/2013 17:00:00" -Sender
"john@contoso.com"
```

Detailed Description

A unique message tracking log exists for the Transport service on a Mailbox server, for the Mailbox Transport service on a Mailbox server, and on an Edge Transport server. The message tracking log is a comma-separated value (CSV) file that contains detailed information about the history of each email message as it travels through an Exchange server.

The field names displayed in the results from the **Get-MessageTrackingLog** cmdlet are similar to the actual field names used in the message tracking logs. The differences are:

- The dashes are removed from the field names. For example **internal-message-id** is displayed as `InternalMessageId`.
- The **date-time** field is displayed as `Timestamp`.
- The **recipient-address** field is displayed as `Recipients`.
- The **sender-address** field is displayed as `Sender`.

For more information about the message tracking log files, see Message tracking.

The **Get-MessageTrackingLog** results are displayed on-screen. You can write the results to a file by piping the output to **ConvertTo-Html** or **ConvertTo-Csv** and adding "> <filename>" to the command. For example:

```
Get-MessageTrackingLog -Start "03/13/2014 09:00:00" -End
"03/13/2014 09:10:00" | ConvertTo-Html > "C:\My Documents
\message track.html"
```

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Message tracking" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
End	Optional	System.DateTime	The *End* parameter specifies the end date and time of the date range. Message delivery information is returned up to, but not including, the specified date and time. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If

			you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
EventId	Optional	System.String	The *EventId* parameter filters the message tracking log entries by the value of the **EventId** field. The **EventId** value classifies classify each message event. Example values include `DSN`, `Defer`, `Deliver`, `Send`, or `Receive`.
InternalMessageId	Optional	System.String	The *InternalMessageId* parameter filters the message tracking log entries by the value of the **InternalMessageId** field. The **InternalMessageId** value is a message identifier that's assigned by the Exchange server that's currently processing the message. The value of the **internal-message-id** for a specific message is different in the message tracking log of every Exchange server that's involved in the

			delivery of the message.
MessageId	Optional	System.String	The *MessageId* parameter filters the message tracking log entries by the value of the **MessageId** field. The value of **MessageId** corresponds to the value of the `Message-Id:` header field in the message. If the `Message-ID` header field is blank or doesn't exist, an arbitrary value is assigned.
MessageSubject	Optional	System.String	The *MessageSubject* parameter filters the message tracking log entries by the value of the message subject. The value of the *MessageSubject* parameter automatically supports partial matches without using wildcards or special characters. For example, if you specify the *MessageSubject* value `sea`, the results include messages with `Seattle` in the subject. By default, message subjects are stored in the message

			tracking logs.
Recipients	Optional	System.String[]	The *Recipients* parameter filters the message tracking log entries by the SMTP email address of the message recipients. Multiple recipients in a single message are logged in a single message tracking log entry. Unexpanded distribution group recipients are logged by using the group's SMTP email address. You can specify multiple recipient email addresses separated by commas.
Reference	Optional	System.String	The *Reference* parameter filters the message tracking log entries by the value of the **Reference** field. The Reference field contains additional information for specific types of events. For example, the **Reference** field value for a DSN message tracking entry contains the **InternalMessageId** value of the message that

			caused the DSN. For many types of events, the value of **Reference** is blank.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Sender	Optional	System.String	The *Sender* parameter filters the message tracking log entries by the sender's SMTP email address.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command

			is run on the local server.
Start	Optional	System.DateTime	The *Start* parameter specifies the start date and time of the date range. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MessageTrackingReport

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MessageTrackingReport** cmdlet to return data for a specific message tracking report. This cmdlet is used by the delivery reports feature.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MessageTrackingReport -Identity <MessageTrackingReportId> [-
BypassDelegateChecking <SwitchParameter>] [-DetailLevel <Basic | Verbose>]
[-DomainController <Fqdn>] [-DoNotResolve <SwitchParameter>] [-
RecipientPathFilter <SmtpAddress>] [-Recipients <String[]>] [-
ReportTemplate <Summary | RecipientPath>] [-ResultSize <Unlimited>] [-
Status <Unsuccessful | Pending | Delivered | Transferred | Read>] [-
TraceLevel <Low | Medium | High>]
```

Examples

EXAMPLE 1

This example gets the message tracking report for messages sent from one user to another. This example returns the summary of the message tracking report for a message that David Jones sent to Wendy Richardson.

```
$Temp = Search-MessageTrackingReport -Identity "David
Jones" -Recipients "wendy@contoso.com"
Get-MessageTrackingReport -Identity
$Temp.MessageTrackingReportID -ReportTemplate Summary
```

EXAMPLE 2

This example gets the message tracking report for the following scenario: The user Cigdem Akin was expecting an email message from joe@contoso.com that never arrived. She contacted the Help desk, which needs to generate the message tracking report on behalf of Cigdem and doesn't need to see the display names.

This example searches the message tracking data for the specific message tracking reports, and then returns detailed troubleshooting information for the specific recipient path.

```
Search-MessageTrackingReport -Identity "Cigdem Akin" -
```

```
Sender "joe@contoso.com" -ByPassDelegateChecking -
DoNotResolve | ForEach-Object { Get-MessageTrackingReport -
Identity $_.MessageTrackingReportID -DetailLevel Verbose -
BypassDelegateChecking -DoNotResolve -RecipientPathFilter
"cigdem@fabrikam.com" -ReportTemplate RecipientPath }
```

Detailed Description

The **Get-MessageTrackingReport** cmdlet requires you to specify the ID for the message tracking report you want to view. Therefore, first you need to use the **Search-MessageTrackingReport** cmdlet to find the message tracking report ID for a specific message. You then pass the message tracking report ID from the output of the **Search-MessageTrackingReport** cmdlet to the **Get-MessageTrackingReport** cmdlet. For more information, see Search-MessageTrackingReport.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Message tracking" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.Tracking.MessageTrackingReportId	The *Identity* parameter specifies the ID of the message tracking report ID to retrieve. You should run the **Search-MessageTrackingReport** cmdlet to find the message tracking report ID for the specific message you're tracking, and then pass the value of the **MessageTrackingReport ID** field to this parameter.

BypassDelegateChecking	Optional	System.Management.Automation.SwitchParameter	The *BypassDelegateChecking* switch allows Help desk staff and administrators to retrieve message tracking reports for any user. You don't have to specify a value with this switch. By default, each user can only see the message tracking reports for messages sent or received by the user. When you use this switch, Exchange allows you to view the message tracking reports for message exchanges among other users.
DetailLevel	Optional	Microsoft.Exchange.InfoWorker.Common.MessageTracking.MessageTrackingDetailLevel	This parameter is available only in on-premises Exchange 2013. The *DetailLevel* parameter specifies the amount of detail to be displayed for the message tracking report. You can use one of the following values: • `Basic` • `Verbose` If you specify `Basic`, simple delivery report information is displayed, which is more appropriate

			for information workers. If you specify `verbose`, full report information is displayed, including server names and physical topology information.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
DoNotResolve	Optional	System.Management.Automation.SwitchParameter	The *DoNotResolve* switch prevents the resolution of email addresses to display names. This improves performance, but the end result may not be as easy to interpret because it's missing the display names. You don't have to specify a value with this switch.
RecipientPathFilter	Optional	Microsoft.Exchange.Data.SmtpAddress	The *RecipientPathFilter* parameter specifies the recipient for which the command returns the

			detailed tracking report. Use this parameter when you're using the `RecipientPath` report template.
Recipients	Optional	System.String[]	The *Recipients* parameter specifies the recipients for whom you want to retrieve the message tracking data. You can use this parameter to specify the recipients in the report details if you're using the `Summary` report template.
ReportTemplate	Optional	Microsoft.Exchange.InfoWorker.Common.MessageTracking.ReportTemplate	The *ReportTemplate* parameter specifies a predefined format for the output. You can either return a summary for all recipients or a detailed tracking report for one recipient. You can specify one of the following values: • `RecipientPath` • `Summary`
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that

			match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Status	Optional	Microsoft.Exchange.Management.Tracking._DeliveryStatus	The *Status* parameter specifies the delivery status codes you're interested in. You can specify one of the following values: • `Delivered` • `Read` • `Pending` • `Transferred` • `Unsuccessful`
TraceLevel	Optional	Microsoft.Exchange.Management.Tracking.TraceLevel	This parameter is available only in on-premises Exchange 2013. The *TraceLevel* parameter specifies whether additional trace details are included in the output of the message tracking report. This parameter is intended to be used when troubleshooting message tracking issues. The acceptable values for the *TraceLevel* parameter are: • `Low` Minimal additional data is returned, including servers that were accessed, timing, message tracking search

			result counts, and any error information. • `Medium` In addition to the data returned for the `Low` setting, the actual message tracking search results are also returned. • `High` Full diagnostic data is returned.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Search-MessageTrackingReport

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Search-MessageTrackingReport** cmdlet to find the unique message tracking report based on the search criteria provided. You can then pass this message tracking report ID to the **Get-MessageTrackingReport** cmdlet to get full message tracking information. For more information, see Get-MessageTrackingReport. The message tracking report cmdlets are used by the delivery reports feature.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Search-MessageTrackingReport -Identity <MailboxIdParameter> -Sender
<SmtpAddress> [-BypassDelegateChecking <SwitchParameter>] [-DoNotResolve
<SwitchParameter>] [-MessageEntryId <String>] [-MessageId <String>] [-
ResultSize <Unlimited>] [-Subject <String>] [-TraceLevel <Low | Medium |
High>] <COMMON PARAMETERS>
```

```
Search-MessageTrackingReport -Identity <MailboxIdParameter> [-
BypassDelegateChecking <SwitchParameter>] [-DoNotResolve
<SwitchParameter>] [-MessageEntryId <String>] [-MessageId <String>] [-
Recipients <SmtpAddress[]>] [-ResultSize <Unlimited>] [-Subject <String>]
[-TraceLevel <Low | Medium | High>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example searches the message tracking report for messages sent from one user to another. This example returns the message tracking report for a message that David Jones sent to Wendy Richardson.

```
Search-MessageTrackingReport -Identity "David Jones" -
Recipients "wendy@contoso.com"
```

EXAMPLE 2

This example searches the message tracking report for the following scenario: The user Cigdem Akin was expecting an email message from joe@contoso.com that never arrived. She contacted the Help desk, which needs to generate the message tracking report on behalf of Cigdem and doesn't need to see the display names.

This example returns the message tracking reports that the Help desk can analyze to resolve the issue.

```
Search-MessageTrackingReport -Identity "Cigdem Akin" -
Sender "joe@contoso.com" -BypassDelegateChecking -
DoNotResolve
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Message tracking" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the name of the mailbox for whom the message tracking report is being searched. The message tracking report only contains the tracking events related to the specified mailbox. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/

			ou=AdministrativeGroup/ cn=Recipients/ cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Sender	Required	Microsoft.Exchange.Data.SmtpAddress	The *Sender* parameter specifies the email address of the message sender. By default, this command searches for messages sent by the user specified in the *Identity* parameter. If you want to search the message tracking report for a message sent to the user, you must specify the email address of the sender of that message using this parameter.
BypassDelegateChecking	Optional	System.Management.Automation.SwitchParameter	The *BypassDelegateChecking* switch allows Help desk staff and administrators to track messages for any user. End-users can only track messages that they send or receive. You don't have to specify a value with this switch.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
DoNotResolve	Optional	System.Management.Automation.SwitchParameter	The *DoNotResolve* switch prevents the resolution of email addresses to display names. This improves performance, but the end result may not be as easy to interpret because it's missing the display names. You don't have to specify a value with this switch.

MessageEntryId	Optional	System.String	This parameter is reserved for internal Microsoft use.
MessageId	Optional	System.String	The *MessageId* parameter specifies the Internet message ID of the message for which you want to get the tracking report.
Recipients	Optional	Microsoft.Exchange.Data.SmtpAddress[]	The *Recipients* parameter specifies the recipients for whom you want to get the tracking report.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Subject	Optional	System.String	The *Subject* parameter specifies text to be used to track messages that have the specified text in their subject lines.
TraceLevel	Optional	Microsoft.Exchange.Management.Tracking.TraceLevel	This parameter is available only in on-premises Exchange 2013. The *TraceLevel* parameter

			specifies whether additional trace details will be included in the output of the message tracking report. This parameter is intended to be used when troubleshooting message tracking issues. The acceptable values for the *TraceLevel* parameter are: • `Low` Minimal additional data is returned, including servers that were accessed, timing, message tracking search result counts, and any error information. • `Medium` In addition to all the data returned for the `Low` setting, the actual message tracking search results are also returned. • `High` Full diagnostic data is returned.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-NetworkConnectionInfo

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-NetworkConnectionInfo** cmdlet to view the network configuration information for all network adapters configured on the local server.

```
Get-NetworkConnectionInfo [-Identity <ServerIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves network configuration information for all network adapters on the server named `Mailbox01`.

```
Get-NetworkConnectionInfo Mailbox01
```

Detailed Description

This cmdlet retrieves the following configuration information for each network adapter configured on the server:

- **Name:** This field displays the name of the network adapter. This name indicates the manufacturer and model of the network adapter, or the administrator-specified name of the network adapter.
- **DnsServers:** This field displays the DNS servers used by the network adapter. The server names are separated by commas.
- **IPAddresses:** This field displays the IP addresses used by the network adapter. The IP addresses are separated by commas.
- **AdapterGuid:** This field displays the GUID assigned to the network adapter by Windows.
- **MacAddress:** This field displays the media access control (MAC) address of the network adapter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport service" and "Edge Transport server" entries in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the server you want to query. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Identity* parameter, the command is run on the local server.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-Queue

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-07*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-Queue** cmdlet to view configuration information for queues on Mailbox servers or

Edge Transport servers.

```
Get-Queue [-Filter <String>] [-Server <ServerIdParameter>] <COMMON
PARAMETERS>


Get-Queue [-Identity <QueueIdentity>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-BookmarkIndex <Int32>] [-BookmarkObject
<ExtensibleQueueInfo>] [-Exclude <QueueViewerIncludesAndExcludes>] [-
Include <QueueViewerIncludesAndExcludes>] [-IncludeBookmark <$true |
$false>] [-ResultSize <Unlimited>] [-ReturnPageInfo <$true | $false>] [-
SearchForward <$true | $false>] [-SortOrder <QueueViewerSortOrderEntry[]>]
```

Examples

EXAMPLE 1

This example displays detailed information for all queues on the Mailbox server on which the command is run.

```
Get-Queue | Format-List
```

EXAMPLE 2

This example lists the queues that contain more than 100 messages.

```
Get-Queue -Filter {MessageCount -gt 100}
```

EXAMPLE 3

This example displays detailed information for a specific queue that exists on the Mailbox server named Server1.

```
Get-Queue Server1\contoso.com | Format-List
```

EXAMPLE 4

This example lists only the external queues.

```
Get-Queue -Exclude Internal
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
BookmarkIndex	Optional	System.Int32	The *BookmarkIndex* parameter specifies the position in the result set where the displayed results start. The value of this parameter is a 1-based index in the total result set. The *BookmarkIndex* parameter can't be used with the *BookmarkObject* parameter.
BookmarkObject	Optional	Microsoft.Exchange.Data.QueueViewer.ExtensibleQueueInfo	The *BookmarkObject* parameter specifies the object in the result set where the displayed results start. The *BookmarkObject* parameter can't be used with the *BookmarkIndex* parameter.
Exclude	Optional	Microsoft.Exchange.Data.QueueViewerIncludesAndExcludes	The *Exclude* parameter specifies the types of queues you want to exclude from the results. Valid values for this parameter are: • Internal • External

			• A valid queue `DeliveryType` value. For details, see the NextHopSolutionKey section in Queues.
Filter	Optional	System.String	The *Filter* parameter specifies one or more queues by using OPath filter syntax. The OPath filter includes a queue property name followed by a comparison operator and value, for example, `{NextHopDomain -eq "contoso.com"}`. For details about filterable queue properties and comparison operators, see Queue filters and Use the Exchange Management Shell to manage queues. You can specify multiple criteria by using the **and** comparison operator. Property values that aren't expressed as an integer must be enclosed in quotation marks (").
Identity	Optional	Microsoft.Exchange.Data.QueueViewer.QueueIdentity	The *Identity* parameter specifies the queue. Valid input for this parameter uses the syntax *Server*

			Queue or *Queue*, for example, `Mailbox01 \contoso.com` or `Unreachable`. For details about queue identity, see the "Queue identity" section in Use the Exchange Management Shell to manage queues.
Include	Optional	Microsoft.Exchange.Data.QueueViewerIncludesAndExcludes	The *Include* parameter specifies the types of queues you want to include the results. Valid values for this parameter are: • Internal • External • A valid queue `DeliveryType` value. For details, see the NextHopSolutionKey section in Queues.
IncludeBookmark	Optional	System.Boolean	The *IncludeBookmark* parameter specifies whether to include the bookmark object when the query results are displayed. The *IncludeBookmark* parameter is valid when it's used with the *BookmarkObject* or

			BookmarkIndex parameters. If you don't specify a value for the *IncludeBookmark* parameter, the default value of `$true` is used.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
ReturnPageInfo	Optional	System.Boolean	The *ReturnPageInfo* parameter is a hidden parameter. Use it to return information about the total number of results and the index of the first object of the current page. The default value is `$false`.
SearchForward	Optional	System.Boolean	The *SearchForward* parameter specifies whether to search forward or backward in the result set. The default value is `$true`. This value causes the result page to be

			calculated forward from either the start of the result set or forward from a bookmark if specified.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can use the *Server* parameter and the *Filter* parameter in the same command. You can't use the *Server* parameter and the *Identity* parameter in the same command.
SortOrder	Optional	Microsoft.Exchange.Data.QueueViewerSortOrderEntry[]	The *SortOrder* parameter specifies an array of message properties used to control the sort order of the result set. Separate each property by using a

			comma. Prepend a plus sign (+) symbol to the beginning of the property name to display the results in ascending order. Prepend a minus sign (-) symbol to the beginning of the property name to display the results in descending order. If you don't specify a sort order, the result set is displayed in ascending order by **QueueIdentity**.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Resume-Queue

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Resume-Queue** cmdlet to restart processing for a suspended queue on a Mailbox server or an Edge Transport server.

```
Resume-Queue -Identity <QueueIdentity> <COMMON PARAMETERS>

Resume-Queue -Filter <String> [-Server <ServerIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resumes processing of all queues where the **NextHopDomain** is Fourthcoffee.com on the server Server1.contoso.com.

```
Resume-Queue -Server Server1.contoso.com -Filter
{NextHopDomain -eq "Fourthcoffee.com"}
```

Detailed Description

If you use the *Identity* parameter, the queue is resumed only if that identity matches a single queue. If the identity matches more than one queue, you receive an error. To resume more than one queue in a single operation, you must use the *Filter* parameter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Filter	Required	System.String	The *Filter* parameter specifies one or more queues by using OPath filter syntax. The OPath filter includes a queue property name followed by a comparison operator and value, for example, `{NextHopDomain -eq "contoso.com"}`. For

			details about filterable queue properties and comparison operators, see Queue filters and Use the Exchange Management Shell to manage queues. You can specify multiple criteria by using the **and** comparison operator. Property values that aren't expressed as an integer must be enclosed in quotation marks (").
Identity	Required	Microsoft.Exchange.Data.QueueViewer.QueueIdentity	The *Identity* parameter specifies the queue. Valid input for this parameter uses the syntax *Server**Queue* or *Queue*, for example, `Mailbox01\contoso.com` or `Unreachable`. For details about queue identity, see the "Queue identity" section in Use the Exchange Management Shell to manage queues.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can use the *Server* parameter and the *Filter* parameter in the same command. You can't use the *Server* parameter and the *Identity* parameter in the same command.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Retry-Queue

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Retry-Queue** cmdlet to force a connection attempt for a queue on a Mailbox server or an Edge Transport server.

```
Retry-Queue -Identity <QueueIdentity> <COMMON PARAMETERS>

Retry-Queue -Filter <String> [-Server <ServerIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-Resubmit <$true |
$false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example forces a connection attempt for all queues that meet the following criteria:

- The queues are holding messages for the domain contoso.com.
- The queues have a status of `Retry`.
- The queues are located on the server on which the command is executed.

```
Retry-Queue -Filter {NextHopDomain -eq "contoso.com" -and
Status -eq "Retry"}
```

Detailed Description

The **Retry-Queue** cmdlet forces a connection attempt for a queue that has a status of Retry. The cmdlet establishes a connection to the next hop if possible. If a connection isn't established, a new retry time is set. To use this command to retry delivery of messages in the Unreachable queue, you must include the *Resubmit* parameter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Filter	Required	System.String	The *Filter* parameter specifies one or more queues by using OPath filter syntax. The OPath filter includes a queue property name followed by a comparison operator and value, for example, `{NextHopDomain -eq "contoso.com"}`. For details about filterable queue properties and comparison operators, see Queue filters and Use

			the Exchange Management Shell to manage queues. You can specify multiple criteria by using the **and** comparison operator. Property values that aren't expressed as an integer must be enclosed in quotation marks (").
Identity	Required	Microsoft.Exchange.Data.QueueViewer.QueueIdentity	The *Identity* parameter specifies the queue. Valid input for this parameter uses the syntax *Server**Queue* or *Queue*, for example, `Mailbox01\contoso.com` or `Unreachable`. For details about queue identity, see the "Queue identity" section in Use the Exchange Management Shell to manage queues.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
Resubmit	Optional	System.Boolean	The *Resubmit* parameter specifies whether the queue contents should be resubmitted to the categorizer before a connection is established. Valid input for this parameter is `$true` or `$false`.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can use the *Server* parameter and the *Filter* parameter in the same command. You can't use the *Server* parameter and the *Identity* parameter in

			the same command.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Suspend-Queue

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Suspend-Queue** cmdlet to stop processing for a queue on a Mailbox server or an Edge Transport server.

```
Suspend-Queue -Identity <QueueIdentity> <COMMON PARAMETERS>
```

```
Suspend-Queue -Filter <String> [-Server <ServerIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example suspends processing on all queues holding messages for delivery to the domain contoso.com and that currently have a status of `Retry`.

```
Suspend-Queue -Filter {NextHopDomain -eq "contoso.com" -and
Status -eq "Retry"}
```

EXAMPLE 2

This example suspends processing on all queues on the server Server1.contoso.com that have more than 100 messages in the queue.

```
Suspend-Queue -Server Server1.contoso.com -Filter
{MessageCount -gt 100}
```

Detailed Description

The **Suspend-Queue** cmdlet stops processing on a queue that has a status of Active or Retry. Messages being processed are delivered, but no additional messages leave the queue. When you use the *Identity* parameter, the queue is suspended only if the identity matches a single queue. If the identity matches more than one queue, you receive an error. To suspend more than one queue in a single operation, you must use the *Filter* parameter.

For instructions on how to resume suspended queues, see Resume-Queue.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Filter	Required	System.String	The *Filter* parameter specifies one or more

			queues by using OPath filter syntax. The OPath filter includes a queue property name followed by a comparison operator and value, for example, `{NextHopDomain -eq "contoso.com"}`. For details about filterable queue properties and comparison operators, see Queue filters and Use the Exchange Management Shell to manage queues. You can specify multiple criteria by using the **and** comparison operator. Property values that aren't expressed as an integer must be enclosed in quotation marks (").
Identity	Required	Microsoft.Exchange.Data.QueueViewer.QueueIdentity	The *Identity* parameter specifies the queue. Valid input for this parameter uses the syntax *Server**Queue* or *Queue*, for example, `Mailbox01\contoso.com` or `Unreachable`. For details about queue identity, see the "Queue identity" section in Use the

			Exchange Management Shell to manage queues.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can use the *Server* parameter and the *Filter* parameter in the same command. You can't use

			the *Server* parameter and the *Identity* parameter in the same command.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-QueueDigest

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-QueueDigest** cmdlet to view information about message delivery queues across

database availability groups (DAGs) Active Directory sites, or Active Directory forests in your organization.

Note:
By default, the **Get-QueueDigest** cmdlet displays delivery queues that contain ten or more messages, and the results are between one and two minutes old. For instructions on how to change these default values, see Configure Get-QueueDigest.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-QueueDigest -Dag <MultiValuedProperty> <COMMON PARAMETERS>

Get-QueueDigest -Server <MultiValuedProperty> <COMMON PARAMETERS>

Get-QueueDigest -Site <MultiValuedProperty> <COMMON PARAMETERS>

Get-QueueDigest -Forest <SwitchParameter> <COMMON PARAMETERS>

COMMON PARAMETERS: [-DetailsLevel <None | Normal | Verbose>] [-Filter
<String>] [-GroupBy <NextHopDomain | NextHopCategory | NextHopKey |
DeliveryType | Status | RiskLevel | LastError | ServerName |
OutboundIPPool>] [-IncludeE14Servers <SwitchParameter>] [-Mtrt
<SwitchParameter>] [-ResultSize <Unlimited>] [-Timeout <EnhancedTimeSpan>]
```

Examples

Example 1

This example returns information about all queues in the Active Directory forest.

```
Get-QueueDigest -Forest
```

Example 2

This example returns information about all queues in the DAG named DAG01.

```
Get-QueueDigest -Dag DAG01
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description

Dag	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *Dag* parameter filters the delivery queue results by DAG. You can specify any value that uniquely identifies the DAG. You can specify multiple values separated by commas. If the value contains spaces, enclose the value in quotation marks ("). You can't use the *Dag* parameter with the *Server*, *Site* or *Forest* parameters.
Forest	Required	System.Management.Automation.SwitchParameter	The *Forest* switch filters the delivery queue results by Active Directory forest. You don't need to specify a value with the *Forest* switch. You can't use the *Forest* switch with the *Server*, *Site* or *Dag* parameters.
Server	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *Server* parameter filters the delivery queue results by Exchange server. You can specify any value that uniquely identifies the server. You can specify multiple values separated by commas. If the value

			contains spaces, enclose the value in quotation marks ("). You can't use the *Server* parameter with the *Dag*, *Site* or *Forest* parameters.
Site	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *Site* parameter filters the delivery queue results by Active Directory site. You can specify any value that uniquely identifies the site. You can specify multiple sites separated by commas. You can't use the *Site* parameter with the *Server*, *Dag* or *Forest* parameters.
DetailsLevel	Optional	Microsoft.Exchange.Net.DiagnosticsAggregation.DetailsLevel	The *DetailsLevel* parameter specifies the level of detail to display in the results. Valid values for this parameter are `None`, `Normal` and `Verbose`. The default value is `Normal`.
Filter	Optional	System.String	The *Filter* parameter specifies one or more queues by using OPath filter syntax. The OPath filter includes a queue property name followed

			by a comparison operator and value, for example, `{NextHopDomain -eq "contoso.com"}`. For details about filterable queue properties and comparison operators, see Queue filters and Use the Exchange Management Shell to manage queues. You can specify multiple criteria by using the **and** comparison operator. Property values that aren't expressed as an integer must be enclosed in quotation marks (").
GroupBy	Optional	Microsoft.Exchange.Net.DiagnosticsAggregation.QueueDigestGroupBy	The *GroupedBy* parameter sorts the messages in the delivery queue results. Valid values for this parameter are: • `DeliveryType` • `LastError` • `NextHopCategory` • `NextHopDomain` • `NextHopKey` • `RiskLevel` • `Status` • `ServerName` • `OutboundIPPool` The default value is `NextHopDomain`.
IncludeE14Servers	Optional	System.Management.Automation.SwitchPar	This parameter is reserved for internal Microsoft use.

		ameter	
Mtrt	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter filters the delivery queue results by the number of messages in the queue. Valid input for this parameter is an integer. The default value is 1000. For example, if you specify the value 50, the command displays the 50 queues that contain the most messages.
Timeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *Timeout* parameter specifies the number of seconds before the operation times out. The default value is 10 seconds. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ReceiveConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ReceiveConnector** cmdlet to view the configuration information for a Receive connector.

```
Get-ReceiveConnector [-Server <ServerIdParameter>] <COMMON PARAMETERS>

Get-ReceiveConnector [-Identity <ReceiveConnectorIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example displays detailed configuration information for the Receive Connector for Contoso.com.

```
Get-ReceiveConnector "Receive Connector for Contoso.com" |
Format-List
```

EXAMPLE 2

This example lists all the Receive connectors on Hub1.

```
Get-ReceiveConnector -Server Hub1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Receive connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ReceiveConnectorIdParameter	Specifies the GUID or connector name that represents a specific Receive connector. You can also include the server name using the format *ServerName**ConnectorName*. You can omit the

			parameter label so that only the connector name or GUID is supplied.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	Specifies the name of the server to query when the command is run. Only Receive connectors configured on the server you specify are displayed.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ReceiveConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-ReceiveConnector** cmdlet to create a new Receive connector.

```
New-ReceiveConnector -Bindings <MultiValuedProperty> -RemoteIPRanges
<MultiValuedProperty> [-Custom <SwitchParameter>] <COMMON PARAMETERS>


New-ReceiveConnector -Bindings <MultiValuedProperty> -Internet
<SwitchParameter> <COMMON PARAMETERS>
```

```
New-ReceiveConnector -Internal <SwitchParameter> -RemoteIPRanges
<MultiValuedProperty> <COMMON PARAMETERS>


New-ReceiveConnector -Client <SwitchParameter> -RemoteIPRanges
<MultiValuedProperty> <COMMON PARAMETERS>


New-ReceiveConnector -Bindings <MultiValuedProperty> -Partner
<SwitchParameter> -RemoteIPRanges <MultiValuedProperty> <COMMON
PARAMETERS>


New-ReceiveConnector -Usage <Custom | Internet | Internal | Client |
Partner> [-Bindings <MultiValuedProperty>] [-RemoteIPRanges
<MultiValuedProperty>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-AdvertiseClientSettings <$true |
$false>] [-AuthMechanism <None | Tls | Integrated | BasicAuth |
BasicAuthRequireTLS | ExchangeServer | ExternalAuthoritative>] [-Banner
<String>] [-BinaryMimeEnabled <$true | $false>] [-Bindings
<MultiValuedProperty>] [-ChunkingEnabled <$true | $false>] [-Comment
<String>] [-Confirm [<SwitchParameter>]] [-ConnectionInactivityTimeout
<EnhancedTimeSpan>] [-ConnectionTimeout <EnhancedTimeSpan>] [-
DefaultDomain <AcceptedDomainIdParameter>] [-
DeliveryStatusNotificationEnabled <$true | $false>] [-DomainController
<Fqdn>] [-DomainSecureEnabled <$true | $false>] [-EightBitMimeEnabled
<$true | $false>] [-EnableAuthGSSAPI <$true | $false>] [-Enabled <$true |
$false>] [-EnhancedStatusCodesEnabled <$true | $false>] [-
ExtendedProtectionPolicy <None | Allow | Require>] [-Fqdn <Fqdn>] [-
LiveCredentialEnabled <$true | $false>] [-LongAddressesEnabled <$true |
$false>] [-MaxAcknowledgementDelay <EnhancedTimeSpan>] [-MaxHeaderSize
<ByteQuantifiedSize>] [-MaxHopCount <Int32>] [-MaxInboundConnection
<Unlimited>] [-MaxInboundConnectionPercentagePerSource <Int32>] [-
MaxInboundConnectionPerSource <Unlimited>] [-MaxLocalHopCount <Int32>] [-
MaxLogonFailures <Int32>] [-MaxMessageSize <ByteQuantifiedSize>] [-
MaxProtocolErrors <Unlimited>] [-MaxRecipientsPerMessage <Int32>] [-
MessageRateLimit <Unlimited>] [-MessageRateSource <None | IPAddress | User
| All>] [-OrarEnabled <$true | $false>] [-PermissionGroups <None |
AnonymousUsers | ExchangeUsers | ExchangeServers | ExchangeLegacyServers |
Partners | Custom>] [-PipeliningEnabled <$true | $false>] [-
ProtocolLoggingLevel <None | Verbose>] [-ProxyEnabled <$true | $false>] [-
RemoteIPRanges <MultiValuedProperty>] [-RequireEHLODomain <$true |
$false>] [-RequireTLS <$true | $false>] [-Server <ServerIdParameter>] [-
ServiceDiscoveryFqdn <Fqdn>] [-SizeEnabled <Disabled | Enabled |
EnabledWithoutValue>] [-SuppressXAnonymousTls <$true | $false>] [-
TarpitInterval <EnhancedTimeSpan>] [-TlsCertificateName
<SmtpX509Identifier>] [-TlsDomainCapabilities <MultiValuedProperty>] [-
TransportRole <None | Cafe | Mailbox | ClientAccess | UnifiedMessaging |
HubTransport | Edge | All | Monitoring | CentralAdmin |
CentralAdminDatabase | DomainController | WindowsDeploymentServer |
ProvisionedServer | LanguagePacks | FrontendTransport | CafeArray |
FfoWebService | OSP | ARR | ManagementFrontEnd | ManagementBackEnd | SCOM
| CentralAdminFrontEnd | NAT | DHCP>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the custom Receive connector Test with the following properties:

- It listens for incoming SMTP connections on the IP address 10.10.1.1 and port 25.
- It accepts incoming SMTP connections only from the IP range 192.168.0.1-192.168.0.24

```
New-ReceiveConnector -Name Test -Usage Custom -Bindings
10.10.1.1:25 -RemoteIPRanges 192.168.0.1-192.168.0.24
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Receive connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Bindings	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *Bindings* parameter specifies the local IP address and TCP port number used by the Receive connector to listen for inbound messages. Valid syntax for this parameter is *<IP Address>*:*<TCP Port>*, such as `192.168.1.1:25`. The IP address `0.0.0.0` indicates that the Receive connector uses all IP addresses configured on all network adapters to listen for inbound messages. **Note:** You must specify a local IP address that's valid for the Mailbox server or Edge server on which the Receive connector is located. If you specify an invalid local IP address, the Microsoft Exchange Transport service may fail

			to start when the service is restarted. You must specify a value for the *Bindings* parameter when the following parameters are specified: • *Internet* or *Usage* parameter with a value of `Internet` • *Partner* or *Usage* parameter with a value of `Partner` • *Custom* or *Usage* parameter with a value of `Custom` The values that you specify using the *Bindings* parameter must satisfy one of the following requirements for uniqueness: • You can specify a unique combination of IP address and TCP port that doesn't conflict with the IP address or TCP port used in the *Bindings* parameter of another Receive connector on the server. • You can use an existing combination of IP

			address and TCP port, but use the *RemoteIPRanges* parameter to restrict the remote servers serviced by the Receive connector. However, when you use this command to create a Receive connector, you can only use the *RemoteIPRanges* parameter and the *Bindings* parameter together when the following parameters are specified: *Custom* or *Usage* parameter with a value of `Custom`. *Partner* or *Usage* parameter with a value of `Partner`. You can't specify a value for the *Bindings* parameter with this command when the following parameters are specified: • *Client* or *Usage* parameter with a value of `Client`. The default value of the *Bindings* parameter is

			`0.0.0.0:587`. This value indicates that the connector accepts connections on TCP port 587 on all IP addresses configured on all network adapters in the server. • *Internal* or *Usage* parameter with a value of `Internal`. The default value of the *Bindings* parameter is `0.0.0.0:25`. This value indicates that the connector accepts connections on TCP port 25 on all IP addresses configured on all network adapters in the server.
Client	Required	System.Management.Automation.SwitchParameter	The *Client* parameter specifies the `Client` usage type. The usage type specifies the default permission groups and authentication methods assigned to the Receive connector. If you use the *Client* parameter, you can't use any of the following parameters: • *Internal*

			• *Internet* • *Partner* • *Custom* • *Usage* If you specify the *Client* parameter, you must specify a value for the *RemoteIPRanges* parameter. If you don't specify a value for a required parameter, this command prompts you so that it may continue. For more information about Receive connector usage types, permission groups, and authentication methods, see Receive connectors.
Internal	Required	System.Management.Automation.SwitchParameter	The *Internal* parameter specifies the `Internal` usage type. The usage type specifies the default permission groups and authentication methods assigned to the Receive connector. If you use the *Internal* parameter, you can't use any of the following parameters: • *Client* • *Internet*

			• *Partner* • *Custom* • *Usage* If you specify the *Internal* parameter, you must specify a value for the *RemoteIPRanges* parameter. If you don't specify a value for a required parameter, the command prompts you so that it may continue. For more information about Receive connector usage types, permission groups, and authentication methods, see Receive connectors.
Internet	Required	System.Management.Automation.SwitchParameter	The *Internet* parameter specifies the `Internet` usage type. The usage type specifies the default permission groups and authentication methods assigned to the Receive connector. If you use the *Internet* parameter, you can't use any of the following parameters: • *Client* • *Internal* • *Partner*

			• *Custom* • *Usage* If you specify the *Internet* parameter, you must specify a value for the *Bindings* parameter. If you don't provide a value for a required parameter, this command prompts you so that it may continue. For more information about Receive connector usage types, permission groups, and authentication methods, see Receive connectors.
Name	Required	System.String	The *Name* parameter specifies the administrator-supplied name of the connector. Enter the *Name* parameter as a string, for example: `"New Receive Connector"`.
Partner	Required	System.Management.Automation.SwitchParameter	The *Partner* parameter specifies the `Partner` usage type. The usage type specifies the default permission groups and authentication methods assigned to the Receive connector. If you use the

			Partner parameter, you can't use any of the following parameters: • *Client* • *Internal* • *Internet* • *Custom* • *Usage* If you specify the *Partner* parameter, you must specify a value for the following parameters: • *Bindings* • *RemoteIPRanges* If you don't provide a value for a required parameter, this command prompts you so that it may continue.
RemoteIPRanges	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *RemoteIPRanges* parameter specifies the remote IP addresses from which this connector accepts messages. Valid syntax for this parameter is *<Starting IP Address>-<Ending IP Address>*, such as `192.168.1.1-192.168.1.10`. You can specify multiple IP address ranges separated by commas.

			You must specify a value for the *RemoteIPRanges* parameter when the following parameters are specified: • *Client* or *Usage* parameter with a value of `Client` • *Internal* or *Usage* parameter with a value of `Internal` • *Partner* or *Usage* parameter with a value of `Partner` • *Custom* or *Usage* parameter with a value of `Custom` Multiple Receive connectors on the same server can have overlapping remote IP address ranges as long as one IP address range is completely overlapped by another IP address. When remote IP address ranges overlap, the remote IP address range with the most specific match to the IP address of the connecting server is used. The default value of the

			RemoteIPRanges parameter for the `Internet` usage type is `0.0.0.0-255.255.255.255`. This value indicates that the connector accepts connections from all remote IP addresses.
Usage	Required	Microsoft.Exchange.Management.SystemConfigurationTasks.NewReceiveConnector+UsageType	The *Usage* parameter specifies the default permission groups and authentication methods assigned to the Receive connector. The valid values for the *Usage* parameter are as follows: `Client`, `Custom`, `Internal`, `Internet`, and `Partner`. If you specify a value for the *Usage* parameter, you can't use any of the following parameters: • *Client* • *Internal* • *Internet* • *Partner* • *Custom* A value for the *Bindings* parameter is required if you specify any of the following values for the *Usage* parameter:

			• `Internet` • `Partner` • `Custom` A value for the *RemoteIPRanges* parameter is required if you specify any of the following values for the *Usage* parameter: • `Client` • `Internal` • `Partner` • `Custom` If you don't specify a value for a required parameter, the command ends unsuccessfully. This command won't prompt you for the missing required parameters. For more information about Receive connector usage types, permission groups, and authentication methods, see Receive connectors.
AdvertiseClientSettings	Optional	System.Boolean	The *AdvertiseClientSettings* parameter specifies whether the SMTP server name, port number, and authentication settings are displayed in the Outlook Web App, accessed from **Settings > Options >**

			Account > my Account > Settings for POP or IMAP access. The default value is `$false`.
AuthMechanism	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthMechanisms	The *AuthMechanism* parameter specifies the advertised and accepted authentication mechanisms. The authentication options are `None`, `Tls`, `Integrated`, `BasicAuth`, `BasicAuthRequireTLS`, `ExchangeServer`, and `ExternalAuthoritative`. You can enter multiple values for the *AuthMechanism* parameter by separating the values with commas. If the *RequireTLS* parameter is set to `$true`, the *AuthMechanism* parameter must be set to `Tls`. If you set the *AuthMechanism* parameter to `BasicAuthRequireTLS`, you must also select `BasicAuth` and `Tls`. The *AuthMechanism*

			parameter value `ExternalAuthoritative` may only coexist with the value `Tls`. If you set the *AuthMechanism* parameter to `ExternalAuthoritative`, the *PermissionGroups* parameter must also have the value `ExchangeServers`.
Banner	Optional	System.String	The *Banner* parameter specifies an SMTP 220 banner and overrides the default SMTP 220 banner. When the value of the *Banner* parameter is blank, the default SMTP banner is the following: `220 <Servername> Microsoft ESMTP MAIL service ready at <RegionalDay-Date-24HourTimeFormat> <RegionalTimeZoneOffset>` When you specify a value for the *Banner* parameter, you must use the following syntax: `220 <RemainingBannerText>`. 220 is the default Service ready SMTP response code as defined in RFC 2821.

BinaryMimeEnabled	Optional	System.Boolean	The *BinaryMimeEnabled* parameter specifies whether the BINARYMIME EHLO keyword is advertised in the EHLO response to the remote server and is available for use. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. When the *BinaryMimeEnabled* parameter is set to `$true`, the BINARYMIME EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *BinaryMimeEnabled* parameter is set to `$false`, the BINARYMIME EHLO keyword isn't advertised in the EHLO response to the remote server and is disabled.
ChunkingEnabled	Optional	System.Boolean	The *ChunkingEnabled* parameter specifies whether the CHUNKING EHLO keyword is advertised in the EHLO response to the remote server and is available for

			use. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. When the *ChunkingEnabled* parameter is set to `$true`, the CHUNKING EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *ChunkingEnabled* parameter is set to `$false`, the CHUNKING EHLO keyword isn't advertised in the EHLO response to the remote server and is disabled.
Comment	Optional	System.String	The *Comment* parameter specifies an optional comment. If you specify a value that contains spaces, enclose the value in quotation marks ("), for example: `"This is an admin note"`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before

			processing continues. You don't have to specify a value with the *Confirm* switch.
ConnectionInactivityTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConnectionInactivityTimeout* parameter specifies the maximum amount of idle time before a connection to a Receive connector is closed. The default value for a Receive connector configured on a Mailbox server is 5 minutes. The default value for a Receive connector configured on an Edge server is 1 minute. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The value specified by the *ConnectionTimeout* parameter must be greater than the value specified by the *ConnectionInactivityTimeout* parameter. The valid input range for either parameter is `00:00:01` to

			1.00:00:00.
ConnectionTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConnectionTimeout* parameter specifies the maximum time that a connection can remain open, even if the connection is actively transmitting data. The default value for a Receive connector configured on a Mailbox server is 10 minutes. The default value for a Receive connector configured on an Edge server is 5 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify a connection time-out of 5 minutes, enter 00.00:05:00. The value specified by the *ConnectionTimeout* parameter must be greater than the value specified by the *ConnectionInactivityTimeout* parameter. The valid input range for either

			parameter is from `00:00:01` through `1.00:00:00`.
Custom	Optional	System.Management.Automation.SwitchParameter	The *Custom* parameter specifies the `Custom` usage type. The usage type specifies the default permission groups and authentication methods assigned to the Receive connector. If you use the *Custom* parameter, you can't use any of the following parameters: • *Client* • *Internal* • *Internet* • *Partner* • *Usage* If you specify the *Custom* parameter, you must specify a value for the following parameters: • *Bindings* • *RemoteIPRanges* If you don't provide a value for a required parameter, this command prompts you so that it may continue.
DefaultDomain	Optional	Microsoft.Exchange.Co	The *DefaultDomain*

		nfiguration.Tasks.AcceptedDomainIdParameter	parameter specifies the domain name to append to values that are submitted to `MAIL FROM` or `RCPT TO` in the message envelope by a sending server if no domain name is provided.
DeliveryStatusNotificationEnabled	Optional	System.Boolean	The *DeliveryStatusNotificationEnabled* parameter specifies whether the delivery status notification (DSN) EHLO keyword is advertised in the EHLO response to the remote server and is available for use. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. When the *DeliveryStatusNotificationEnabled* parameter is set to `$true`, the DSN EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *DeliveryStatusNotificationEnabled* parameter is set to `$false`, the DSN EHLO

			keyword isn't advertised in the EHLO response to the remote server and is disabled. The DSN extension to extended SMTP (ESMTP) provides enhanced DSN functionality specified in RFC 1891.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
DomainSecureEnabled	Optional	System.Boolean	The *DomainSecureEnabled* parameter enables mutual Transport Layer Security (TLS) authentication for

			the domains serviced by this Receive connector. Mutual TLS authentication functions correctly only if the following conditions are true: • The value of the *DomainSecureEnabled* parameter is `$true`. • The *AuthMechanism* parameter contains the value `Tls` and doesn't contain the value `ExternalAuthoritative`. • The value of the *AuthMechanism* parameter contains `Tls`. • The *TLSReceiveDomainSecureList* parameter in the **Get-TransportConfig** command contains at least one domain serviced by this Receive connector. The wildcard character (*) isn't supported in domains that are configured for mutual TLS authentication. The same domain must also be defined on the corresponding Send

			connector, and in the value of the *TLSSendDomainSecureList* parameter in the **Get-TransportConfig** command. The default value for the *DomainSecureEnabled* parameter is `$false` for the following types of Receive connectors: • All Receive connectors defined in the Transport service on a Mailbox server • User-created Receive connectors defined on an Edge server The default value for the *DomainSecureEnabled* parameter is `$true` for default Receive connectors defined on an Edge server.
EightBitMimeEnabled	Optional	System.Boolean	The *EightBitMimeEnabled* parameter specifies whether the 8BITMIME EHLO keyword is advertised in the EHLO response to the remote server and is available for use. Valid values for this

			parameter are `$true` or `$false`. The default value is `$true`. When the *EightBitMimeEnabled* parameter is set to `$true`, the 8BITMIME EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *EightBitMimeEnabled* parameter is set to `$false`, the 8BITMIME EHLO keyword isn't advertised in the EHLO response to the remote server and is disabled.
EnableAuthGSSAPI	Optional	System.Boolean	The *EnableAuthGSSAPI* parameter specifies the advertisement of the Generic Security Services application programming interface (GSSAPI) authentication method, when Integrated Windows authentication is enabled on this connector. If the *AuthMechanism* parameter contains `Integrated`, and the *EnableAuthGSSAPI* parameter is set to `$true`, the AUTH GSSAPI NTLM

			keyword is advertised in the EHLO response of the Receive connector. Clients may use Kerberos or NTLM to authenticate with the Receive connector. If the *AuthMechanism* parameter contains `Integrated`, and the *EnableAuthGSSAPI* parameter is set to `$false`, the AUTH NTLM keyword is advertised in the EHLO response of the Receive connector. Clients may use only NTLM to authenticate with the Receive connector. If you have Internet Information Services (IIS) messaging servers that authenticate with this Receive connector, you should set the value of the *EnableAuthGSSAPI* parameter to `$false`. Authentication with computers running Microsoft Exchange Server 2003 aren't affected by the value of the *EnableAuthGSSAPI* parameter. Exchange

			2003 servers use the authentication methods advertised in the X-EXPS keyword. The X-EXPS keyword is advertised in the EHLO response of the Receive connector when the *AuthMechanism* parameter contains `ExchangeServer`. The valid values for this parameter is `$true` or `$false`. The default value is `$false`. By default, the *EnableAuthGSSAPI* parameter is set to `$true` only on the default Receive connector Client *<Server Name>* created only in the Transport service on a Mailbox server.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether to enable the connector. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. Use the *Enabled* parameter to enable or disable the connector.
EnhancedStatusCodes	Optional	System.Boolean	The

Enabled			*EnhancedStatusCodesEnabled* parameter specifies whether the ENHANCEDSTATUSCODES EHLO keyword is advertised in the EHLO response to the remote server and is available for use. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. When the *EnhancedStatusCodesEnabled* parameter is set to `$true`, the ENHANCEDSTATUSCODES EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *EnhancedStatusCodesEnabled* parameter is set to `$false`, the ENHANCEDSTATUSCODES EHLO keyword isn't advertised in the EHLO response to the remote server and is disabled. The ENHANCEDSTATUSCODES extension provides enhanced error and status information in delivery

			status notification messages sent to remote servers.
ExtendedProtectionPolicy	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionPolicySetting	The *ExtendedProtectionPolicy* parameter specifies how Extended Protection for Authentication is implemented on this Receive connector. By default, this parameter is set to `None`. The *ExtendedProtectionPolicy* parameter may have the following values: • `None` Extended Protection for Authentication isn't used. • `Allow` Extended Protection for Authentication is used only if the connecting host supports it. Otherwise, connections are established without Extended Protection for Authentication. • `Require` Extended Protection for Authentication is required for all incoming connections to this Receive connector. If the connecting host doesn't support Extended Protection for Authentication, the

			connection is rejected. Extended Protection for Authentication enhances the protection and handling of credentials when authenticating network connections using Integrated Windows authentication. Integrated Windows authentication is also known as NTLM. We strongly recommend that you use Extended Protection for Authentication if you are using Integrated Windows authentication.
Fqdn	Optional	Microsoft.Exchange.Data.Fqdn	The *Fqdn* parameter specifies the FQDN used as the destination server for connected messaging servers that use the Receive connector to send incoming messages. The value of this parameter is displayed to connected messaging servers whenever a destination server name is required, as in the following examples: • In the default SMTP banner of the Receive

			connector • In the EHLO/HELO response of the Receive connector • In the most recent `Received` header field in the incoming message when the message enters the Transport service of a Mailbox server or an Edge server • During TLS authentication The default value of the *Fqdn* parameter is the FQDN of the Mailbox server or Edge server that contains the Receive connector. **Note:** Don't modify the FQDN value on the default Receive connector Default <*Server Name*> that's automatically created in the Transport service of a Mailbox server. If you have multiple Mailbox servers in your Exchange organization and you change the FQDN value on the Default <*Server Name*> Receive connector, internal mail flow between Mailbox servers will fail.
LiveCredentialEnabled	Optional	System.Boolean	This parameter is reserved

			for internal Microsoft use.
LongAddressesEnabled	Optional	System.Boolean	The *LongAddressesEnabled* parameter specifies whether the Receive connector accepts long X.400 email addresses. The X.400 email addresses are encapsulated in SMTP email addresses by using the Internet Mail Connector Encapsulated Address (IMCEA) encapsulation method. When the value of this parameter is `$false`, the maximum length for a complete SMTP email address is 571 characters. When the value of this parameter is `$true`, the following changes are made: • The XLONGADDR keyword is advertised in the EHLO response of the Receive connector. • The accepted line length of an SMTP session is increased to 8,000 characters. • Valid long addresses are

			accepted by the MAIL FROM and RCPT TO SMTP commands. Therefore, X.400 email addresses can be up to 1,860 characters long after IMCEA encapsulation. The valid values for this parameter are `$true` or `$false`. The default value is `$false`. You can modify this parameter only on Receive connectors configured in the Transport service on a Mailbox server.
MaxAcknowledgementDelay	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter isn't used by Microsoft Exchange Server 2013. It's only used by Microsoft Exchange 2010 servers in a coexistence environment. The *MaxAcknowledgementDelay* parameter specifies the maximum period the transport server delays acknowledgement until it verifies that the message has been successfully delivered to all recipients.

			When receiving messages from a host that doesn't support shadow redundancy, an Exchange Server 2010 transport server will delay issuing an acknowledgement until it verifies that the message has been successfully delivered to all recipients. However, if it takes too long to verify successful delivery, the transport server will time out and issue an acknowledgement anyway. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The default value is 30 seconds.
MaxHeaderSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *MaxHeaderSize* parameter specifies in bytes the maximum size of the SMTP message header that the Receive connector accepts before it closes the connection. The default value is 65536

			bytes. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 1 through 2147483647 bytes.
MaxHopCount	Optional	System.Int32	The *MaxHopCount* parameter specifies the maximum number of hops that a message can take before the message is rejected by the Receive connector. The maximum number of hops is determined by the number of `Received` header fields that exist in a submitted message. The default value is 30. The valid input range for this parameter is from 1 through 500.
MaxInboundConnecti	Optional	Microsoft.Exchange.Da	The

on		ta.Unlimited	*MaxInboundConnection* parameter specifies the maximum number of inbound connections that this Receive connector serves at the same time. The default value is 5000. The valid input range for this parameter is from 1 through 2147483647. To disable the inbound connection limit on a Receive connector, enter a value of `unlimited`.
MaxInboundConnectionPercentagePerSource	Optional	System.Int32	The *MaxInboundConnectionPercentagePerSource* parameter specifies the maximum number of connections that a Receive connector serves at the same time from a single IP address, expressed as the percentage of available remaining connections on a Receive connector. Enter the value as an integer without the percent sign (%). The default value is 2 percent. The valid input range for this parameter is 1 to 100.

MaxInboundConnectionPerSource	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxInboundConnectionPerSource* parameter specifies the maximum number of connections that this Receive connector serves at the same time from a single IP address. The default value is 100. The valid input range for this parameter is from 1 through 10000. To disable the inbound connection per source limit on a Receive connector, enter a value of `unlimited`.
MaxLocalHopCount	Optional	System.Int32	The *MaxLocalHopCount* parameter specifies the maximum number of local hops that a message can take before the message is rejected by the Receive connector. The maximum number of local hops is determined by the number of `Received` headers with local server addresses in a submitted message. The default value is 8. The valid input range for this parameter

			is 0 to 50. When you specify a value of 0, the message is never rejected based on the number of local hops.
MaxLogonFailures	Optional	System.Int32	The *MaxLogonFailures* parameter specifies the number of logon failures that the Receive connector retries before it closes the connection. The default value is 3. The valid input range for this parameter is from 0 through 10. When you specify a value of 0, the connection is never closed because of logon failures.
MaxMessageSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *MaxMessageSize* parameter specifies the maximum size of a message. The default value is 25 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes.

			The valid input range for this parameter is from 65536 through 2147483647 bytes.
MaxProtocolErrors	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxProtocolErrors* parameter specifies the maximum number of SMTP protocol errors that the Receive connector accepts before closing the connection. The default value is 5. The valid input range for this parameter is from 0 through 2147483647. When you specify a value of `unlimited`, a connection is never closed because of protocol errors.
MaxRecipientsPerMessage	Optional	System.Int32	The *MaxRecipientsPerMessage* parameter specifies the maximum number of recipients per message that the Receive connector accepts before closing the connection. The default value is 200. The valid input range for this parameter is 1 to 512000.
MessageRateLimit	Optional	Microsoft.Exchange.Da	The *MessageRateLimit*

		ta.Unlimited	parameter specifies the maximum number of messages that can be sent by a single client IP address per minute. The default value for a Receive connector configured on a Mailbox server is `unlimited`. The default value for a Receive connector configured on an Edge server is 600 messages per minute. The valid input range for this parameter is 1 to 2147483647. To remove the message rate limit on a Receive connector, enter a value of `unlimited`.
MessageRateSource	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MessageRateSourceFlags	The *MessageRateSource* parameter specifies how the message submission rate is calculated. It can have one of the following values: • `None`: The message submission rate isn't calculated. • `User`: The message submission rate is calculated for sending users (specified with the MAIL FROM SMTP command). • `IPAddress`: The message

			submission rate is calculated for sending hosts. • `All`: The message submission rate is calculated for both sending users and sending hosts.
OrarEnabled	Optional	System.Boolean	The *OrarEnabled* parameter enables the Originator Requested Alternate Recipient (ORAR). When the value of this parameter is `$false`, ORAR isn't supported. When the value of this parameter is `$true`, ORAR is supported by advertising the XORAR keyword in the EHLO response of the Receive connector. The actual ORAR information is transmitted in the RCPT TO SMTP command. The valid values for this parameter are `$true` or `$false`. The default value is `$false`. If the email address specified in the ORAR information is a long X.400 email address, the *LongAddressesEnabled* parameter must be `$true`.

PermissionGroups	Optional	Microsoft.Exchange.Data.PermissionGroups	The *PermissionGroups* parameter specifies the groups or roles that can submit messages to the Receive connector and the permissions assigned to those groups. A permission group is a predefined set of permissions granted to well-known security principals. The valid values for this parameter are as follows: `None`, `AnonymousUsers`, `ExchangeUsers`, `ExchangeServers`, `ExchangeLegacyServers` `Partners`, and `Custom`. The default permission groups assigned to a Receive connector depend on the connector usage type that was specified by the *Usage* parameter when the Receive connector was created. For more information about Receive connector usage types, see Receive connectors.
PipeliningEnabled	Optional	System.Boolean	The *PipeliningEnabled*

			parameter specifies whether the PIPELINING EHLO keyword is advertised in the EHLO response to the remote server and is available for use. Valid values for this parameter are `$true` or `$false`. When the *PipeliningEnabled* parameter is set to `$true`, the PIPELINING EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *PipeliningEnabled* parameter is set to `$false`, the PIPELINING EHLO keyword isn't advertised in the EHLO response to the remote server and is disabled. The PIPELINING extension enables the remote server to send requests without waiting for a response from this Receive connector. The default value is `$true`.
ProtocolLoggingLevel	Optional	Microsoft.Exchange.Data.ProtocolLoggingLev	The *ProtocolLoggingLevel* parameter specifies

		el	whether to enable or disable protocol logging for a Receive connector. A value of `Verbose` enables protocol logging for the connector. A value of `None` disables protocol logging for the connector. The default value is `None`. The location of the Receive connector protocol logs for all Receive connectors configured on a Mailbox server or an Edge server is specified by using the *ReceiveProtocolLogPath* parameter of the **Set-TransportService** cmdlet.
ProxyEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RequireEHLODomain	Optional	System.Boolean	The *RequireEHLODomain* parameter specifies whether the remote computer must provide a domain name in the EHLO handshake after the SMTP connection is established. Valid values for this parameter are `$true` or `$false`. The default value is `$false`. When the *RequireEHLODomain*

			parameter is set to `$true`, the remote computer must provide a domain name in the EHLO handshake after the SMTP connection is established. If the remote computer doesn't provide the domain name, the SMTP connection is closed.
RequireTLS	Optional	System.Boolean	The *RequireTLS* parameter specifies that all messages received by this connector require TLS transmission. Valid values for this parameter are `$true` or `$false`. The default value is `$false`. When the *RequireTLS* parameter is set to `$true`, all messages received by this connector require TLS transmission.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server on which the new Receive connector is created.
ServiceDiscoveryFqdn	Optional	Microsoft.Exchange.Data.Fqdn	The service discovery fully-qualified domain name (FQDN).
SizeEnabled	Optional	Microsoft.Exchange.Data.SizeMode	The *SizeEnabled* parameter specifies

			whether the SIZE SMTP extension is enabled. Valid values for this parameter are `Enabled`, `Disabled`, or `EnabledwithoutValue`. The default value is `Enabled`. When the *SizeEnabled* parameter is set to `Enabled`, the SIZE SMTP extension is enabled, and the maximum allowed message size value from the *MaxMessageSize* parameter is advertised in the EHLO banner. When the *SizeEnabled* parameter is set to `Disabled`, the SIZE SMTP extension isn't used, and the maximum allowed message size value is never disclosed to the remote server. When the *SizeEnabled* parameter is set to `EnabledWithoutValue`, the SIZE SMTP extension is enabled. However, the maximum allowed message size value from the *MaxMessageSize* parameter isn't advertised

			in the EHLO banner. This allows the message to bypass message size checks for authenticated connections between Mailbox servers. The SIZE SMTP extension is defined in RFC 1870. The SIZE SMTP extension enables the source server to declare the size of the inbound message to the target server. It also allows the target server to declare the maximum message size that it's allowed to accept to the sending server. If the advertised size of the inbound message exceeds the value in the *MaxMessageSize* parameter, the Receive connector responds to the remote server by using an error code and closes the connection.
SuppressXAnonymousTls	Optional	System.Boolean	The *SuppressXAnonymousTls* parameter specifies whether this Receive connector supports the standard TLS encryption

			for incoming connections. By default, all communications between servers is protected with TLS. However, if you need to disable TLS on a specific connection in your organization, you can create a specific Receive connector and set the *SuppressXAnonymousTls* parameter to `$true`. The default value is `$false`. Before you can set this parameter to `$true`, you must use the **Set-TransportService** cmdlet to set the *UseDownGradedExchangeServerAuth* parameter to `$true` on the server this Receive connector is configured.
TarpitInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *TarpitInterval* parameter specifies the amount of time to delay an SMTP response to a remote server that may be abusing the connection. Authenticated connections are never delayed in this

			manner. The default value is 5 seconds. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The valid input range for this parameter is from `00:00:00` through `00:10:00`. When you set the value to `00:00:00`, you disable the tarpit interval.
TlsCertificateName	Optional	Microsoft.Exchange.Data.SmtpX509Identifier	The *TlsCertificateName* parameter specifies the X.509 certificate to use with TLS sessions and secure mail. Valid input for this parameter is `[I]`*Issuer*`[S]`*Subject*. The *Issuer* value is found in the certificate's `Issuer` field, and the *Subject* value is found in the certificate's `Subject` field. You can find these values by running the **Get-ExchangeCertificate** cmdlet.
TlsDomainCapabilities	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *TlsDomainCapabilities* parameter specifies the

		y	capabilities this Receive connector will make available to specific hosts outside your organization. TLS with certificate validation is used to authenticate remote hosts before these capabilities are offered. To specify capabilities for a domain, use the following syntax: *<domain name 1>:<capability 1, capability 2,...,capability N>* If you're specifying capabilities for multiple domains, list the configuration for each domain in quotation marks ("), separated by commas. For example: "contoso.com:AcceptOorgProtocol","fabrikam.com:AcceptOorgProtocol,AcceptOorgHeader" You can configure the capabilities for non-TLS encrypted incoming connections using the special "NO-TLS" domain.

			You can configure the following capabilities for a domain: • `AcceptOorgProtocol` • `AcceptOorgHeader`
TransportRole	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ServerRole	The *TransportRole* parameter designates the server role associated with this connector. Types include `FrontendTransport` and `HubTransport`. Typically used to specify the server role when you host multiple server roles on a single computer.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ReceiveConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-ReceiveConnector** cmdlet to delete a Receive connector.

```
Remove-ReceiveConnector -Identity <ReceiveConnectorIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the Receive connector Contoso.com Receive Connector.

```
Remove-ReceiveConnector "Contoso.com Receive Connector"
```

Detailed Description

The **Remove-ReceiveConnector** cmdlet deletes the object and the configuration information for a Receive connector.

Caution:
Deleting a Receive connector may affect mail flow throughout the organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Receive connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ReceiveConnectorIdParameter	The *Identity* parameter specifies the GUID or connector name that represents a specific Receive connector. You can also include the server name by using the format *ServerName**ConnectorName*.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge

			Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ReceiveConnector

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ReceiveConnector** cmdlet to change an existing Receive connector.

```
Set-ReceiveConnector -Identity <ReceiveConnectorIdParameter> [-
AdvertiseClientSettings <$true | $false>] [-AuthMechanism <None | Tls |
Integrated | BasicAuth | BasicAuthRequireTLS | ExchangeServer |
ExternalAuthoritative>] [-Banner <String>] [-BareLinefeedRejectionEnabled
<$true | $false>] [-BinaryMimeEnabled <$true | $false>] [-Bindings
<MultiValuedProperty>] [-ChunkingEnabled <$true | $false>] [-Comment
<String>] [-Confirm [<SwitchParameter>]] [-ConnectionInactivityTimeout
<EnhancedTimeSpan>] [-ConnectionTimeout <EnhancedTimeSpan>] [-
DefaultDomain <AcceptedDomainIdParameter>] [-
DeliveryStatusNotificationEnabled <$true | $false>] [-DomainController
<Fqdn>] [-DomainSecureEnabled <$true | $false>] [-EightBitMimeEnabled
<$true | $false>] [-EnableAuthGSSAPI <$true | $false>] [-Enabled <$true |
$false>] [-EnhancedStatusCodesEnabled <$true | $false>] [-
ExtendedProtectionPolicy <None | Allow | Require>] [-Fqdn <Fqdn>] [-
LiveCredentialEnabled <$true | $false>] [-LongAddressesEnabled <$true |
$false>] [-MaxAcknowledgementDelay <EnhancedTimeSpan>] [-MaxHeaderSize
<ByteQuantifiedSize>] [-MaxHopCount <Int32>] [-MaxInboundConnection
<Unlimited>] [-MaxInboundConnectionPercentagePerSource <Int32>] [-
MaxInboundConnectionPerSource <Unlimited>] [-MaxLocalHopCount <Int32>] [-
MaxLogonFailures <Int32>] [-MaxMessageSize <ByteQuantifiedSize>] [-
MaxProtocolErrors <Unlimited>] [-MaxRecipientsPerMessage <Int32>] [-
MessageRateLimit <Unlimited>] [-MessageRateSource <None | IPAddress | User
| All>] [-Name <String>] [-OrarEnabled <$true | $false>] [-
PermissionGroups <None | AnonymousUsers | ExchangeUsers | ExchangeServers
| ExchangeLegacyServers | Partners | Custom>] [-PipeliningEnabled <$true |
$false>] [-ProtocolLoggingLevel <None | Verbose>] [-ProxyEnabled <$true |
$false>] [-RemoteIPRanges <MultiValuedProperty>] [-RequireEHLODomain
<$true | $false>] [-RequireTLS <$true | $false>] [-ServiceDiscoveryFqdn
<Fqdn>] [-SizeEnabled <Disabled | Enabled | EnabledWithoutValue>] [-
SmtpUtf8Enabled <$true | $false>] [-SuppressXAnonymousTls <$true |
$false>] [-TarpitInterval <EnhancedTimeSpan>] [-TlsCertificateName
<SmtpX509Identifier>] [-TlsDomainCapabilities <MultiValuedProperty>] [-
TransportRole <None | Cafe | Mailbox | ClientAccess | UnifiedMessaging |
HubTransport | Edge | All | Monitoring | CentralAdmin |
CentralAdminDatabase | DomainController | WindowsDeploymentServer |
ProvisionedServer | LanguagePacks | FrontendTransport | CafeArray |
FfoWebService | OSP | ARR | ManagementFrontEnd | ManagementBackEnd | SCOM
| CentralAdminFrontEnd | NAT | DHCP>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example makes the following configuration changes to the Receive connector Internet Receive Connector:

- Sets the *Banner* to `220 SMTP OK`.
- Configures the Receive connector to time out connections after 15 minutes.

```
Set-ReceiveConnector -Identity "Internet Receive Connector"
-Banner "220 SMTP OK" -ConnectionTimeout 00:15:00
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Receive connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ReceiveConnectorIdParameter	The *Identity* parameter specifies the GUID or connector name that represents the Receive connector. The parameter label can be omitted.
AdvertiseClientSettings	Optional	System.Boolean	The *AdvertiseClientSettings* parameter specifies whether the SMTP server name, port number, and authentication settings are displayed in Microsoft Office Outlook Web App, accessed from **Settings > Options > Account > my Account > Settings for POP or IMAP access**. The default value is `$false`.

AuthMechanism	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthMechanisms	The *AuthMechanism* parameter specifies the advertised and accepted authentication mechanisms. The authentication options are `None`, `Tls`, `Integrated`, `BasicAuth`, `BasicAuthRequireTLS`, `ExchangeServer`, and `ExternalAuthoritative`. You can enter multiple values for the *AuthMechanism* parameter by separating the values with commas. If the *RequireTLS* parameter is set to `$true`, the *AuthMechanism* parameter must be set to `Tls` (Transport Layer Security). If you set the *AuthMechanism* parameter to `BasicAuthRequireTLS`, you must also select `BasicAuth` and `Tls`. The *AuthMechanism* parameter value `ExternalAuthoritative` may only coexist with the value `Tls`. If you set the *AuthMechanism*

			parameter to `ExternalAuthoritative`, the *PermissionGroups* parameter must also have the value `ExchangeServers`.

Banner	Optional	System.String	The *Banner* parameter specifies an override to the default SMTP 220 banner. When the value of the *Banner* parameter is blank, the default SMTP banner is the following:
			`220 <Servername>`
			When you specify a value for the *Banner* parameter, you must use the following syntax:
			`"220 <RemainingBa`
			220 is the default service ready SMTP response code as defined in RFC 2821.

BareLinefeedRejectionEnabled	Optional	System.Boolean	The *BareLinefeedRejectionEnabled* parameter specifies whether this Receive connector

			rejects messages that contain bare line feed (LF) characters in the SMTP DATA stream. Line feed characters that aren't immediately preceded by carriage return (CR) characters are known as bare line feeds. Bare line feeds aren't allowed in SMTP communications. Although it may be possible for a message containing a bare line feed to be delivered successfully, such messages don't adhere to the SMTP protocol standards and may cause problems with messaging servers. If you set this parameter to `$true`, the Receive connector rejects any messages that contain bare line feeds. The default value is `$false`.
BinaryMimeEnabled	Optional	System.Boolean	The *BinaryMimeEnabled* parameter specifies

			whether the BINARYMIME EHLO keyword is advertised in the EHLO response to the remote server and is available for use. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. When the *BinaryMimeEnabled* parameter is set to `$true`, the BINARYMIME EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *BinaryMimeEnabled* parameter is set to `$false`, the BINARYMIME EHLO keyword isn't advertised in the EHLO response to the remote server and is disabled. The BINARYMIME extension enables remote computers to send binary message data to this Receive

			connector. The BINARYMIME extension requires the data-chunking service extension, CHUNKING, to be enabled. Therefore, if you set the *BinaryMimeEnabled* parameter to `$true`, you should also set the *ChunkingEnabled* parameter to `$true`.
Bindings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Bindings* parameter specifies the local IP address and TCP port numbers used by the Receive connector to listen for inbound messages. Valid syntax for this parameter is *<IP Address>*:*<TCP Port>*, such as `192.168.1.1:25`. The IP address `0.0.0.0` indicates that the Receive connector uses all IP addresses configured on all network adapters to listen for inbound messages.

			Note: You must specify a local IP address that's valid for the Mailbox server or Edge server on which the Receive connector is located. If you specify an invalid local IP address, the Microsoft Exchange Transport service may fail to start when the service is restarted. To specify all IP addresses configured on all network adapters, you can use the IP address `0.0.0.0`. The values that you specify by using the *Bindings* parameter must satisfy one of the following requirements for uniqueness: • You can specify a unique combination of IP address and TCP port that doesn't conflict with the IP address or TCP port used in the *Bindings* parameter of another Receive connector on the server. • You can use an existing combination of IP address and TCP port, but use the

			RemoteIPRanges parameter to restrict the remote servers serviced by the Receive connector.
ChunkingEnabled	Optional	System.Boolean	The *ChunkingEnabled* parameter specifies whether the CHUNKING EHLO keyword is advertised in the EHLO response to the remote server and is available for use. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. When the *ChunkingEnabled* parameter is set to `$true`, the CHUNKING EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *ChunkingEnabled* parameter is set to `$false`, the CHUNKING EHLO keyword isn't advertised in the EHLO response to the remote server and is disabled. The CHUNKING

			extension enables large message bodies to be relayed by the remote server to the Receive connector in multiple, smaller chunks.
Comment	Optional	System.String	The *Comment* parameter specifies an optional comment. If you specify a value that contains spaces, enclose the value in quotation marks ("), for example: `"This is an admin note"`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ConnectionInactivityTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConnectionInactivityTimeout* parameter specifies the maximum amount of idle time before a connection to

			a Receive connector is closed. The default value for a Receive connector configured in the Transport service of a Mailbox server is 5 minutes. The default value for a Receive connector configured on an Edge server is 1 minute. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify a connection inactivity time-out of 5 minutes, enter `00:05:00`. The value specified by the *ConnectionTimeout* parameter must be greater than the value specified by the *ConnectionInactivityTimeout* parameter. The valid input range for either parameter is from `00:00:01` through `1.00:00:00`.

ConnectionTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConnectionTimeout* parameter specifies the maximum time that a connection can remain open, even if the connection is actively transmitting data. The default value for a Receive connector configured on a Mailbox server is 10 minutes The default value for a Receive connector configured on an Edge server is 5 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify a connection time-out of 5 minutes, enter `00:05:00`. The value specified by the *ConnectionTimeout* parameter must be greater than the value specified by the

			ConnectionInactivityTimeout parameter. The valid input range for either parameter is from `00:00:01` through `1.00:00:00`.
DefaultDomain	Optional	Microsoft.Exchange.Configuration.Tasks.AcceptedDomainIdParameter	The *DefaultDomain* parameter specifies the domain name to append to values submitted to `MAIL FROM` or `RCPT TO` in the message envelope by a sending server if no domain name is provided.
DeliveryStatusNotificationEnabled	Optional	System.Boolean	The *DeliveryStatusNotificationEnabled* parameter specifies whether the delivery status notification (DSN) EHLO keyword is advertised in the EHLO response to the remote server and is available for use. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. When the *DeliveryStatusNotificati*

			onEnabled parameter is set to `$true`, the DSN EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *DeliveryStatusNotificationEnabled* parameter is set to `$false`, the DSN EHLO keyword isn't advertised in the EHLO response to the remote server and is disabled. The DSN extension to extended SMTP (ESMTP) provides enhanced DSN functionality specified in RFC 1891.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An

			Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
DomainSecureEnabled	Optional	System.Boolean	The *DomainSecureEnabled* parameter specifies the first part of the process to enable mutual Transport Layer Security (TLS) authentication for the domains serviced by this Receive connector. Mutual TLS authentication functions correctly only if the following conditions are true: • The value of the *DomainSecureEnabled* parameter is `$true`. • The value of the *AuthMechanism* parameter contains `Tls` and doesn't contain `ExternalAuthoritative`. • The *TLSReceiveDomainSe*

			cureList attribute of your transport configuration contains at least one domain serviced by this Receive connector. The wildcard character (*) isn't supported in domains configured for mutual TLS authentication. The same domain must also be defined on the corresponding Send connector, and in the value of the *TLSSendDomainSecureList* attribute of your Transport configuration. The default value for the *DomainSecureEnabled* parameter is `$false` for the following types of Receive connectors: • All Receive connectors defined on a Mailbox server. • User-created Receive connectors defined on an Edge server.

			The default value for *DomainSecureEnabled* is `$true` for default Receive connectors defined on an Edge server.
EightBitMimeEnabled	Optional	System.Boolean	The *EightBitMimeEnabled* parameter specifies whether the 8BITMIME EHLO keyword is advertised in the EHLO response to the remote server and is available for use. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. When the *EightBitMimeEnabled* parameter is set to `$true`, the 8BITMIME EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *EightBitMimeEnabled* parameter is set to `$false`, the 8BITMIME EHLO keyword isn't advertised in the EHLO

			response to the remote server and is disabled.
EnableAuthGSSAPI	Optional	System.Boolean	The *EnableAuthGSSAPI* parameter specifies how to control the advertisement of the Generic Security Services application programming interface (GSSAPI) authentication method when Integrated Windows authentication is enabled on this connector. If the *AuthMechanism* parameter contains `Integrated`, and the *EnableAuthGSSAPI* parameter is set to `$true`, the AUTH GSSAPI NTLM keyword is advertised in the EHLO response of the Receive connector. Clients may use Kerberos or NTLM to authenticate with the Receive connector. If the *AuthMechanism* parameter contains `Integrated`, and the

			EnableAuthGSSAPI parameter is set to `$false`, the AUTH NTLM keyword is advertised in the EHLO response of the Receive connector. Clients may only use NTLM to authenticate with the Receive connector. If you have Internet Information Services (IIS) messaging servers that authenticate with this Receive connector, you should set the value of the *EnableAuthGSSAPI* parameter to `$false`. Authentication with computers running Microsoft Exchange Server 2003 isn't affected by the value of the *EnableAuthGSSAPI* parameter. Exchange 2003 servers use the authentication methods advertised in the X-EXPS keyword. The X-EXPS keyword is advertised in the EHLO response of the Receive

			connector when the *AuthMechanism* parameter contains `ExchangeServer`. Valid values for this parameter are `$true` or `$false`. The default value is `$false`. By default, the *EnableAuthGSSAPI* parameter is set to `$true` only on the default Receive connector Client <*Server Name*> that's created only in the Transport service on a Mailbox server.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether to enable the connector. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. Use the *Enabled* parameter to enable or disable the connector.
EnhancedStatusCodesEnabled	Optional	System.Boolean	The *EnhancedStatusCodesEnabled* parameter specifies whether the

			ENHANCEDSTATUSCODES EHLO keyword is advertised in the EHLO response to the remote server and is available for use. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. When the *EnhancedStatusCodesEnabled* parameter is set to `$true`, the ENHANCEDSTATUSCODES EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *EnhancedStatusCodesEnabled* parameter is set to `$false`, the ENHANCEDSTATUSCODES EHLO keyword isn't advertised in the EHLO response to the remote server and is disabled. The ENHANCEDSTATUSCODES extension provides enhanced error and status information in DSNs sent to remote

			servers.
ExtendedProtectionPolicy	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ExtendedProtectionPolicySetting	The *ExtendedProtectionPolicy* parameter specifies how you want to use Extended Protection for Authentication on this Receive connector. By default, this parameter is set to `None`. The following are the valid values of this parameter: • `None` Extended Protection for Authentication won't be used. • `Allow` Extended Protection for Authentication will be used only if the connecting host supports it. Otherwise, the connections will be established without Extended Protection for Authentication. • `Require` Extended Protection for Authentication will be required for all incoming connections to this Receive connector. If the connecting host doesn't support Extended Protection for Authentication,

			the connection will be rejected. Extended Protection for Authentication enhances the protection and handling of credentials when authenticating network connections using Integrated Windows authentication. Integrated Windows authentication is also known as NTLM. We strongly recommend that you use Extended Protection for Authentication if you are using Integrated Windows authentication.
Fqdn	Optional	Microsoft.Exchange.Data.Fqdn	The *Fqdn* parameter specifies the FQDN used as the destination server for connected messaging servers that use the Receive connector to send incoming messages. The value of this parameter is displayed to connected

			messaging servers whenever a destination server name is required, as in the following examples: • In the default SMTP banner of the Receive connector • In the EHLO/HELO response of the Receive connector • In the most recent `Received` header field in the incoming message when the message enters the Transport service on a Mailbox server or an Edge server • During TLS authentication The default value of the *Fqdn* parameter is the FQDN of the Mailbox server or Edge server that contains the Receive connector. **Note:** Don't modify the FQDN value on the default Receive connector Default *<Server Name>* that's automatically created on Mailbox servers. If

			you have multiple Mailbox servers in your Exchange organization and you change the FQDN value on the Default <*Server Name*> Receive connector, internal mail flow between Mailbox servers fails.
LiveCredentialEnabled	Optional	System.Boolean	The *LiveCredentialEnabled* parameter is reserved for internal Microsoft use.
LongAddressesEnabled	Optional	System.Boolean	The *LongAddressesEnabled* parameter enables the Receive connector to accept long X.400 email addresses. The X.400 email addresses are encapsulated in SMTP email addresses by using the Internet Mail Connector Encapsulated Address (IMCEA) encapsulation method. When the value of this parameter is `$false`, the maximum length for a complete SMTP email address is 571 characters.

			When the value of this parameter is `$true`, the following changes are made: • The XLONGADDR keyword is advertised in the EHLO response of the Receive connector. • The accepted line length of an SMTP session is increased to 8,000 characters. • Valid long addresses are accepted by the `MAIL FROM` and `RCPT TO` SMTP commands. Therefore, X.400 email addresses can be up to 1,860 characters long after IMCEA encapsulation. Valid values for this parameter are `$true` or `$false`. The default value is `$false`. You can only modify this parameter on Receive connectors configured in the Transport service on a Mailbox server.
MaxAcknowledgement	Optional	Microsoft.Exchange.Dat	This parameter isn't

Delay		a.EnhancedTimeSpan	used by Microsoft Exchange Server 2013. It's only used by Microsoft Exchange 2010 servers in a coexistence environment. The *MaxAcknowledgementDelay* parameter specifies the period the transport server delays acknowledgement when receiving messages from a host that doesn't support shadow redundancy. When receiving messages from a host that doesn't support shadow redundancy, a Microsoft Exchange Server 2010 transport server delays issuing an acknowledgement until it verifies that the message has been successfully delivered to all recipients. However, if it takes too long to verify successful delivery, the transport server times

			out and issues an acknowledgement anyway. The default value is 30 seconds. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
MaxHeaderSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *MaxHeaderSize* parameter specifies in bytes the maximum size of the SMTP message header that the Receive connector accepts before it closes the connection. The default value is 65536 bytes. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is

			from 1 through 2147483647 bytes.
MaxHopCount	Optional	System.Int32	The *MaxHopCount* parameter specifies the maximum number of hops that a message can take before the message is rejected by the Receive connector. The maximum number of hops is determined by the number of `Received` headers in a submitted message. The default value is 60. The valid input range for this parameter is from 1 through 500.
MaxInboundConnection	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxInboundConnection* parameter specifies the maximum number of inbound connections that this Receive connector serves at the same time. The default value is 5000. The valid input range for this parameter is from 1 through 2147483647. To disable the inbound connection limit on a

			Receive connector, enter a value of `unlimited`.
MaxInboundConnectionPercentagePerSource	Optional	System.Int32	The *MaxInboundConnectionPercentagePerSource* parameter specifies the maximum number of connections that a Receive connector serves at the same time from a single IP address. The value is expressed as the percentage of available remaining connections on a Receive connector. Enter the value as an integer without the percent (%) character. The default value is 2 percent. The valid input range for this parameter is from 1 through 100.
MaxInboundConnectionPerSource	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxInboundConnectionPerSource* parameter specifies the maximum number of inbound connections that this Receive connector

			serves at the same time from a single IP address. The default value is 100. The valid input range for this parameter is from 1 through 10000. To disable the inbound connection per source limit on a Receive connector, enter a value of `unlimited`.
MaxLocalHopCount	Optional	System.Int32	The *MaxLocalHopCount* parameter specifies the maximum number of local hops that a message can take before the message is rejected by the Receive connector. The maximum number of local hops is determined by the number of `Received` headers that have local server addresses in a submitted message. The default value is 8. The valid input range for this parameter is from 0 through 50.

			When you specify a value of 0, the message is never rejected based on the number of local hops.
MaxLogonFailures	Optional	System.Int32	The *MaxLogonFailures* parameter specifies the number of logon failures that the Receive connector retries before closing the connection. The default value is 3. The valid input range for this parameter is from 0 through 10.
MaxMessageSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *MaxMessageSize* parameter specifies the maximum size of a message. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes.

			The valid input range for this parameter is from 65536 through 2147483647 bytes.
MaxProtocolErrors	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxProtocolErrors* parameter specifies the maximum number of SMTP protocol errors that the Receive connector accepts before closing the connection. The default value is 5. The valid input range for this parameter is from 0 through 2147483647. When you specify a value of `unlimited`, a connection is never closed because of protocol errors.
MaxRecipientsPerMessage	Optional	System.Int32	The *MaxRecipientsPerMessage* parameter specifies the maximum number of recipients per message that the Receive connector accepts before closing the connection. The default value is 200. The valid input range

			for this parameter is from 1 through 512000.
MessageRateLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *MessageRateLimit* parameter specifies the maximum number of messages that can be sent by a single client IP address per minute. The default value for a Receive connector configured in the Transport service on a Mailbox server is `unlimited`. The default value for a Receive connector configured on an Edge server is 600 messages per minute. The valid input range for this parameter is 1 to 2147483647. To remove the message rate limit on a Receive connector, enter a value of `unlimited`.
MessageRateSource	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MessageRateSourceFlags	The *MessageRateSource* parameter specifies how the message submission rate is

			calculated. It can have one of the following values: • `None` No message submission rate is calculated. • `IPAddress` The message submission rate is calculated for sending hosts. • `User` The message submission rate is calculated for sending users (specified with the `MAIL FROM` SMTP command). • `All` The message submission rate is calculated for both the sending users and sending hosts.
Name	Optional	System.String	The *Name* parameter specifies the administrator-supplied name of the connector. Enter the *Name* parameter as a string, for example: `New Receive Connector`.
OrarEnabled	Optional	System.Boolean	The *OrarEnabled* parameter specifies whether to enable the Originator Requested Alternate Recipient (ORAR). When the value of this parameter

			is `$false`, ORAR isn't supported. When the value of this parameter is `$true`, ORAR is supported by advertising the XORAR keyword in the EHLO response of the Receive connector. The actual ORAR information is transmitted in the RCPT TO SMTP command. Valid values for this parameter are `$true` or `$false`. The default value is `$false`. If the email address specified in the ORAR information is a long X.400 email address, the *LongAddressesEnabled* parameter must be `$true`.
PermissionGroups	Optional	Microsoft.Exchange.Data.PermissionGroups	The *PermissionGroups* parameter specifies the groups or roles that can submit messages to the Receive connector and the permissions assigned to those groups. A

			permission group is a predefined set of permissions granted to well-known security principals. The valid values for this parameter are as follows: `None`, `AnonymousUsers`, `Custom`, `ExchangeUsers`, `ExchangeServers`, `ExchangeLegacyServers`, and `Partners`. The default permission groups assigned to a Receive connector depend on the connector usage type specified by the *Usage* parameter when the Receive connector was created. For more information about Receive connector usage types, see Receive connectors.
PipeliningEnabled	Optional	System.Boolean	The *PipeliningEnabled* parameter specifies whether the PIPELINING EHLO keyword is advertised in the EHLO response to the remote server

			and is available for use. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. When the *PipeliningEnabled* parameter is set to `$true`, the PIPELINING EHLO keyword is advertised in the EHLO response to the remote server and is available for use. When the *PipeliningEnabled* parameter is set to `$false`, the PIPELINING EHLO keyword isn't advertised in the EHLO response to the remote server and is disabled. The PIPELINING extension enables the remote server to send requests without awaiting a response from this Receive connector.
ProtocolLoggingLevel	Optional	Microsoft.Exchange.Data.ProtocolLoggingLevel	The *ProtocolLoggingLevel* parameter specifies whether to enable protocol logging for

			the specified Receive connector. A value of `verbose` enables protocol logging for the connector. A value of `None` disables protocol logging for the connector. The default value is `None`. The location of the Receive connector protocol logs for all Receive connectors configured in the Transport service on a Mailbox server or an Edge server is specified by using the **Set-TransportService** cmdlet with the *ReceiveProtocolLogPath* parameter.
ProxyEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RemoteIPRanges	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RemoteIPRanges* parameter specifies the remote IP addresses from which this connector accepts messages. Valid syntax for this parameter is

			<*Starting IP Address*>-<*Ending IP Address*>, such as `192.168.1.1-192.168.1.10`. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...`. To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. Multiple Receive connectors on the same server can have overlapping remote IP address ranges as long as one IP address range is completely overlapped by another IP address range. When

			remote IP address ranges overlap, the remote IP address range with the most specific match to the IP address of the connecting server is used.
RequireEHLODomain	Optional	System.Boolean	The *RequireEHLODomain* parameter specifies whether the remote computer must provide a domain name in the EHLO handshake after the SMTP connection is established. Valid values for this parameter are `$true` or `$false`. The default value is `$false`. When the *RequireEHLODomain* parameter is set to `$true`, the remote computer must provide a domain name in the EHLO handshake after the SMTP connection is established. If the remote computer doesn't provide the

			domain name, the SMTP connection is closed.
RequireTLS	Optional	System.Boolean	The *RequireTLS* parameter specifies whether all messages received by this connector require TLS transmission. Valid values for this parameter are `$true` or `$false`. The default value is `$false`. When the *RequireTLS* parameter is set to `$true`, all messages received by this connector require TLS transmission.
ServiceDiscoveryFqdn	Optional	Microsoft.Exchange.Data.Fqdn	The service discovery fully-qualified domain name (FQDN).
SizeEnabled	Optional	Microsoft.Exchange.Data.SizeMode	The *SizeEnabled* parameter specifies whether the SIZE SMTP extension is enabled. Valid values for this parameter are `Enabled`, `Disabled`, or `EnabledwithoutValue`. The default value is

			`Enabled`. When the *SizeEnabled* parameter is set to `Enabled`, the SIZE SMTP extension is enabled, and the maximum allowable message size value from the *MaxMessageSize* parameter is advertised in the EHLO banner. When the *SizeEnabled* parameter is set to `Disabled`, the SIZE SMTP extension isn't used, and the maximum allowable message size value is never disclosed to the remote server. When the *SizeEnabled* parameter is set to `EnabledWithoutValue`, the SIZE SMTP extension is enabled, but the maximum allowable message size value from the *MaxMessageSize* parameter isn't advertised in the EHLO banner. This allows the message to bypass

			message size checks for authenticated connections between Mailbox servers. The SIZE SMTP extension is defined in RFC 1870. SIZE enables the source server to declare the size of the inbound message to the target server. It also allows the target server to declare the maximum message size that it's allowed to accept to the sending server. If the advertised size of the inbound message exceeds the value in the *MaxMessageSize* parameter, the Receive connector responds to the remote server by using an error code and closes the connection.
SmtpUtf8Enabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
SuppressXAnonymousTls	Optional	System.Boolean	The *SuppressXAnonymousTls* parameter specifies

			whether this Receive connector supports the standard TLS encryption for incoming connections. By default, all communications between Exchange 2010 Mailbox servers is protected with TLS. However, if you need to disable TLS on a specific connection in your organization, you can create a specific Receive connector and set the *SuppressXAnonymousTls* parameter to `$true`. The default value is `$false`. Before you can set this parameter to `$true`, you must use the **Set-TransportService** command to set the *UseDownGradedExchangeServerAuth* parameter to `$true` on the server that this Receive connector is configured on.

TarpitInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *TarpitInterval* parameter specifies the period of time to delay an SMTP response to a remote server that may be abusing the connection. Authenticated connections are never delayed in this manner. The default value is 5 seconds. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The valid input range for this parameter is from `00:00:00` through `00:10:00`. When you set the value to `00:00:00`, you disable the tarpit interval.
TlsCertificateName	Optional	Microsoft.Exchange.Data.SmtpX509Identifier	The *TlsCertificateName* parameter specifies the X.509 certificate to use with TLS sessions and secure mail. Valid input for this parameter is `[I]`*Issuer*`[S]`*Subject*.

			The *Issuer* value is found in the certificate's `Issuer` field, and the *Subject* value is found in the certificate's `Subject` field. You can find these values by running the **Get-ExchangeCertificate** cmdlet.
TlsDomainCapabilities	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *TlsDomainCapabilities* parameter specifies the different capabilities this Receive connector will make available to specific hosts outside your organization. TLS with certificate validation is used to authenticate remote hosts before these capabilities are offered. To specify capabilities for a domain, use the following syntax: <domain name 1>:<capability 1, capability 2,...,capability N> If you're specifying

			capabilities for multiple domains, list the configuration for each domain in quotation marks, separated by commas. For example: "contoso.com:AcceptOorgProtocol","fabrikam.com:AcceptOorgProtocol,AcceptOorgHeader" You can configure the capabilities for non-TLS encrypted incoming connections using the special "NO-TLS" domain. You can configure the following capabilities for a domain: • `AcceptOorgProtocol` • `AcceptOorgHeader`
TransportRole	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ServerRole	The *TransportRole* parameter designates the server role associated with this connector. Types include `FrontendTransport` and `HubTransport`. Typically used to specify the server role when you host multiple server roles on a single

			computer.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-RemoteDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-RemoteDomain** cmdlet to view the configuration information for the remote domains

configured in your organization. You can view the remote domain configuration from inside the Exchange organization or from an Edge Transport server in the perimeter network.

```
Get-RemoteDomain [-Identity <RemoteDomainIdParameter>] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example returns all remote domains configured in the Active Directory forest in which you run the command.

```
Get-RemoteDomain
```

EXAMPLE 2

This example returns the configuration for the remote domain Contoso.

```
Get-RemoteDomain Contoso
```

EXAMPLE 3

This example queries Active Directory for all remote domains and displays only those remote domains for which Transport Neutral Encapsulation Format (TNEF) encoding isn't used.

```
Get-RemoteDomain | Where {$_.TNEFEnabled -eq $false}
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remote domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.RemoteDomainIdParameter	The *Identity* parameter specifies the remote domain you want to view. Enter either the GUID or name of the remote domain.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-RemoteDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-RemoteDomain** cmdlet to create a managed connection for a remote domain. When you create a remote domain, you can control mail flow with more precision, apply message formatting and messaging policies, and specify acceptable character sets for messages sent to and received from the remote domain.

```
New-RemoteDomain -DomainName <SmtpDomainWithSubdomains> -Name <String> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the remote domain Contoso.

```
New-RemoteDomain -DomainName Contoso.com -Name Contoso
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remote domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description

DomainName	Required	Microsoft.Exchange.Data.SmtpDomainWithSubdomains	The *DomainName* parameter specifies the SMTP domain that you want to establish as a remote domain. Valid input for the *DomainName* parameter is an SMTP domain. You can use a wildcard character to specify all subdomains of a specified domain, as shown in the following example: `*.contoso.com`. However, you can't embed a wildcard character, as shown in the following example: `domain.*.contoso.com`. The domain name string may contain a maximum of 256 characters.
Name	Required	System.String	The *Name* parameter specifies a unique name for the remote domain object.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.Orga	The *Organization* parameter is reserved

		nizationIdParameter	for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-RemoteDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-RemoteDomain** cmdlet to remove a remote domain. When you remove a remote domain, the remote domain object is deleted. Removing a remote domain doesn't disable mail flow to that domain.

```
Remove-RemoteDomain -Identity <RemoteDomainIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the remote domain object named Contoso.

```
Remove-RemoteDomain Contoso
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remote domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RemoteDomainIdParameter	The *Identity* parameter specifies the remote domain you want to remove. Enter either the GUID or name of the remote domain.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt,

			use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the

			WhatIf switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-RemoteDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-RemoteDomain** cmdlet to configure a managed connection for a remote domain.

```
Set-RemoteDomain -Identity <RemoteDomainIdParameter> [-AllowedOOFType
<External | InternalLegacy | ExternalLegacy | None>] [-AutoForwardEnabled
<$true | $false>] [-AutoReplyEnabled <$true | $false>] [-
ByteEncoderTypeFor7BitCharsets <Use7Bit | UseQP | UseBase64 |
UseQPHtmlDetectTextPlain | UseBase64HtmlDetectTextPlain |
UseQPHtml7BitTextPlain | UseBase64Html7BitTextPlain | Undefined>] [-
CharacterSet <String>] [-Confirm [<SwitchParameter>]] [-ContentType
<MimeHtmlText | MimeText | MimeHtml>] [-DeliveryReportEnabled <$true |
$false>] [-DisplaySenderName <$true | $false>] [-DomainController <Fqdn>]
[-IsInternal <$true | $false>] [-LineWrapSize <Unlimited>] [-
MeetingForwardNotificationEnabled <$true | $false>] [-
MessageCountThreshold <Int32>] [-Name <String>] [-NDRDiagnosticInfoEnabled
<$true | $false>] [-NDREnabled <$true | $false>] [-NonMimeCharacterSet
<String>] [-PreferredInternetCodePageForShiftJis <Undefined | Iso2022Jp |
```

```
Esc2022Jp | Sio2022Jp>] [-RequiredCharsetCoverage <Int32>] [-
TargetDeliveryDomain <$true | $false>] [-TNEFEnabled <$true | $false>] [-
TrustedMailInboundEnabled <$true | $false>] [-TrustedMailOutboundEnabled
<$true | $false>] [-UseSimpleDisplayName <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example performs the following actions:

- It disables out-of-office notifications to the remote domain.
- It suppresses read receipts sent from clients in your organization to the remote domain.
- It enables TNEF message data on messages sent to the remote domain.

```
Set-RemoteDomain Contoso -AllowedOOFType None -
DeliveryReportEnabled $false -TNEFEnabled $true
```

EXAMPLE 2

This example queries Active Directory for all remote domains for which auto replies are disabled. Using the pipelining feature, it also disables auto forwards and NDRs to those domains.

```
Get-RemoteDomain | Where {$_.AutoReplyEnabled -eq $false} |
Set-RemoteDomain -AutoForwardEnabled $false -NDREnabled
$false
```

Detailed Description

When you set a remote domain, you can control mail flow with more precision, specify message formatting and policy, and specify acceptable character sets for messages sent to or received from the remote domain.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remote domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RemoteDomainIdParamete	The *Identity* parameter specifies the display name of the remote domain.

		r	The length of the name can't exceed 64 characters.
AllowedOOFType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AllowedOOFType	The *AllowedOOFType* parameter specifies the type of out-of-office notification returned to users at the remote domain. Valid values are `External`, `ExternalLegacy`, `None`, and `InternalLegacy`. The default value is `External`.
AutoForwardEnabled	Optional	System.Boolean	The *AutoForwardEnabled* parameter specifies whether to allow messages that are auto-forwarded by client e-mail programs in your organization. Setting this parameter to `$true` enables auto-forwarded messages to be delivered to the remote domain. The default value is `$false`.
AutoReplyEnabled	Optional	System.Boolean	The *AutoReplyEnabled* parameter specifies whether to allow messages that are automatic replies from client e-mail programs in

			your organization. Setting this parameter to `$true` enables automatic replies to be delivered to the remote domain. The default value is `$false`.
ByteEncoderTypeFor7BitCharsets	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ByteEncoderTypeFor7BitCharsetsEnum	The *ByteEncoderTypeFor7BitCharsets* parameter specifies the 7-bit transfer encoding method for MIME format for messages sent to this remote domain. The valid values for this parameter are: • `Use7Bit` Always use default 7-bit transfer encoding for HTML and plain text. • `UseQP` Always use QP (quoted-printable) encoding for HTML and for plain text. • `UseBase64` Always use Base64 encoding for HTML and for plain text. • `UseQPHtmlDetectTextPlain` Use QP encoding for HTML and for plain text unless line wrapping is enabled in plain text. If line wrapping is enabled, use 7-bit encoding for plain text. • `UseBase64HtmlDetectTextPlain` Use Base64

			encoding for HTML and for plain text, unless line wrapping is enabled in plain text. If line wrapping is enabled in plain text, use Base64 encoding for HTML, and use 7-bit encoding for plain text. • `UseQPHtml7BitTextPlain` Always use QP encoding for HTML. Always use 7-bit encoding for plain text. • `UseBase64Html7BitTextPlain` Always use Base64 encoding for HTML. Always use 7-bit encoding for plain text. • `Undefined` Always use QP encoding for HTML and plain text. The default value is `Undefined`.
CharacterSet	Optional	System.String	The *CharacterSet* parameter specifies a character set for this remote domain. The character set that you specify is only used for MIME messages that don't have their own character set specified. Setting this parameter doesn't overwrite character sets already specified in the outbound mail. To remove the character set value, set

			the value to `$null`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ContentType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ContentType	The *ContentType* parameter specifies the outbound message content type and formatting. Valid values for this parameter are `MimeHtmlText`, `MimeText`, or `MimeHtml`: • `MimeHtmlText` converts messages to MIME messages that use HTML formatting, unless the original message is a text message. If the original message is a text message, the outbound message is a MIME message that uses text formatting. • `MimeText` converts all messages to MIME messages that use text formatting. • `MimeHtml` converts all messages to MIME

			messages that use HTML formatting. The default value is `MimeHtmlText`.
DeliveryReportEnabled	Optional	System.Boolean	The *DeliveryReportEnabled* parameter specifies whether to allow delivery reports from client software in your organization to the remote domain. The default value is `$true`.
DisplaySenderName	Optional	System.Boolean	The *DisplaySenderName* parameter specifies whether to display the sender name. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. This parameter is used for older versions of Exchange and should only be set under the direction of Microsoft Customer Service and Support.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
IsInternal	Optional	System.Boolean	The *IsInternal* parameter specifies whether the recipients in this remote domain should be considered internal recipients. Set this parameter to `$true` if this remote domain is part of your cross-premises deployment. When you set this parameter to `$true`, all transport components, like transport rules or any agents you may have deployed, treat this remote domain as an

			internal domain. The default value is `$false`.
LineWrapSize	Optional	Microsoft.Exchange.Data.Unlimited	The *LineWrapSize* parameter specifies the line-wrap size for outbound messages. The parameter takes an integer from 0 through 132, or you can overload the parameter by setting the value to `unlimited`. The default value is `unlimited`.
MeetingForwardNotificationEnabled	Optional	System.Boolean	The *MeetingForwardNotificationEnabled* parameter specifies whether to enable meeting forward notifications. When this parameter is enabled, meeting requests forwarded to recipients in the remote domain generate a meeting forward notification to the meeting organizer. When this parameter is disabled, meeting requests forwarded to recipients in the remote domain won't generate a meeting

			forward notification to the meeting organizer. Valid values for this parameter are `$true` or `$false`. The default value is `$true`.
MessageCountThreshold	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *MessageCountThreshold* parameter specifies the acceptable message count for the remote domain. If the message count exceeds this value, an event is generated that's visible using the **Get-ServerHealth** and **Get-HealthReport** cmdlets. Valid input for this parameter is an integer. The default value is **Int32** (2147483647). The default value indicates there is no message count threshold defined, and that the message count to the remote domain is unmonitored.
Name	Optional	System.String	The *Name* parameter

			specifies a unique name for the remote domain object.
NDRDiagnosticInfoEnabled	Optional	System.Boolean	The *NDRDiagnosticInfoEnabled* parameter specifies whether the diagnostic information is included in NDRs sent to the remote domain. The diagnostic information of an NDR includes details that help administrators troubleshoot delivery problems. This detailed information includes internal server names. You may not want to expose this information to NDRs sent to external users. If you set this parameter to `$false`, the diagnostic information section in the NDR body as well as internal server headers from the attached original message headers are removed from the NDR. The default value is `$true`.

NDREnabled	Optional	System.Boolean	The *NDREnabled* parameter specifies whether to allow non-delivery reports (NDRs) from your organization. Setting this parameter to `$false` suppresses NDRs to the remote domain. The default value is `$true`.
NonMimeCharacterSet	Optional	System.String	The *NonMimeCharacterSet* parameter specifies a character set for this remote domain. The character set that you specify is only used for non-MIME (RFC 822 text) messages that don't have their own character set specified. Setting this parameter doesn't overwrite character sets already specified in the outbound mail. To remove the character set value, set the value to `$null`.
PreferredInternetCodePageForShiftJis	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.PreferredInternetCodePageForShiftJisEnum	The *PreferredInternetCodePageForShiftJis* parameter specifies the specific code page to use for Shift JIS character encoding when

			sending messages to this remote domain. The valid values for this parameter are: • 50220: Use ISO-2022-JP codepage. • 50221: Use ESC-2022-JP codepage. • 50222: Use SIO-2022-JP codepage.
RequiredCharsetCoverage	Optional	System.Int32	The *RequiredCharsetCoverage* parameter specifies a percentage threshold for characters in a message that must match to apply your organization's preferred character set before switching to automatic character set detection. For example, if you set this parameter to 60, the preferred character sets will still be used during content conversion for messages that contain characters from non-preferred character sets as long as the percentage of those characters is 40 percent or less. If the percentage of characters in a message doesn't

			belong to preferred character sets, Exchange analyzes the UNICODE characters and automatically determines the best matching character set to use. If users in this remote domain use characters that span character sets, you may want to specify a lower percentage to ensure that your organization's preferred character set is used during content conversion.
TargetDeliveryDomain	Optional	System.Boolean	The *TargetDeliveryDomain* parameter specifies the e-mail domain that's used when generating target addresses for new mail users in a cross-premises deployment scenario. When you have a cross-premises deployment, the user mailboxes on the remote location are represented as mail user objects. For example, all mailboxes hosted on

			Exchange Online are represented as mail users in your on-premises organization. The value of this parameter is used to determine what domain should be used for the target e-mail address for these mail users.
TNEFEnabled	Optional	System.Boolean	The *TNEFEnabled* parameter specifies whether Transport Neutral Encapsulation Format (TNEF) message encoding is used on messages sent to the remote domain. Valid values for this parameter are `$true`, `$false`, or `$null`. The action associated with each value is as follows: `$true` TNEF encoding is used on all messages sent to the remote domain. `$false` TNEF encoding isn't used on any messages sent to the remote domain. `$null` TNEF encoding isn't specified for the remote domain. TNEF encoding for recipients in the remote domain may be specified by the

			following: • Value of the *UseMapiRichTextFormat* parameter for any mail user or mail contact objects • Sender's per-recipient settings in Microsoft Outlook • Sender's default Internet message settings in Outlook The default value is `$null`.
TrustedMailInboundEnabled	Optional	System.Boolean	The *TrustedMailInboundEnabled* parameter specifies whether Exchange will treat e-mail received from this remote domain as trusted messages. If you set this parameter to `$true`, all incoming messages from this remote domain are considered safe and they will bypass content and recipient filtering. We recommend that you set this parameter to `$true` for cross-premises deployment scenarios. The default value is

			`$false`.
TrustedMailOutboundEnabled	Optional	System.Boolean	The *TrustedMailOutboundEnabled* parameter specifies whether the remote domain is considered a trusted domain. We recommend that you set this parameter to `$true` for cross-premises deployment scenarios. The default value is `$false`.
UseSimpleDisplayName	Optional	System.Boolean	The *UseSimpleDisplayName* parameter specifies whether simple display names for senders appear in messages sent to this domain. Setting this parameter to `$true` enables simple display names for this remote domain. The default value is `$false`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what

			changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-ResubmitRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Add-ResubmitRequest** cmdlet to add requests to replay redundant copies of messages from Safety Net after a mailbox database recovery.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-ResubmitRequest -Destination <Guid> -EndTime <DateTime> -StartTime
<DateTime> [-Confirm [<SwitchParameter>]] [-CorrelationId <Guid>] [-Server
<ServerIdParameter>] [-TestOnly <$true | $false>] [-
UnresponsivePrimaryServers <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example replays the redundant copies of messages delivered from 6:00 PM June 1, 2012 to

5:00 AM June 2 2012 to the recovered mailbox database 5364aeea-6e6b-4055-8258-229b2c6ac9a2.

```
Add-ResubmitRequest -Destination 5364aeea-6e6b-4055-8258-
229b2c6ac9a2 -StartTime "06/01/2012 6:00 PM" -EndTime
"06/02/2012 5:00 AM"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Destination	Required	System.Guid	The *Destination* parameter specifies the GUID of the destination mailbox database. To find the GUID of the mailbox database, run the command: `Get-MailboxDatabase -Server <servername> | Format-List Name,GUID`.
EndTime	Required	System.DateTime	The *EndTime* parameter specifies the delivery time of the latest messages that need to be resubmitted from Safety Net. Use the short date format defined in the **Regional Options** settings for the computer on which the

			command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**. The date and time specified by the *EndTime* parameter must be later than the date and time specified by the *StartTime* parameter. The date and time specified by both parameters must be in the past.
StartTime	Required	System.DateTime	The *StartTime* parameter specifies the delivery time of the oldest messages that need to be resubmitted from Safety Net. Use the short date format defined in the **Regional**

			Options settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**. The date and time specified by the *StartTime* parameter must be earlier than the date and time specified by the *EndTime* parameter. The date and time specified by both parameters must be in the past.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before

			processing continues. You don't have to specify a value with the *Confirm* switch.
CorrelationId	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.
TestOnly	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
UnresponsivePrimaryServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UnresponsivePrimaryServers* parameter identifies the primary servers that should resubmit the messages from Safety Net as being unavailable so other servers can

			resubmit the messages. If the primary servers are unavailable, you can designate other servers that hold redundant copies of the messages in Safety Net to resubmit their copies of the messages. However, you must identify the unresponsive primary servers to the other servers using this parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ResubmitRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ResubmitRequest** cmdlet to view requests to replay redundant copies of messages from Safety Net after a mailbox database recovery.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ResubmitRequest [-Identity <ResubmitRequestIdentityParameter>] [-
Server <ServerIdParameter>]
```

Examples

Example 1

This example returns the details of all resubmit requests.

```
Get-ResubmitRequest
```

Example 2

This example returns details about the resubmit request with the identity 1.

```
Get-ResubmitRequest 1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ResubmitRequestIdentityParameter	The *Identity* parameter specifies the resubmit request you want to view. Each resubmit request is identified by an incremented integer value.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ResubmitRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-ResubmitRequest** cmdlet to remove requests to replay redundant copies of messages from Safety Net after a mailbox database recovery.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ResubmitRequest -Identity <ResubmitRequestIdentityParameter> [-
Confirm [<SwitchParameter>]] [-Server <ServerIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example removes the resubmit request with the identity 5.

```
Remove-ResubmitRequest 5
```

Example 2

This example removes all resubmit requests.

```
Get-ResubmitRequest | Remove-ResubmitRequest
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.ResubmitRequestIdentityParameter	specifies the resubmit request you want to remove. Each resubmit request is identified by an incremented integer value.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ResubmitRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ResubmitRequest** cmdlet to enable or disable requests to replay redundant copies of messages from Safety Net after a mailbox database recovery.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ResubmitRequest -Identity <ResubmitRequestIdentityParameter> -Enabled
<$true | $false> [-Confirm [<SwitchParameter>]] [-Server
```

```
<ServerIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example disables the resubmit request with the identity 8.

```
Set-ResubmitRequest 8 -Enabled $false
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Queues" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Enabled	Required	System.Boolean	The *Enabled* parameter enables or disables an active resubmit request. Valid input for this parameter is `$true` or `$false`. Setting the value to `$false` disables the resubmit request.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ResubmitRequestIdentityParameter	The *Identity* parameter specifies the resubmit request you want to modify. Each resubmit request is identified by an incremented integer value.
Confirm	Optional	System.Management.	The *Confirm* switch

		Automation.SwitchParameter	causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply

			any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SendConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-SendConnector** cmdlet to view the settings for a Send connector.

```
Get-SendConnector [-Identity <SendConnectorIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example displays detailed information about the Send connector named Contoso.com Send Connector.

```
Get-SendConnector "Contoso.com Send Connector" | Format-
List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Send connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.SendConnectorIdParameter	The *Identity* parameter specifies the name, or GUID of the Send connector. If the *Identity* name contains spaces, enclose the

			name in quotation marks ("). You can omit the *Identity* parameter label. You can also include the server name by using the format *ServerName**ConnectorName*.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-SendConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-17*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-SendConnector** cmdlet to create a new Send connector.

```
New-SendConnector -AddressSpaces <MultiValuedProperty> -Name <String> [-
IsScopedConnector <$true | $false>] [-AuthenticationCredential
<PSCredential>] [-CloudServicesMailEnabled <$true | $false>] [-Comment
<String>] [-Confirm [<SwitchParameter>]] [-ConnectionInactivityTimeOut
<EnhancedTimeSpan>] [-Custom <SwitchParameter>] [-DNSRoutingEnabled <$true
| $false>] [-DomainController <Fqdn>] [-DomainSecureEnabled <$true |
$false>] [-Enabled <$true | $false>] [-ErrorPolicies <Default |
DowngradeDnsFailures | DowngradeCustomFailures | UpgradeCustomFailures>]
[-Force <SwitchParameter>] [-ForceHELO <$true | $false>] [-Fqdn <Fqdn>] [-
FrontendProxyEnabled <$true | $false>] [-IgnoreSTARTTLS <$true | $false>]
[-Internal <SwitchParameter>] [-Internet <SwitchParameter>] [-
MaxMessageSize <Unlimited>] [-Partner <SwitchParameter>] [-Port <Int32>]
[-ProtocolLoggingLevel <None | Verbose>] [-RequireOorg <$true | $false>]
[-RequireTLS <$true | $false>] [-SmartHostAuthMechanism <None | BasicAuth
| BasicAuthRequireTLS | ExchangeServer | ExternalAuthoritative>] [-
SmartHosts <MultiValuedProperty>] [-SmtpMaxMessagesPerConnection <Int32>]
```

```
[-SourceIPAddress <IPAddress>] [-SourceTransportServers
<MultiValuedProperty>] [-TlsAuthLevel <EncryptionOnly |
CertificateValidation | DomainValidation>] [-TlsCertificateName
<SmtpX509Identifier>] [-TlsDomain <SmtpDomainWithSubdomains>] [-Usage
<Custom | Internal | Internet | Partner>] [-UseExternalDNSServersEnabled
<$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the Send connector named MySendConnector with the following properties:

- It sends email messages over the Internet.
- It processes messages addressed only to Contoso.com and Fabrikam.com domains.

```
New-SendConnector -Internet -Name MySendConnector -
AddressSpaces contoso.com,fabrikam.com
```

EXAMPLE 2

This example creates the Send connector Secure Email to Contoso.com with the following properties:

- It processes messages only for the Contoso.com domain.
- It uses Basic authentication.
- It uses a specific authentication credential.

To assign a specific authentication credential for the Send connector, you must first run the **Get-Credential** command and store the user input in a temporary variable. When you run the **Get-Credential** command, the command asks for the user name and password of the account used during authentication with the Contoso.com email server. The temporary variable can then be used in the **New-SendConnector** cmdlet to create the new connector.

```
$CredentialObject = Get-Credential
New-SendConnector -Name "Secure Email to Contoso.com" -
AddressSpaces contoso.com -AuthenticationCredential
$CredentialObject -SmartHostAuthMechanism BasicAuth
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Send connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
AddressSpaces	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *AddressSpaces* parameter specifies the domain names to which the Send connector routes mail. The complete syntax for entering each address space is as follows: *<AddressSpaceType>:<AddressSpace>;<AddressSpaceCost>* • **AddressSpaceType:** On an Edge server, the address space type must be SMTP. In the Transport service on a Mailbox server, the address space type may be SMTP, X400, or any other text string. If you omit the address space type, SMTP is assumed. • **AddressSpace:** For SMTP address space types, the address space that you enter must be RFC 1035-compliant. For example, *, *.com, and *.contoso.com are permitted, but *contoso.com is not. For X.400 address space types, the address space that you enter must be

			RFC 1685-compliant, such as **o=MySite;p=MyOrg;a=adatum;c=us**. For all other values of address space type, you can enter any text for the address space. • **AddressSpaceCost:** The valid input range for the cost is from 1 through 100. A lower cost indicates a better route. This parameter is optional. If you omit the address space cost, a cost of 1 is assumed. If you enter a non-SMTP address space that contains the semicolon character (;), you must specify the address space cost. If you specify the address space type or the address space cost, you must enclose the address space in quotation marks ("). For example, the following address space entries are equivalent: • `"SMTP:contoso.com;1"` • `"contoso.com;1"` • `"SMTP:contoso.com"` • `contoso.com`

			You may specify multiple address spaces by separating the address spaces with commas, for example: `contoso.com,fabrikam.com`. If you specify the address space type or the address space cost, enclose the address space in quotation marks ("), for example: `"contoso.com;2","fabrikam.com;3"`. If you specify a non-SMTP address space type on a Send connector configured in the Transport service on a Mailbox server, you must configure the following parameters: • The *SmartHosts* parameter must be set to a value that specifies a smart host. • The *DNSRoutingEnabled* parameter must be set to `$false`. **Note:** Although you can configure non-SMTP address spaces on a Send connector in the Transport service on a

			Mailbox server, the Send connector uses SMTP as the transport mechanism to send messages to other messaging servers. Foreign connectors in the Transport service on a Mailbox server are used to send messages to local messaging servers, such as third-party fax gateway servers, which don't use SMTP as their primary transport mechanism. For more information, see Foreign connectors.
Name	Required	System.String	The *Name* parameter specifies a descriptive name for the connector.
AuthenticationCredential	Optional	System.Management.Automation.PSCredential	The *AuthenticationCredential* parameter specifies a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information about the **Get-Credential** cmdlet, enter **Get-Help Get-Credential** in the Exchange Management Shell.
CloudServicesMailEnabled	Optional	System.Boolean	Set to `$true` to enable this connector to send messages to the cloud service.

Comment	Optional	System.String	The *Comment* parameter specifies an optional comment. If you specify a value that contains spaces, enclose the value in quotation marks ("), for example: "This is an admin note".
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ConnectionInactivityTimeOut	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConnectionInactivityTimeOut* parameter specifies the maximum time an idle connection can remain open. The default value is ten minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify

			fifteen minutes, set it to `00:15:00`. The valid input range for this parameter is `00:00:01` to `1.00:00:00`.
Custom	Optional	System.Management.Automation.SwitchParameter	The *Custom* parameter specifies the `custom` usage type. The usage type specifies the permissions and authentication methods assigned to the Send connector. If you use the *Custom* parameter, you can't use any of the following parameters: • *Internal* • *Internet* • *Partner* • *Usage* For more information about Send connector usage types, permissions, and authentication methods, see Send connectors.
DNSRoutingEnabled	Optional	System.Boolean	The *DNSRoutingEnabled* parameter specifies whether the Send connector uses Domain Name System (DNS) to route mail. Valid values for this parameter are

			`$true` or `$false`. The default value is `$true`. If you specify a *SmartHosts* parameter, the *DNSRoutingEnabled* parameter must be `$false`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
DomainSecureEnabled	Optional	System.Boolean	The *DomainSecureEnabled* parameter enables mutual Transport Layer Security (TLS) authentication for the domains serviced by the Send connector when

			set to `$true`. Mutual TLS authentication functions correctly only if the following conditions are met: • *DomainSecureEnabled* is set to `$true`. • *DNSRoutingEnabled* is set to `$true`. • *IgnoreSTARTTLS* is set to `$false`. The wildcard character (*) isn't supported in domains configured for mutual TLS authentication. The same domain must also be defined on the corresponding Receive connector, and in the value of the *TLSReceiveDomainSecureList* attribute of the transport configuration. The default value for the *DomainSecureEnabled* parameter is `$false` for the following types of Send connectors: • Those defined in the Transport service on a Mailbox server. • User-created Send

			connectors defined on an Edge server. The default value is `$true` for a default Send connector defined on an Edge server.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether to enable the Send connector to process email messages. Valid values are `$true` or `$false`. The default value is `$true`.
ErrorPolicies	Optional	Microsoft.Exchange.Data.ErrorPolicies	The *ErrorPolicies* parameter specifies how communication errors are treated. Possible values are the following: • `Default` A non-delivery report (NDR) is generated for communication errors. • `DowngradeDnsFailures` All DNS errors are treated as transient. • `DowngradeCustomFailures` Particular SMTP errors are treated as transient. • `UpgradeCustomFailures` Custom transient failures are upgraded and treated as permanent failures. Multiple values can be

			specified for this parameter, separated by commas. Specify a value other than `Default` for this parameter only if this Send connector is used to send messages over a reliable and well-defined communication channel where communication errors aren't expected. For example, consider specifying a value other than `Default` if this Send connector is used to send messages to a partner.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.

ForceHELO	Optional	System.Boolean	The *ForceHELO* parameter specifies whether HELO is sent instead of the default EHLO. Valid values are `$true` or `$false`. The default value is `$false`.
Fqdn	Optional	Microsoft.Exchange.Data.Fqdn	The *Fqdn* parameter specifies the FQDN used as the source server for connected messaging servers that use the Send connector to receive outgoing messages. The value of this parameter is displayed to connected messaging servers whenever a source server name is required, as in the following examples: • In the EHLO/HELO command when the Send connector communicates with the next hop messaging server • In the most recent `Received` header field added to the message by the next hop messaging server after the message leaves the Transport service on a Mailbox server or an

			Edge server • During TLS authentication The default value of the *Fqdn* parameter is `$null`. This means the default FQDN value is the FQDN of the Mailbox server or Edge server that contains the Send connector.
FrontendProxyEnabled	Optional	System.Boolean	The *FrontendProxyEnabled* parameter routes outbound messages through the CAS server, where destination specific routing, such as DNS or IP address, is set, when the parameter is set to `$true`.
IgnoreSTARTTLS	Optional	System.Boolean	The *IgnoreSTARTTLS* parameter specifies whether to ignore the StartTLS option offered by a remote sending server. This parameter is used with remote domains. This parameter must be set to `$false` if the *RequireTLS* parameter is set to `$true`. Valid values for this parameter are `$true` or `$false`.

Internal	Optional	System.Management.Automation.SwitchParameter	The *Internal* parameter specifies the `Internal` usage type. The usage type specifies the permissions and authentication methods assigned to the Send connector. If you use the *Internal* parameter, you can't use any of the following parameters: • *Custom* • *Internet* • *Partner* • *Usage* For more information about Send connector usage types, permissions, and authentication methods, see Send connectors.
Internet	Optional	System.Management.Automation.SwitchParameter	The *Internet* parameter specifies the `Internet` usage type. The usage type specifies the permissions and authentication methods assigned to the Send connector. If you use the *Internet* parameter, you can't use any of the following parameters:

			• *Custom* • *Internal* • *Partner* • *Usage* For more information about Send connector usage types, permissions, and authentication methods, see Send connectors.
IsScopedConnector	Optional	System.Boolean	The *IsScopedConnector* parameter specifies the availability of the connector to other Mailbox servers with the Transport service. When the value of this parameter is `$false`, the connector can be used by all Mailbox servers in the Exchange organization. When the value of this parameter is `$true`, the connector can only be used by Transport service on Mailbox servers in the same Active Directory site. The default value is `$false`.
MaxMessageSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxMessageSize* parameter specifies the maximum size of a

			message that can pass through a connector. The default value is 25 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. Values entered in bytes are rounded up to nearest kilobyte. The valid input range for this parameter is 0 to 2147483647 bytes. To remove the message size limit on a Send connector, enter a value of `unlimited`.
Partner	Optional	System.Management.Automation.SwitchParameter	The *Partner* parameter specifies the `Partner` usage type. The usage type specifies the permissions and authentication methods assigned to the Send connector. If you use the *Partner* parameter, you can't use any of the following parameters:

			• *Custom* • *Internal* • *Internet* • *Usage* For more information about Send connector usage types, permissions, and authentication methods, see Send connectors.
Port	Optional	System.Int32	The *Port* parameter specifies the port number for smart host forwarding, if you specify a value in the *SmartHosts* parameter. The valid input range is an integer from 0 through 65535. The default value is 25. In most organizations, the port number is set to 25.
ProtocolLoggingLevel	Optional	Microsoft.Exchange.Data.ProtocolLoggingLevel	The *ProtocolLoggingLevel* parameter specifies whether to enable protocol logging. `Verbose` enables protocol logging. `None` disables protocol logging. The location of the Send connector protocol logs for all Send connectors configured in the Transport service on a

			Mailbox server or on an Edge server is specified with the *SendProtocolLogPath* parameter of the **Set-TransportService** cmdlet.
RequireOorg	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RequireTLS	Optional	System.Boolean	The *RequireTLS* parameter specifies whether all messages sent through this connector must be transmitted using TLS. The default value is `$false`.
SmartHostAuthMechanism	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.SmtpSendConnectorConfig+AuthMechanisms	The *SmartHostAuthMechanism* parameter specifies the smart host authentication mechanism to use for authentication with a remote server. Use this parameter only when a smart host is configured and the *DNSRoutingEnabled* parameter is set to `$false`. Valid values are `None`, `BasicAuth`, `BasicAuthRequireTLS`, `ExchangeServer`, and `ExternalAuthoritative`.

			All values are mutually exclusive. If you select `BasicAuth` or `BasicAuthRequireTLS`, you must use the *AuthenticationCredential* parameter to specify the authentication credential.
SmartHosts	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SmartHosts* parameter specifies the smart hosts the Send connector uses to route mail. This parameteris required if you set the *DNSRoutingEnabled* parameter to `$false` and it must be specified on the same command line. The *SmartHosts* parameter takes one or more FQDNs, such as `server.contoso.com`, or one or more IP addresses, or a combination of both FQDNs and IP addresses. If you enter an IP address, you must enter the IP address as a literal. For example, `10.10.1.1`. The smart host identity can be the FQDN of a smart-host server, a mail exchanger (MX) record, or an address

			(A) record. If you configure an FQDN as the smart host identity, the source server for the Send connector must be able to use DNS name resolution to locate the smart-host server. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
SmtpMaxMessagesPerConnection	Optional	System.Int32	The *SmtpMaxMessagesPerConnection* parameter specifies the maximum number of messages the

			server can send per connection.
SourceIPAddress	Optional	System.Net.IPAddress	The *SourceIPAddress* parameter specifies the local IP address to use as the endpoint for an SMTP connection to a remote messaging server. The default IP address is `0.0.0.0`. This value means that the server can use any available local IP address. This parameter is valid only for Send connectors configured on an Edge server.
SourceTransportServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SourceTransportServers* parameter specifies the names of the Mailbox servers that can use this Send connector. This parameter isn't valid for Send connectors configured on an Edge server. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces

			or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
TlsAuthLevel	Optional	Microsoft.Exchange.Data.TlsAuthLevel	The *TlsAuthLevel* parameter specifies the TLS authentication level that is used for outbound TLS connections established by this Send connector. Valid values are: • `EncryptionOnly`: TLS is used only to encrypt the communication channel. No certificate authentication is performed. • `CertificateValidation`: TLS is used to encrypt the channel and certificate chain validation and revocation lists checks are performed. • `DomainValidation`: In addition to channel

			encryption and certificate validation, the Send connector also verifies that the FQDN of the target certificate matches the domain specified in the *TlsDomain* parameter. If no domain is specified in the *TlsDomain* parameter, the FQDN on the certificate is compared with the recipient's domain. You can't specify a value for this parameter if the *IgnoreSTARTTLS* parameter is set to `$true`, or if the *RequireTLS* parameter is set to `$false`.
TlsCertificateName	Optional	Microsoft.Exchange.Data.SmtpX509Identifier	The *TlsCertificateName* parameter specifies the X.509 certificate to use with TLS sessions and secure mail. Valid input for this parameter is `[I]`*Issuer*`[s]`*Subject*. The *Issuer* value is found in the certificate's `Issuer` field, and the *Subject* value is found in the certificate's `subject` field. You can find these values by running the **Get-ExchangeCertificate**

			cmdlet.
TlsDomain	Optional	Microsoft.Exchange.Data.SmtpDomainWithSubdomains	The *TlsDomain* parameter specifies the domain name that the Send connector uses to verify the FQDN of the target certificate when establishing a TLS secured connection. This parameter is used only if the *TlsAuthLevel* parameter is set to `DomainValidation`. A value for this parameter is required if: • The *TLSAuthLevel* parameter is set to `DomainValidation`. • The *DNSRoutingEnabled* parameter is set to `$false` (smart host Send connector).
Usage	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.NewSendConnector+UsageType	The *Usage* parameter specifies the default permissions and authentication methods assigned to the Send connector. The valid values are as follows: `Custom`, `Internal`, `Internet`, or `Partner`. The default is `Custom`.

			If you use the *Usage* parameter, you can't use any of the following parameters: • *Custom* • *Internal* • *Internet* • *Partner* For more information about Send connector usage types, permissions, and authentication methods, see Send connectors.
UseExternalDNSServersEnabled	Optional	System.Boolean	The *UseExternalDNSServersEnabled* parameter specifies whether this Send connector uses the external DNS list specified by the *ExternalDNSServers* parameter of the **Set-TransportService** cmdlet. The default value is `$false`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what

			changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-SendConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-SendConnector** cmdlet to delete a Send connector.

```
Remove-SendConnector -Identity <SendConnectorIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the Send connector named Contoso.com Send Connector.

```
Remove-SendConnector "Contoso.com Send Connector"
```

Detailed Description

The **Remove-SendConnector** cmdlet deletes the object and the configuration settings for the Send connector.

Caution:
Although a Send connector is configured locally in the Transport service on a Mailbox server or on an Edge server, deleting a Send connector may affect mail flow throughout the organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Send connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.SendConnectorIdParameter	The *Identity* parameter specifies the name, or GUID of the Send connector. If the *Identity* name contains spaces, enclose the name in quotation marks ("). You can omit the *Identity* parameter label. You can also include the server name by using the format *ServerName**ConnectorName*.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default

			when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes

			would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-SendConnector

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-17*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-SendConnector** cmdlet to modify a Send connector.

```
Set-SendConnector -Identity <SendConnectorIdParameter> [-AddressSpaces
<MultiValuedProperty>] [-AuthenticationCredential <PSCredential>] [-
CloudServicesMailEnabled <$true | $false>] [-Comment <String>] [-Confirm
[<SwitchParameter>]] [-ConnectionInactivityTimeOut <EnhancedTimeSpan>] [-
DNSRoutingEnabled <$true | $false>] [-DomainController <Fqdn>] [-
DomainSecureEnabled <$true | $false>] [-Enabled <$true | $false>] [-
ErrorPolicies <Default | DowngradeDnsFailures | DowngradeCustomFailures |
UpgradeCustomFailures>] [-Force <SwitchParameter>] [-ForceHELO <$true |
$false>] [-Fqdn <Fqdn>] [-FrontendProxyEnabled <$true | $false>] [-
IgnoreSTARTTLS <$true | $false>] [-IsScopedConnector <$true | $false>] [-
MaxMessageSize <Unlimited>] [-Name <String>] [-Port <Int32>] [-
ProtocolLoggingLevel <None | Verbose>] [-RequireOorg <$true | $false>] [-
RequireTLS <$true | $false>] [-SmartHostAuthMechanism <None | BasicAuth |
BasicAuthRequireTLS | ExchangeServer | ExternalAuthoritative>] [-
SmartHosts <MultiValuedProperty>] [-SmtpMaxMessagesPerConnection <Int32>]
[-SourceIPAddress <IPAddress>] [-SourceTransportServers
<MultiValuedProperty>] [-TlsAuthLevel <EncryptionOnly |
CertificateValidation | DomainValidation>] [-TlsCertificateName
<SmtpX509Identifier>] [-TlsDomain <SmtpDomainWithSubdomains>] [-
UseExternalDNSServersEnabled <$true | $false>] [-WhatIf
```

```
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example makes the following configuration changes to the Send connector named Contoso.com Send Connector:

- Sets the maximum message size limit to 10 MB.
- Changes the connection inactivity time-out to 15 minutes.

```
Set-SendConnector "Contoso.com Send Connector" -
MaxMessageSize 10MB -ConnectionInactivityTimeOut 00:15:00
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Send connectors" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.SendConnectorIdParameter	The GUID or connector name that represents the Send connector you want to modify.
AddressSpaces	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AddressSpaces* parameter specifies the domain names to which the Send connector routes mail. The complete syntax for entering each address space is as follows: *<AddressSpaceType>:<AddressSpace>;<AddressSpaceCost>*

			• **AddressSpaceType:** On an Edge server, the address space type must be SMTP. In the Transport service on a Mailbox server, the address space type may be SMTP, X400, or any other text string. If you omit the address space type, SMTP is assumed. • **AddressSpace:** For SMTP address space types, the address space that you enter must be RFC 1035-compliant. For example, *, *.com, and *.contoso.com are permitted, but *contoso.com is not. For X.400 address space types, the address space that you enter must be RFC 1685-compliant, such as **o=MySite;p=MyOrg;a=adatum;c=us**. For all other values of address space type, you can enter any text for the address space. • **AddressSpaceCost:** The valid input range for the cost is from 1

			through 100. A lower cost indicates a better route. This parameter is optional. If you omit the address space cost, a cost of 1 is assumed. If you enter a non-SMTP address space that contains the semicolon character (;), you must specify the address space cost. If you specify the address space type or the address space cost, you must enclose the address space in quotation marks ("). For example, the following address space entries are equivalent: • `"SMTP:contoso.com;1"` • `"contoso.com;1"` • `"SMTP:contoso.com"` • `contoso.com` You may specify multiple address spaces by separating the address spaces with commas, for example: `contoso.com,fabrikam.com`. If you specify the address space type or the address space cost, enclose the address space in quotation marks ("), for

			example: `"contoso.com;2","fabrikam.com;3"`. If you specify a non-SMTP address space type on a Send connector configured in the Transport service on a Mailbox server, you must configure the following parameters: • The *SmartHosts* parameter must be set to a value that specifies a smart host. • The *DNSRoutingEnabled* parameter must be set to `$false`. **Note:** Although you can configure non-SMTP address spaces on a Send connector in the Transport service on a Mailbox server, the Send connector uses SMTP as the transport mechanism to send messages to other messaging servers. Foreign connectors in the Transport service on a Mailbox server are used to send messages to local messaging servers, such as third-party fax gateway servers, which don't use SMTP as their primary transport mechanism. For more information, see

			Foreign connectors.
AuthenticationCredential	Optional	System.Management.Automation.PSCredential	The *AuthenticationCredential* parameter specifies a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information about the **Get-Credential** cmdlet, enter **Get-Help Get-Credential** in the Exchange Management Shell.
CloudServicesMailEnabled	Optional	System.Boolean	Set to `$true` to enable this connector to send messages to the cloud service.
Comment	Optional	System.String	The*Comment* parameter specifies an optional comment. You must enclose the *Comment* parameter in quotation marks ("), for example: `"this is an admin note"`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before

			processing continues. You don't have to specify a value with the *Confirm* switch.
ConnectionInactivityTimeOut	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConnectionInactivityTimeOut* parameter specifies the maximum time an idle connection can remain open. The default value is ten minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify fifteen minutes, set it to `00:15:00`. The valid input range for this parameter is `00:00:01` to `1.00:00:00`.
DNSRoutingEnabled	Optional	System.Boolean	The *DNSRoutingEnabled* parameter specifies whether the Send connector uses Domain Name System (DNS) to route mail. Valid values for this parameter are `$true` or `$false`. The default value is `$true`. If

			you specify a *SmartHosts* parameter, the *DNSRoutingEnabled* parameter must be `$false`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
DomainSecureEnabled	Optional	System.Boolean	The *DomainSecureEnabled* parameter is part of the process to enable mutual Transport Layer Security (TLS) authentication for the domains serviced by this Send connector. Mutual TLS authentication

			functions correctly only when the following conditions are met: • The value of the *DomainSecureEnabled* parameter must be `$true`. • The value of the *DNSRoutingEnabled* parameter must be `$true`. • The value of the *IgnoreStartTLS* parameter must be `$false`. The wildcard character (*) is not supported in domains that are configured for mutual TLS authentication. The same domain must also be defined on the corresponding Receive connector and in the *TLSReceiveDomainSecureList* attribute of the transport configuration. The default value for the *DomainSecureEnabled* parameter is `$false` for the following types of Send connectors: • All Send connectors

			defined in the Transport service on a Mailbox server. • User-created Send connectors defined on an Edge server. The default value for the *DomainSecureEnabled* parameter is `$true` for default Send connectors defined on an Edge server.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether to enable the Send connector to process email messages. Valid values are `$true` or `$false`. The default value is `$true`.
ErrorPolicies	Optional	Microsoft.Exchange.Data.ErrorPolicies	The *ErrorPolicies* parameter specifies how communication errors are treated. Possible values are the following: • `Default` A non-delivery report (NDR) is generated for communication errors. • `DowngradeDnsFailures` All DNS errors are treated as transient. • `DowngradeCustomFailures` Particular SMTP errors are treated as transient.

			• `UpgradeCustomFailures` Custom transient failures are upgraded and treated as permanent failures. Multiple values can be specified for this parameter, separated by commas. Specify a value other than `Default` for this parameter only if this Send connector is used to send messages over a reliable and well-defined communication channel where communication errors aren't expected. For example, consider specifying a value other than `Default` if this Send connector is used to send messages to a partner.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in

			the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
ForceHELO	Optional	System.Boolean	The *ForceHELO* parameter specifies whether HELO is sent instead of the default EHLO. Valid values are `$true` or `$false`. The default value is `$false`.
Fqdn	Optional	Microsoft.Exchange.Data.Fqdn	The *Fqdn* parameter specifies the FQDN used as the source server for connected messaging servers that use the Send connector to receive outgoing messages. The value of this parameter is displayed to connected messaging servers whenever a source server name is required, as in the following examples: • In the EHLO/HELO command when the Send connector communicates with the next hop messaging server • In the most recent `Received` header field

			added to the message by the next hop messaging server after the message leaves the Transport service on a Mailbox server or an Edge server • During TLS authentication The default value of the *Fqdn* parameter is `$null`. This means the default FQDN value is the FQDN of the Mailbox server or Edge server that contains the Send connector.
FrontendProxyEnabled	Optional	System.Boolean	The *FrontendProxyEnabled* parameter routes outbound messages through the CAS server, where destination specific routing, such as DNS or IP address, is set, when the parameter is set to `$true`.
IgnoreSTARTTLS	Optional	System.Boolean	The *IgnoreSTARTTLS* parameter specifies whether to ignore the StartTLS option offered by a remote sending server. This parameter is used with remote domains. This

			parameter must be set to `$false` if the *RequireTLS* parameter is set to `$true`. Valid values for this parameter are `$true` or `$false`.
IsScopedConnector	Optional	System.Boolean	The *IsScopedConnector* parameter specifies the availability of the connector to other Mailbox servers with the Transport service. When the value of this parameter is `$false`, the connector can be used by all Mailbox servers in the Exchange organization. When the value of this parameter is `$true`, the connector can only be used by Transport service on Mailbox servers in the same Active Directory site. The default value is `$false`.
MaxMessageSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxMessageSize* parameter specifies the maximum size of a message that can pass through a connector. The default value is 25 MB. When you enter a value,

			qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is 0 to 2147483647 bytes. To remove the message size limit on a Send connector, enter a value of `unlimited`.
Name	Optional	System.String	The *Name* parameter specifies the administrator-supplied name of the connector. You must enclose the *Name* parameter in quotation marks (") if the name contains spaces. For example, `"New Send Connector"`.
Port	Optional	System.Int32	The *Port* parameter specifies the port number for smart host forwarding, if you specify a value in the *SmartHosts* parameter. The valid input range is an integer from 0

			through 65535. The default value is 25. In most organizations, the port number is set to 25.
ProtocolLoggingLevel	Optional	Microsoft.Exchange.Data.ProtocolLoggingLevel	The *ProtocolLoggingLevel* parameter specifies whether to enable protocol logging. `verbose` enables protocol logging. `None` disables protocol logging. The location of the Send connector protocol logs for all Send connectors configured in the Transport service on a Mailbox server or on an Edge server is specified with the *SendProtocolLogPath* parameter of the **Set-TransportService** cmdlet.
RequireOorg	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RequireTLS	Optional	System.Boolean	The *RequireTLS* parameter specifies whether all messages sent through this connector must be transmitted using TLS. The default value is `$false`.
SmartHostAuthMechanism	Optional	Microsoft.Exchange.Data.Directory.SystemCo	The *SmartHostAuthMechanis*

		nfiguration.SmtpSendConnectorConfig+AuthMechanisms	*m* parameter specifies the smart host authentication mechanism to use for authentication with a remote server. Use this parameter only when a smart host is configured and the *DNSRoutingEnabled* parameter is set to `$false`. Valid values are `None`, `BasicAuth`, `BasicAuthRequireTLS`, `ExchangeServer`, and `ExternalAuthoritative`. All values are mutually exclusive. If you select `BasicAuth` or `BasicAuthRequireTLS`, you must use the *AuthenticationCredential* parameter to specify the authentication credential.
SmartHosts	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SmartHosts* parameter specifies the smart hosts the Send connector uses to route mail. This parameteris required if you set the *DNSRoutingEnabled* parameter to `$false` and it must be specified on the same command line. The

			SmartHosts parameter takes one or more FQDNs, such as `server.contoso.com`, or one or more IP addresses, or a combination of both FQDNs and IP addresses. If you enter an IP address, you must enter the IP address as a literal. For example, `10.10.1.1`. The smart host identity can be the FQDN of a smart-host server, a mail exchanger (MX) record, or an address (A) record. If you configure an FQDN as the smart host identity, the source server for the Send connector must be able to use DNS name resolution to locate the smart-host server. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax:

			"<value1>","<value2>"... To add or remove one or more values without affecting any existing entries, use the following syntax: @{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}.
SmtpMaxMessagesPerConnection	Optional	System.Int32	The *SmtpMaxMessagesPerConnection* parameter specifies the maximum number of messages the server can send per connection.
SourceIPAddress	Optional	System.Net.IPAddress	The *SourceIPAddress* parameter specifies the local IP address to use as the endpoint for an SMTP connection to a remote messaging server. The default IP address is 0.0.0.0. This value means that the server can use any available local IP address. This parameter is valid only for Send connectors configured on an Edge server.
SourceTransportServe	Optional	Microsoft.Exchange.Da	The

rs		ta.MultiValuedProperty	*SourceTransportServers* parameter specifies the names of the Mailbox servers that can use this Send connector. This parameter isn't valid for Send connectors configured on an Edge server. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
TlsAuthLevel	Optional	Microsoft.Exchange.Data.TlsAuthLevel	The *TlsAuthLevel* parameter specifies the TLS authentication level that is used for outbound

			TLS connections established by this Send connector. Valid values are: • `EncryptionOnly`: TLS is used only to encrypt the communication channel. No certificate authentication is performed. • `CertificateValidation`: TLS is used to encrypt the channel and certificate chain validation and revocation lists checks are performed. • `DomainValidation`: In addition to channel encryption and certificate validation, the Send connector also verifies that the FQDN of the target certificate matches the domain specified in the *TlsDomain* parameter. If no domain is specified in the *TlsDomain* parameter, the FQDN on the certificate is compared with the recipient's domain. You can't specify a value for this parameter if the *IgnoreSTARTTLS* parameter is set to `$true`, or if the *RequireTLS* parameter is set to `$false`.

TlsCertificateName	Optional	Microsoft.Exchange.Data.SmtpX509Identifier	The *TlsCertificateName* parameter specifies the X.509 certificate to use with TLS sessions and secure mail. Valid input for this parameter is `[I]`*Issuer*`[S]`*Subject*. The *Issuer* value is found in the certificate's `Issuer` field, and the *Subject* value is found in the certificate's `Subject` field. You can find these values by running the **Get-ExchangeCertificate** cmdlet.
TlsDomain	Optional	Microsoft.Exchange.Data.SmtpDomainWithSubdomains	The *TlsDomain* parameter specifies the domain name that the Send connector uses to verify the FQDN of the target certificate when establishing a TLS secured connection. This parameter is used only if the *TlsAuthLevel* parameter is set to `DomainValidation`. A value for this parameter is required if: • The *TLSAuthLevel* parameter is set to

			`DomainValidation`. • The *DNSRoutingEnabled* parameter is set to `$false` (smart host Send connector).
UseExternalDNSServersEnabled	Optional	System.Boolean	The *UseExternalDNSServersEnabled* parameter specifies whether this Send connector uses the external DNS list specified by the *ExternalDNSServers* parameter of the **Set-TransportService** cmdlet. The default value is `$false`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-SmtpConnectivity

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-09*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-SmtpConnectivity** cmdlet to diagnose whether an SMTP connection can successfully be established to the Receive connectors on a specific server. Although you can run this cmdlet manually to verify SMTP connectivity for a specified server, it's primarily used by Microsoft System Center Operations Manager 2007 to test your transport servers' ability to receive SMTP connections to each of the bindings on all the Receive connectors on those servers.

```
Test-SmtpConnectivity [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Identity <ServerIdParameter>] [-MonitoringContext <$true |
$false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example verifies SMTP connectivity for all Receive connectors on the Mailbox server named Mailbox01.

```
Test-SmtpConnectivity Mailbox01
```

EXAMPLE 2

This example verifies SMTP connectivity for all Receive connectors on all Mailbox servers in the organization.

```
Get-TransportService | Test-SmtpConnectivity
```

Detailed Description

When you run the **Test-SmtpConnectivity** cmdlet against a Mailbox server, the cmdlet attempts to establish an SMTP connection to all bindings of all Receive connectors hosted on that server. For each attempt, the cmdlet returns the following information:

- **Server:** The name of the server that hosts the Receive connector.
- **ReceiveConnector:** The name of the Receive connector to which the SMTP connection was attempted.
- **Binding:** The binding that was configured on the Receive connector.
- **EndPoint:** The actual IP address and port to which the SMTP connection was attempted.
- **StatusCode:** The result of the connection attempt. This can be one of the following values: Success, Unable to connect, Transient error, Permanent error, External error.
- **Details:** The actual response received from the server being tested. If the connection attempt isn't successful, this field contains an error string.

The **Test-SmtpConnectivity** results are displayed on-screen. You can write the results to a file by piping the output to **ConvertTo-Html** or **ConvertTo-Csv** and adding "> <filename>" to the command. For example:

```
Test-SmtpConnectivity Mailbox01 | ConvertTo-Csv > "C:\My Documents\SMTP Test.csv"
```

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Testing mail flow" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value

			with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the transport server for which the cmdlet verifies SMTP connectivity. The cmdlet verifies SMTP connectivity for all Receive connectors hosted on the specified server. If no server is specified, the cmdlet attempts to perform the SMTP connectivity

			test against all Receive connectors on the local server.
MonitoringContext	Optional	System.Boolean	The *MonitoringContext* parameter includes or excludes the associated monitoring events and performance counters in the results. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you specify the value `$true`, the monitoring events and performance counters are included in the command results. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SystemMessage

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-SystemMessage** cmdlet to view the delivery status notification (DSN) or quota messages on Mailbox servers or Edge Transport servers.

```
Get-SystemMessage [-Original <SwitchParameter>] <COMMON PARAMETERS>

Get-SystemMessage [-Identity <SystemMessageIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example displays a list of all customized DSN and quota messages configured in your organization.

```
Get-SystemMessage
```

EXAMPLE 2

This example displays detailed configuration information of the specific customized DSN message for DSN code 5.3.2.

```
Get-SystemMessage En\Internal\5.3.2 | Format-List
```

EXAMPLE 3

This example displays detailed configuration information of the specific customized quota message for mailbox size warning.

```
Get-SystemMessage En\WarningMailbox | Format-List
```

EXAMPLE 4

This example displays a list of all built-in DSN and quota messages.

```
Get-SystemMessage -Original
```

Detailed Description

The **Get-SystemMessage** cmdlet displays default and customized DSN and quota messages. You can retrieve a customized DSN or quota message by specifying the identity of the message by using the *Identity* parameter. You can retrieve a list of built-in DSN or quota messages by using the *Original* parameter.

DSN messages are issued to the sender of e-mail messages that haven't reached their intended recipients. Quota messages are issued to users whose mailboxes or public folders have reached the specific warning, prohibit send, or prohibit receive quotas. Customized DSN and quota messages replace the built-in DSN or quota messages included with Exchange.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "DSNs" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.SystemMessageIdParameter	The *Identity* parameter specifies the DSN or quota message you want to view. You can also retrieve a customized DSN message by using the following format: *language**internal | external**system code*. For more information about the syntax of the DSN message identity,

			see DSN message identity. You can also retrieve a customized quota message by using the following format: *language**QuotaMessageType*. *Language* is expressed as the two-character locale code. *QuotaMessageType* accepts the following values: Quota message types related to mailbox size: • `ProhibitSendReceiveMailBox` Issued when a mailbox exceeds its **ProhibitSendReceiveQuota** limit. • `ProhibitSendMailbox` Issued when a mailbox exceeds its **ProhibitSendQuota** limit. • `WarningMailbox` Issued when a mailbox that has a **ProhibitSendQuota** limit or a **ProhibitSendReceiveQuota** limit configured exceeds its **IssueWarningQuota** limit. • `WarningMailboxUnlimitedSize` Issued when a mailbox that doesn't have a **ProhibitSendQuota** limit or a

			ProhibitSendReceiveQuota limit configured exceeds its **IssueWarningQuota** limit. Quota message types related to public folder size: • `ProhibitPostPublicFolder` Issued when a public folder exceeds its **ProhibitPostQuota** limit. • `WarningPublicFolder` Issued when a public folder that has a **ProhibitPostQuota** limit configured exceeds its **IssueWarningQuota** limit. • `WarningPublicFolderUnlimitedSize` Issued when a public folder that doesn't have a **ProhibitPostQuota** limit configured exceeds its **IssueWarningQuota** limit. Quota message types related to the number of messages allowed in a mailbox folder: • `ProhibitReceiveMailboxMessagesPerFolderCount` Issued when a mailbox exceeds its **MailboxMessagesPerFolderCountReceiveQuota** limit. • `WarningMailboxMessage`

			sPerFolderCount Issued when a mailbox that has a **MailboxMessagesPerFolderCountReceiveQuota** limit configured exceeds its **MailboxMessagesPerFolderCountWarningQuota** limit. • WarningMailboxMessagesPerFolderUnlimitedCount Issued when a mailbox that doesn't have a **MailboxMessagesPerFolderCountReceiveQuota** limit configured exceeds its **MailboxMessagesPerFolderCountWarningQuota** limit. Quota message types related to the number of subfolders that can be created in a mailbox folder: • ProhibitReceiveFolderHierarchyChildrenCountCount Issued when a mailbox exceeds its **FolderHierarchyChildrenCountReceiveQuota** limit. • WarningFolderHierarchyChildrenCount Issued when a mailbox that has a **FolderHierarchyChildrenCountReceiveQuota** limit configured exceeds its **FolderHierarchyChildr**

			enCountWarningQuota limit. • `WarningFolderHierarchyChildrenUnlimitedCount` Issued when a mailbox that doesn't have a **FolderHierarchyChildrenCountReceiveQuota** limit configured exceeds its **FolderHierarchyChildrenCountWarningQuota** limit. • `WarningFoldersCount` Issued when the number of folders in the hierarchy exceeds the **FoldersCountWarningQuota** limit. • `ProhibitReceiveFolderscount` Issued when new public folder creation fails because the public folder hierarchy is too big and has reached the **FoldersCountReceiveQuota** limit. • `WarningFoldersCountUnlimited` Issued when the public folder hierarchy is getting too big and does not have the **FoldersCountReceiveQuota** limit configured. Quota message types related to the number of levels allowed in the folder hierarchy of a mailbox folder: • `ProhibitReceiveFolder`

			`HierarchyDepth` Issued when a mailbox exceeds its **FolderHierarchyDepth WarningQuota** limit. • `WarningFolderHierarchyDepth` Issued when a mailbox that has a **FolderHierarchyDepth ReceiveQuota** limit configured exceeds its **FolderHierarchyDepth WarningQuota** limit. • `WarningFolderHierarchyDepthUnlimited` Issued when a mailbox that doesn't have a **FolderHierarchyDepth ReceiveQuota** limit configured exceeds its **FolderHierarchyDepth WarningQuota** limit.
Original	Optional	System.Management.Automation.SwitchParameter	The *Original* parameter specifies whether to retrieve the list of default DSN or quota messages that were included with Exchange when you installed Exchange.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-SystemMessage

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-SystemMessage** cmdlet to create customized delivery status notification (DSN) or quota messages, in the specified language, on Mailbox servers or Edge Transport servers.

```
New-SystemMessage -DsnCode <EnhancedStatusCode> -Internal <$true | $false>
-Language <CultureInfo> -Text <String> <COMMON PARAMETERS>


New-SystemMessage -Language <CultureInfo> -QuotaMessageType
<WarningMailboxUnlimitedSize | WarningPublicFolderUnlimitedSize |
WarningMailbox | WarningPublicFolder | ProhibitSendMailbox |
ProhibitPostPublicFolder | ProhibitSendReceiveMailBox |
WarningMailboxMessagesPerFolderCount |
ProhibitReceiveMailboxMessagesPerFolderCount |
WarningFolderHierarchyChildrenCount |
ProhibitReceiveFolderHierarchyChildrenCountCount |
WarningMailboxMessagesPerFolderUnlimitedCount |
WarningFolderHierarchyChildrenUnlimitedCount | WarningFolderHierarchyDepth
| ProhibitReceiveFolderHierarchyDepth |
WarningFolderHierarchyDepthUnlimited | WarningFoldersCount |
ProhibitReceiveFoldersCount | WarningFoldersCountUnlimited> -Text <String>
<COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a customized DSN message for the DSN code 5.3.5 with the following settings:

- This DSN message is only displayed to external users.
- The text for the DSN message is provided for the English locale.

```
New-SystemMessage -DsnCode 5.3.5 -Language en -Internal
$false -Text "The recipient e-mail system can't process
this e-mail message. Please contact your system
administrator for more information."
```

EXAMPLE 2

This example creates a customized `WarningMailbox` quota message. This message is displayed to users who meet the following criteria:

- The mailbox has exceeded the warning mailbox limit configured on the mailbox.
- The mailbox is located on a server that uses the English locale.

```
New-SystemMessage -QuotaMessageType WarningMailbox -
Language en -Text "Your mailbox has exceeded the warning
limit specified by your e-mail administrator. Please reduce
the size of your mailbox."
```

Detailed Description

DSN messages are issued to the sender of e-mail messages that haven't reached their intended recipients. Quota messages are issued to users whose mailboxes or public folders have reached the specific warning, prohibit send, or prohibit receive quotas. Customized DSN and quota messages replace the built-in DSN or quota messages included with Exchange.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "DSNs" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DsnCode	Required	Microsoft.Exchange.Data.EnhancedStatusCode	The *DsnCode* parameter specifies which DSN code the DSN message applies to. This parameter can be a built-in code, or it can be a customized administrator-defined code. **Note:** This parameter can't be used when the *QuotaMessageType* parameter is specified.
Internal	Required	System.Boolean	The *Internal* parameter specifies whether the message is displayed to

			users inside the Exchange organization. For messages that are only displayed internally, set this parameter to `$true`. For messages that are only displayed to external users, set this parameter to `$false`. **Note:** This parameter can't be used when the *QuotaMessageType* parameter is specified.
Language	Required	System.Globalization.CultureInfo	The *Language* parameter specifies the language of the message. The message can be created in any Exchange-supported language. Languages are specified by using their locale name. For example, the locale name for English is en and the locale name for Japanese is ja. For a complete list of locales, see Supported languages for system messages.
QuotaMessageType	Required	Microsoft.Exchange.Data.Directory.SystemConfiguration.QuotaMessageType	The *QuotaMessageType* parameter selects the type of quota message to create.

			The *QuotaMessageType* parameter accepts the following values: Quota message types related to mailbox size: • `ProhibitSendReceiveMailBox` Issued when a mailbox exceeds its **ProhibitSendReceiveQuota** limit. • `ProhibitSendMailbox` Issued when a mailbox exceeds its **ProhibitSendQuota** limit. • `WarningMailbox` Issued when a mailbox that has a **ProhibitSendQuota** limit or a **ProhibitSendReceiveQuota** limit configured exceeds its **IssueWarningQuota** limit. • `WarningMailboxUnlimitedSize` Issued when a mailbox that doesn't have a **ProhibitSendQuota** limit or a **ProhibitSendReceiveQuota** limit configured exceeds its **IssueWarningQuota** limit. Quota message types related to public folder size: • `ProhibitPostPublicFolder` Issued when a public folder exceeds its

			ProhibitPostQuota limit. • `WarningPublicFolder` Issued when a public folder that has a **ProhibitPostQuota** limit configured exceeds its **IssueWarningQuota** limit. • `WarningPublicFolderUnlimitedSize` Issued when a public folder that doesn't have a **ProhibitPostQuota** limit configured exceeds its **IssueWarningQuota** limit. Quota message types related to the number of messages allowed in a mailbox folder: • `ProhibitReceiveMailboxMessagesPerFolderCount` Issued when a mailbox exceeds its **MailboxMessagesPerFolderCountReceiveQuota** limit. • `WarningMailboxMessagesPerFolderCount` Issued when a mailbox that has a **MailboxMessagesPerFolderCountReceiveQuota** limit configured exceeds its **MailboxMessagesPerFolderCountWarningQuota** limit. • `WarningMailboxMessagesPerFolderUnlimitedCount` Issued when a

			mailbox that doesn't have a **MailboxMessagesPerFolderCountReceiveQuota** limit configured exceeds its **MailboxMessagesPerFolderCountWarningQuota** limit. Quota message types related to the number of subfolders that can be created in a mailbox folder: • `ProhibitReceiveFolderHierarchyChildrenCountCount` Issued when a mailbox exceeds its **FolderHierarchyChildrenCountReceiveQuota** limit. • `WarningFolderHierarchyChildrenCount` Issued when a mailbox that has a **FolderHierarchyChildrenCountReceiveQuota** limit configured exceeds its **FolderHierarchyChildrenCountWarningQuota** limit. • `WarningFolderHierarchyChildrenUnlimitedCount` Issued when a mailbox that doesn't have a **FolderHierarchyChildrenCountReceiveQuota** limit configured exceeds its **FolderHierarchyChildrenCountWarningQuot**

			a limit. • `WarningFoldersCount` Issued when the number of folders in the hierarchy exceeds the **FoldersCountWarningQuota** limit. • `ProhibitReceiveFoldersCount` Issued when new public folder creation fails because the public folder hierarchy is too big and has reached the **FoldersCountReceiveQuota** limit. • `WarningFoldersCountUnlimited` Issued when the public folder hierarchy is getting too big and does not have the **FoldersCountReceiveQuota** limit configured. Quota message types related to the number of levels allowed in the folder hierarchy of a mailbox folder: • `ProhibitReceiveFolderHierarchyDepth` Issued when a mailbox exceeds its **FolderHierarchyDepthWarningQuota** limit. • `WarningFolderHierarchyDepth` Issued when a mailbox that has a **FolderHierarchyDepthReceiveQuota** limit configured exceeds its **FolderHierarchyDepth**

			WarningQuota limit. • `WarningFolderHierarchyDepthUnlimited` Issued when a mailbox that doesn't have a **FolderHierarchyDepthReceiveQuota** limit configured exceeds its **FolderHierarchyDepthWarningQuota** limit. **Note:** This parameter can't be used when the *DsnCode* parameter is specified.
Text	Required	System.String	The *Text* parameter specifies the text of the message displayed to senders or mailbox owners. The text should explain why the message was created and what actions the sender or mailbox owner should take, if any.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Da	The *DomainController*

		ta.Fqdn	parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-SystemMessage

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-SystemMessage** cmdlet to delete customized delivery status notification (DSN) or quota messages on Mailbox servers or Edge Transport servers.

```
Remove-SystemMessage -Identity <SystemMessageIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes a customized DSN message for the DSN code 5.7.9.

```
Remove-SystemMessage En\Internal\5.7.9
```

EXAMPLE 2

This example removes a customized quota message for a mailbox size warning.

```
Remove-SystemMessage En\WarningMailbox
```

Detailed Description

DSN messages are issued to the sender of email messages that haven't reached their intended recipients. Quota messages are issued to users whose mailboxes or public folders have reached the specific warning, prohibit send, or prohibit receive quotas. Customized DSN and quota messages replace the built-in DSN or quota messages included with Exchange.

Note:

Only customized DSN and quota messages can be removed from the server. Built-in DSN and quota messages can't be removed. When a customized DSN or quota message is removed, the message text reverts to the built-in text included with Exchange.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "DSNs" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.SystemMessageIdParameter	The *Identity* parameter specifies the DSN or quota message you want to remove. You can also identify a customized DSN message by using the following format: *language**internal | external**system code*. For more information about the syntax of the DSN message identity, see DSN message identity. You can identify a customized quota message by using the following format: *Language**QuotaMessageType*. *Language* is expressed as the two-character locale code. *QuotaMessageType* accepts the following

			values: Quota message types related to mailbox size: • `ProhibitSendReceiveMailBox` Issued when a mailbox exceeds its **ProhibitSendReceiveQuota** limit. • `ProhibitSendMailbox` Issued when a mailbox exceeds its **ProhibitSendQuota** limit. • `WarningMailbox` Issued when a mailbox that has a **ProhibitSendQuota** limit or a **ProhibitSendReceiveQuota** limit configured exceeds its **IssueWarningQuota** limit. • `WarningMailboxUnlimitedSize` Issued when a mailbox that doesn't have a **ProhibitSendQuota** limit or a **ProhibitSendReceiveQuota** limit configured exceeds its **IssueWarningQuota** limit. Quota message types related to public folder size: • `ProhibitPostPublicFolder` Issued when a public folder exceeds its **ProhibitPostQuota** limit.

			• `WarningPublicFolder` Issued when a public folder that has a **ProhibitPostQuota** limit configured exceeds its **IssueWarningQuota** limit. • `WarningPublicFolderUnlimitedSize` Issued when a public folder that doesn't have a **ProhibitPostQuota** limit configured exceeds its **IssueWarningQuota** limit. Quota message types related to the number of messages allowed in a mailbox folder: • `ProhibitReceiveMailboxMessagesPerFolderCount` Issued when a mailbox exceeds its **MailboxMessagesPerFolderCountReceiveQuota** limit. • `WarningMailboxMessagesPerFolderCount` Issued when a mailbox that has a **MailboxMessagesPerFolderCountReceiveQuota** limit configured exceeds its **MailboxMessagesPerFolderCountWarningQuota** limit. • `WarningMailboxMessagesPerFolderUnlimitedCount` Issued when a mailbox that doesn't have a

			MailboxMessagesPerFolderCountReceiveQuota limit configured exceeds its **MailboxMessagesPerFolderCountWarningQuota** limit. Quota message types related to the number of subfolders that can be created in a mailbox folder: • `ProhibitReceiveFolderHierarchyChildrenCountCount` Issued when a mailbox exceeds its **FolderHierarchyChildrenCountReceiveQuota** limit. • `WarningFolderHierarchyChildrenCount` Issued when a mailbox that has a **FolderHierarchyChildrenCountReceiveQuota** limit configured exceeds its **FolderHierarchyChildrenCountWarningQuota** limit. • `WarningFolderHierarchyChildrenUnlimitedCount` Issued when a mailbox that doesn't have a **FolderHierarchyChildrenCountReceiveQuota** limit configured exceeds its **FolderHierarchyChildrenCountWarningQuota** limit. • `WarningFoldersCount`

			Issued when the number of folders in the hierarchy exceeds the **FoldersCountWarningQuota** limit. • `ProhibitReceiveFoldersCount` Issued when new public folder creation fails because the public folder hierarchy is too big and has reached the **FoldersCountReceiveQuota** limit. • `WarningFoldersCountUnlimited` Issued when the public folder hierarchy is getting too big and does not have the **FoldersCountReceiveQuota** limit configured. Quota message types related to the number of levels allowed in the folder hierarchy of a mailbox folder: • `ProhibitReceiveFolderHierarchyDepth` Issued when a mailbox exceeds its **FolderHierarchyDepthWarningQuota** limit. • `WarningFolderHierarchyDepth` Issued when a mailbox that has a **FolderHierarchyDepthReceiveQuota** limit configured exceeds its **FolderHierarchyDepthWarningQuota** limit. • `WarningFolderHierarchyDepthUnlimited`

			Issued when a mailbox that doesn't have a **FolderHierarchyDepthReceiveQuota** limit configured exceeds its **FolderHierarchyDepthWarningQuota** limit.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write

			data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-SystemMessage

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-SystemMessage** cmdlet to modify existing customized delivery status notification (DSN) or quota messages.

```
Set-SystemMessage -Identity <SystemMessageIdParameter> [-Confirm
```

```
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Name <String>] [-
Original <SwitchParameter>] [-Text <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the English text of an existing customized DSN message that has the DSN code 5.3.5.

```
Set-SystemMessage En\Internal\5.3.5 -Text "The recipient e-
mail system can't process this e-mail message. Please
contact your system administrator for more information."
```

EXAMPLE 2

This example modifies the English text of an existing customized `WarningMailbox` quota message.

```
Set-SystemMessage En\WarningMailbox -Text "Your mailbox has
exceeded the warning limit specified by your e-mail
administrator. Please reduce the size of your mailbox."
```

Detailed Description

DSN messages are issued to the sender of e-mail messages that haven't reached their intended recipients. Quota messages are issued to users whose mailboxes or public folders have reached the specific warning, prohibit send, or prohibit receive quotas. Customized DSN and quota messages replace the built-in DSN or quota messages included with Exchange.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "DSNs" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.SystemMessageIdParameter	The*Identity* parameter specifies the identity of the DSN or quota message to modify. To modify a DSN

			message, use the following format: *language*\internal | external*system code*. For more information about the syntax of the DSN message identity, see DSN message identity. To modify a customized quota message, use the following format: *language* *QuotaMessageType*. *Language* is expressed as the two-character locale code. The *QuotaMessageType* parameter accepts the following values: Quota message types related to mailbox size: • `ProhibitSendReceiveMailBox` Issued when a mailbox exceeds its **ProhibitSendReceiveQuota** limit. • `ProhibitSendMailbox` Issued when a mailbox exceeds its **ProhibitSendQuota** limit. • `WarningMailbox` Issued when a mailbox that has a **ProhibitSendQuota** limit or a **ProhibitSendReceiveQuota** limit configured

			exceeds its **IssueWarningQuota** limit. • `WarningMailboxUnlimitedSize` Issued when a mailbox that doesn't have a **ProhibitSendQuota** limit or a **ProhibitSendReceiveQuota** limit configured exceeds its **IssueWarningQuota** limit. Quota message types related to public folder size: • `ProhibitPostPublicFolder` Issued when a public folder exceeds its **ProhibitPostQuota** limit. • `WarningPublicFolder` Issued when a public folder that has a **ProhibitPostQuota** limit configured exceeds its **IssueWarningQuota** limit. • `WarningPublicFolderUnlimitedSize` Issued when a public folder that doesn't have a **ProhibitPostQuota** limit configured exceeds its **IssueWarningQuota** limit. Quota message types related to the number of messages allowed in a

			mailbox folder: • `ProhibitReceiveMailboxMessagesPerFolderCount` Issued when a mailbox exceeds its **MailboxMessagesPerFolderCountReceiveQuota** limit. • `WarningMailboxMessagesPerFolderCount` Issued when a mailbox that has a **MailboxMessagesPerFolderCountReceiveQuota** limit configured exceeds its **MailboxMessagesPerFolderCountWarningQuota** limit. • `WarningMailboxMessagesPerFolderUnlimitedCount` Issued when a mailbox that doesn't have a **MailboxMessagesPerFolderCountReceiveQuota** limit configured exceeds its **MailboxMessagesPerFolderCountWarningQuota** limit. Quota message types related to the number of subfolders that can be created in a mailbox folder: • `ProhibitReceiveFolderHierarchyChildrenCountCount` Issued when a mailbox exceeds its **FolderHierarchyChildrenCountReceiveQuota** limit.

			• `WarningFolderHierarchyChildrenCount` Issued when a mailbox that has a **FolderHierarchyChildrenCountReceiveQuota** limit configured exceeds its **FolderHierarchyChildrenCountWarningQuota** limit. • `WarningFolderHierarchyChildrenUnlimitedCount` Issued when a mailbox that doesn't have a **FolderHierarchyChildrenCountReceiveQuota** limit configured exceeds its **FolderHierarchyChildrenCountWarningQuota** limit. • `WarningFoldersCount` Issued when the number of folders in the hierarchy exceeds the **FoldersCountWarningQuota** limit. • `ProhibitReceiveFoldersCount` Issued when new public folder creation fails because the public folder hierarchy is too big and has reached the **FoldersCountReceiveQuota** limit. • `WarningFoldersCountUnlimited` Issued when the public folder hierarchy is getting too big and does not have the **FoldersCountReceive**

			Quota limit configured. Quota message types related to the number of levels allowed in the folder hierarchy of a mailbox folder: • `ProhibitReceiveFolderHierarchyDepth` Issued when a mailbox exceeds its **FolderHierarchyDepthWarningQuota** limit. • `WarningFolderHierarchyDepth` Issued when a mailbox that has a **FolderHierarchyDepthReceiveQuota** limit configured exceeds its **FolderHierarchyDepthWarningQuota** limit. • `WarningFolderHierarchyDepthUnlimited` Issued when a mailbox that doesn't have a **FolderHierarchyDepthReceiveQuota** limit configured exceeds its **FolderHierarchyDepthWarningQuota** limit.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Name	Optional	System.String	The *Name* parameter specifies a name that you can use to describe the DSN or quota message.
Original	Optional	System.Management.Automation.SwitchParameter	The *Original* parameter reverts the DSN or quota message text back to the original shipped text. This parameter is useful when you don't want to remove the customized DSN or quota entry but want to revert to the original text.

Text	Optional	System.String	The *Text* parameter specifies the text of the message displayed to senders or mailbox owners. The text should explain why the message was created and what actions the sender or mailbox owner should take, if any.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-TransportAgent

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-TransportAgent** cmdlet to disable a transport agent.

```
Disable-TransportAgent -Identity <TransportAgentObjectId> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-TransportService <Hub |
Edge | FrontEnd | MailboxSubmission | MailboxDelivery>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example shows how a fictitious application named Test App is disabled in the Transport service on a Mailbox server.

```
Disable-TransportAgent "Test App" -TransportService Hub
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport agents" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.AgentTasks.TransportAgentObjectId	The *Identity* parameter specifies the display name of the transport agent to be disabled. The length of the name can't exceed 64 characters.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
TransportService	Optional	Microsoft.Exchange.Data.TransportService	The *TransportService* parameter specifies the transport service that you want to view or modify.

			Valid values for this parameter are: • `Hub` for the Transport service on Mailbox servers. • `MailboxSubmission` for the Mailbox Transport Submission service on Mailbox servers. • `MailboxDelivery` for the Mailbox Transport Delivery service on Mailbox servers. • `FrontEnd` for the Front End Transport service on Client Access servers. • `Edge` on Edge Transport servers.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-TransportAgent

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-TransportAgent** cmdlet to enable a transport agent.

```
Enable-TransportAgent -Identity <TransportAgentObjectId> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-TransportService <Hub |
Edge | FrontEnd | MailboxSubmission | MailboxDelivery>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables a fictitious application named Test App in the Transport service on a Mailbox server.

```
Enable-TransportAgent "Test App" -TransportService Hub
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport agents" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.AgentTasks.TransportAgentObje	The *Identity* parameter specifies the display name of the transport agent to

		ctId	be enabled. The length of the name can't exceed 64 characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

TransportService	Optional	Microsoft.Exchange.Data.TransportService	The *TransportService* parameter specifies the transport service that you want to view or modify. Valid values for this parameter are: • `Hub` for the Transport service on Mailbox servers. • `MailboxSubmission` for the Mailbox Transport Submission service on Mailbox servers. • `MailboxDelivery` for the Mailbox Transport Delivery service on Mailbox servers. • `FrontEnd` for the Front End Transport service on Client Access servers. • `Edge` on Edge Transport servers.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-TransportAgent

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-TransportAgent** cmdlet to view the configuration of a transport agent.

```
Get-TransportAgent [-Identity <TransportAgentObjectId>] [-DomainController
<Fqdn>] [-TransportService <Hub | Edge | FrontEnd | MailboxSubmission |
MailboxDelivery>]
```

Examples

EXAMPLE 1

This example displays a summary list of all transport agents installed on all Exchange servers in your organization.

```
Get-TransportAgent
```

EXAMPLE 2

This example displays detailed information about the Transport Rule agent that's installed in the Transport service on a Mailbox server. The output of the **Get-TransportAgent** command is piped to the **Format-List** command to display the detailed configuration of the transport agent.

```
Get-TransportAgent "Transport Rule Agent" -TransportService
Hub | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport agents" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Management.AgentTasks.TransportAgentObjectId	The *Identity* parameter specifies the display name of the transport agent to be displayed. The length of the name can't exceed 64 characters.
TransportService	Optional	Microsoft.Exchange.Data.TransportService	The *TransportService* parameter specifies the

			transport service that you want to view or modify. Valid values for this parameter are: • `Hub` for the Transport service on Mailbox servers. • `MailboxSubmission` for the Mailbox Transport Submission service on Mailbox servers. • `MailboxDelivery` for the Mailbox Transport Delivery service on Mailbox servers. • `FrontEnd` for the Front End Transport service on Client Access servers. • `Edge` on Edge Transport servers.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Install-TransportAgent

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Install-TransportAgent** cmdlet to register a transport agent in the Transport service on a Mailbox server, in the Front End Transport service on a Client Access server, or on an Edge Transport server.

```
Install-TransportAgent -AssemblyPath <String> -Name <String> -
TransportAgentFactory <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-TransportService <Hub | Edge | FrontEnd |
MailboxSubmission | MailboxDelivery>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example shows how a fictitious application named Test App is installed in the Transport service on a Mailbox server.

```
Install-TransportAgent -Name "Test App" -
TransportAgentFactory
"vendor.exchange.avTransportAgentfactory" -AssemblyPath "c:
\Program Files\Vendor\TransportAgent
\AvTransportAgentFactory.dll" -TransportService Hub
```

Detailed Description

Caution:
Transport agents have full access to all email messages that they encounter. Exchange puts no restrictions on a transport agent's behavior. Transport agents that are unstable or contain security flaws may affect the stability and security of Exchange. Therefore, you must only install transport agents that you fully trust and that have been fully tested in a test environment.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport agents" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
AssemblyPath	Required	System.String	The *AssemblyPath* parameter specifies the location of the transport agent Microsoft .NET

			assembly. Universal Naming Convention (UNC) file paths can't be used.
Name	Required	System.String	The *Name* parameter specifies the display name of the transport agent to be installed. The length of the name can't exceed 64 characters.
TransportAgentFactory	Required	System.String	The *TransportAgentFactory* parameter specifies the Microsoft .NET class type of the transport agent factory. The developer of the transport agent being installed provides the transport agent factory and related information. For more information, see the documentation provided by the developer of the transport agent.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You

			don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
TransportService	Optional	Microsoft.Exchange.Data.TransportService	The *TransportService* parameter specifies the transport service that you want to view or modify. Valid values for this parameter are: • `Hub` for the Transport service on Mailbox servers. • `MailboxSubmission` for the Mailbox Transport Submission service on Mailbox servers.

			• `MailboxDelivery` for the Mailbox Transport Delivery service on Mailbox servers. • `FrontEnd` for the Front End Transport service on Client Access servers. • `Edge` on Edge Transport servers.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-TransportAgent

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-TransportAgent** cmdlet to modify a transport agent.

```
Set-TransportAgent -Identity <TransportAgentObjectId> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Priority <Int32>] [-
TransportService <Hub | Edge | FrontEnd | MailboxSubmission |
MailboxDelivery>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the priority of a fictitious agent named Test App in the Front End Transport service on a Client Access server.

```
Set-TransportAgent "Test App" -Priority 3 -TransportService
-FrontEnd
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport agents" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.AgentTasks.TransportAgentObjectId	The *Identity* parameter specifies the display name of the transport agent to be modified. The length of the name can't exceed 64 characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Priority	Optional	System.Int32	The *Priority* parameter specifies the priority of the transport agent. The priority of the transport agent controls the order in which the transport agents process email messages. The priority

			must be a value between 0 and the maximum number of transport agents. The default behavior is to append a new transport agent to the end of the priority list. Transport agents with a priority closest to 0 process email messages first.
TransportService	Optional	Microsoft.Exchange.Data.TransportService	The *TransportService* parameter specifies the transport service that you want to view or modify. Valid values for this parameter are: • `Hub` for the Transport service on Mailbox servers. • `MailboxSubmission` for the Mailbox Transport Submission service on Mailbox servers. • `MailboxDelivery` for the Mailbox Transport Delivery service on Mailbox servers. • `FrontEnd` for the Front End Transport service on Client Access servers. • `Edge` on Edge Transport servers.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it

			would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Uninstall-TransportAgent

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Uninstall-TransportAgent** cmdlet to unregister a transport agent from the Transport service on a Mailbox server, the Front End Transport service on a Client Access server, or from an Edge Transport server.

```
Uninstall-TransportAgent -Identity <TransportAgentObjectId> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-TransportService <Hub |
Edge | FrontEnd | MailboxSubmission | MailboxDelivery>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example uninstalls a fictitious application named Test App from the Transport service on a Mailbox server.

```
Uninstall-TransportAgent "Test App" -TransportService Hub
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport agents" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.AgentTasks.TransportAgentObjectId	The *Identity* parameter specifies the display name of the transport agent to be uninstalled. The length of the name can't exceed 64 characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
TransportService	Optional	Microsoft.Exchange.Data.TransportService	The *TransportService* parameter specifies the transport service that you want to view or modify. Valid values for this parameter are: • `Hub` for the Transport service on Mailbox servers. • `MailboxSubmission` for the Mailbox Transport Submission service on Mailbox servers. • `MailboxDelivery` for the Mailbox Transport Delivery service on Mailbox servers. • `FrontEnd` for the Front End Transport service on Client Access servers. • `Edge` on Edge Transport

			servers.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-TransportConfig

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-TransportConfig** cmdlet to view organization-wide transport configuration settings.

```
Get-TransportConfig [-Identity <OrganizationIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example lists the organization-wide transport settings on Mailbox server, or the local transport settings on an Edge Transport server.

```
Get-TransportConfig
```

EXAMPLE 2

This example lists all delivery status notification-related (DSN) configuration settings for your organization when run on a Mailbox server. When run on an Edge Transport server, it displays the DSN-related settings configured on that Edge Transport server.

```
Get-TransportConfig | Format-List *DSN*
```

Detailed Description

The **Get-TransportConfig** cmdlet displays configuration information for global transport settings applied across the organization when the cmdlet is run on a Mailbox server. When this cmdlet is run on an Edge Transport server, only the transportation configuration settings for the local computer are shown.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport configuration" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-TransportConfig

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Set-TransportConfig** cmdlet to modify the transport configuration settings for the whole Exchange organization.

```
Set-TransportConfig [-Identity <OrganizationIdParameter>] [-
AddressBookPolicyRoutingEnabled <$true | $false>] [-
AgentGeneratedMessageLoopDetectionInSmtpEnabled <$true | $false>] [-
AgentGeneratedMessageLoopDetectionInSubmissionEnabled <$true | $false>] [-
AnonymousSenderToRecipientRatePerHour <Int32>] [-ClearCategories <$true |
$false>] [-Confirm [<SwitchParameter>]] [-ConvertDisclaimerWrapperToEml
<$true | $false>] [-DiagnosticsAggregationServicePort <Int32>] [-
DomainController <Fqdn>] [-DSNConversionMode <UseExchangeDSNs |
PreserveDSNBody | DoNotConvert>] [-ExternalDelayDsnEnabled <$true |
$false>] [-ExternalDsnDefaultLanguage <CultureInfo>] [-
ExternalDsnLanguageDetectionEnabled <$true | $false>] [-
ExternalDsnMaxMessageAttachSize <ByteQuantifiedSize>] [-
ExternalDsnReportingAuthority <SmtpDomain>] [-ExternalDsnSendHtml <$true |
$false>] [-ExternalPostmasterAddress <SmtpAddress>] [-GenerateCopyOfDSNFor
<MultiValuedProperty>] [-HeaderPromotionModeSetting <NoCreate | MayCreate
| MustCreate>] [-HygieneSuite <Standard | Premium>] [-
InternalDelayDsnEnabled <$true | $false>] [-InternalDsnDefaultLanguage
<CultureInfo>] [-InternalDsnLanguageDetectionEnabled <$true | $false>] [-
InternalDsnMaxMessageAttachSize <ByteQuantifiedSize>] [-
InternalDsnReportingAuthority <SmtpDomain>] [-InternalDsnSendHtml <$true |
$false>] [-InternalSMTPServers <MultiValuedProperty>] [-
JournalArchivingEnabled <$true | $false>] [-JournalingReportNdrTo
<SmtpAddress>] [-JournalReportDLMemberSubstitutionEnabled <$true |
$false>] [-LegacyArchiveJournalingEnabled <$true | $false>] [-
LegacyArchiveLiveJournalingEnabled <$true | $false>] [-
LegacyJournalingMigrationEnabled <$true | $false>] [-
MaxAllowedAgentGeneratedMessageDepth <UInt32>] [-
MaxAllowedAgentGeneratedMessageDepthPerAgent <UInt32>] [-
MaxDumpsterSizePerDatabase <ByteQuantifiedSize>] [-MaxDumpsterTime
<EnhancedTimeSpan>] [-MaxReceiveSize <Unlimited>] [-
MaxRecipientEnvelopeLimit <Unlimited>] [-MaxRetriesForLocalSiteShadow
<Int32>] [-MaxRetriesForRemoteSiteShadow <Int32>] [-MaxSendSize
<Unlimited>] [-MigrationEnabled <$true | $false>] [-
OpenDomainRoutingEnabled <$true | $false>] [-OrganizationFederatedMailbox
<SmtpAddress>] [-QueueDiagnosticsAggregationInterval <EnhancedTimeSpan>]
[-RedirectDLMessagesForLegacyArchiveJournaling <$true | $false>] [-
RedirectUnprovisionedUserMessagesForLegacyArchiveJournaling <$true |
$false>] [-RejectMessageOnShadowFailure <$true | $false>] [-
Rfc2231EncodingEnabled <$true | $false>] [-SafetyNetHoldTime
<EnhancedTimeSpan>] [-ShadowHeartbeatFrequency <EnhancedTimeSpan>] [-
ShadowHeartbeatRetryCount <Int32>] [-ShadowHeartbeatTimeoutInterval
<EnhancedTimeSpan>] [-ShadowMessageAutoDiscardInterval <EnhancedTimeSpan>]
[-ShadowMessagePreferenceSetting <PreferRemote | LocalOnly | RemoteOnly>]
[-ShadowRedundancyEnabled <$true | $false>] [-ShadowResubmitTimeSpan
<EnhancedTimeSpan>] [-SupervisionTags <MultiValuedProperty>] [-
TLSReceiveDomainSecureList <MultiValuedProperty>] [-
TLSSendDomainSecureList <MultiValuedProperty>] [-
TransportRuleAttachmentTextScanLimit <ByteQuantifiedSize>] [-
TransportRuleCollectionAddedRecipientsLimit <Int32>] [-
TransportRuleCollectionRegexCharsLimit <ByteQuantifiedSize>] [-
TransportRuleLimit <Int32>] [-TransportRuleMinProductVersion <Version>] [-
TransportRuleRegexValidationTimeout <EnhancedTimeSpan>] [-
TransportRuleSizeLimit <ByteQuantifiedSize>] [-VerifySecureSubmitEnabled
<$true | $false>] [-VoicemailJournalingEnabled <$true | $false>] [-WhatIf
[<SwitchParameter>]] [-Xexch50Enabled <$true | $false>]
```

Examples

EXAMPLE 1

This example configures the Exchange organization to forward all DSN messages that have the DSN codes 5.7.1, 5.7.2, and 5.7.3 to the postmaster email account.

```
Set-TransportConfig -GenerateCopyOfDSNFor 5.7.1,5.7.2,5.7.3
```

EXAMPLE 2

This example configures the Exchange organization to redirect all journaling reports that can't be delivered to the journaling mailbox to the email account `journalingndr@contoso.com`.

```
Set-TransportConfig -JournalingReportNdrTo
journalingndr@contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport configuration" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
AddressBookPolicyRoutingEnabled	Optional	System.Boolean	The *AddressBookPolicyRoutingEnabled* parameter controls how recipients are resolved in an organization that uses address book policies to create separate virtual organizations within the same Exchange organization. Specifically, the global address list (GAL) that's specified in the user's address book policy controls how recipients are resolved. When the value of this parameter is `$true`, users

			that are assigned different GALs appear as external recipients. When the value of this parameter is `$false`, users that are assigned different GALs appear as internal recipients. The default value is `$false`. Note that this parameter has no effect if your organization doesn't use address book policies, or if the address book policy routing agent isn't installed and enabled. Also note that changing the value of this parameter may take up to 30 minutes to take effect. For more information about address book policies, see Address book policies.
AgentGeneratedMessageLoopDetectionInSmtpEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *AgentGeneratedMessageLoopDetectionInSmtpEnabled* parameter controls the behavior of messages

			loop detection in for loops caused by transport agents in the Transport service. An agent-generated loop occurs when an agent creates a new copy of a message or adds recipients to a message, and the agent continues to process these resulting messages by creating copies or adding recipients. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When Exchange detects an agent-generated message loop, the loop is stopped. When this parameter is set to `$false`, the loop is logged in the message tracking log. When this parameter is set to `$true`, the message is rejected with an NDR when the loop generates the number of messages specified by the *MaxAllowedAgentGenerat*

			edMessageDepth and *MaxAllowedAgentGeneratedMessageDepthPerAgent* parameters.
AgentGeneratedMessageLoopDetectionInSubmissionEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *AgentGeneratedMessageLoopDetectionInSubmissionEnabled* parameter controls the behavior of messages loop detection in for loops caused by transport agents in the Mailbox Transport Submission service. An agent-generated loop occurs when an agent creates a new copy of a message or adds recipients to a message, and the agent continues to process these resulting messages by creating copies or adding recipients. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. When Exchange detects

			an agent-generated message loop, the loop is stopped. When this parameter is set to `$true`, the loop is logged in the message tracking log. When this parameter is set to `$false`, the message is rejected with an NDR when the loop generates the number of messages specified by the *MaxAllowedAgentGeneratedMessageDepthPerAgent* and *MaxAllowedAgentGeneratedMessageDepth* parameters.
AnonymousSenderToRecipientRatePerHour	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
ClearCategories	Optional	System.Boolean	The *ClearCategories* parameter keeps or removes Microsoft Outlook message categories during content conversion. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. This means that by default, Outlook message categories are

			removed during content conversion.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ConvertDisclaimerWrapperToEml	Optional	System.Boolean	The *ConvertDisclaimerWrapperToEml* parameter specifies whether the original message will be added as a TNEF attachment or a regular EML attachment to a disclaimer when all of the following are true: • Message is sent to an external user. • The sender has signed the message. • The message is processed by a Transport rule that adds a disclaimer. When a Transport rule that adds disclaimers to

			outbound messages encounters a message signed by the sender, the Transport rule can't add the disclaimer directly to the message. As a result, the disclaimer is sent to the intended recipient with the original message as an attachment. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you set this parameter to `$true`, the original message is sent as an EML attachment. Otherwise, it is sent as a TNEF attachment.
DiagnosticsAggregationServicePort	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *DiagnosticsAggregationServicePort* parameter specifies the TCP port that's used to collect message queue diagnostic information. The default value is 9710.
DomainController	Optional	Microsoft.Exchange.Da	This parameter is available only in on-

		ta.Fqdn	premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
DSNConversionMode	Optional	Microsoft.Exchange.Data.DSNConversionOption	The *DSNConversionMode* parameter controls how Exchange handles delivery status notifications (DSNs) that are generated by earlier versions of Exchange or non-Exchange messaging systems. You can specify one of the following values for this parameter: • `UseExchangeDSNs` • `PreserveDSNBody` • `DoNotConvert`

			By default, this parameter is set to `UseExchangeDSNs` and Exchange converts the DSNs to the Exchange 2013 DSN format, which is the same as the Exchange 2010 DSN format. Any customized text or attachments that were associated with the original DSN are overwritten. If you set this parameter to `PreserveDSNBody`, Exchange converts the DSNs to Exchange 2013 DSN format. However, the text in the body of the DSN message is retained. If you set this parameter to `DoNotConvert`, Exchange does not modify the DSN message. Instead, Exchange delivers the message as a standard message.
ExternalDelayDsnEnabled	Optional	System.Boolean	The *ExternalDelayDsnEnabled* parameter specifies whether a delay delivery status notification (DSN) message should be

			created for external messages that couldn't be immediately delivered. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
ExternalDsnDefaultLanguage	Optional	System.Globalization.CultureInfo	The *ExternalDsnDefaultLanguage* parameter specifies which Exchange server language should be used by default when you create external DSN messages. The default value is the default Windows server language.
ExternalDsnLanguageDetectionEnabled	Optional	System.Boolean	The *ExternalDsnLanguageDetectionEnabled* parameter specifies whether the server should try to send an external DSN message in the same language as the original message that generated the notification. Valid input for this parameter is `$true` or `$false`.The default value is `$true`.
ExternalDsnMaxMessa	Optional	Microsoft.Exchange.Da	The *ExternalDsnMaxMessageA*

geAttachSize		ta.ByteQuantifiedSize	*ttachSize* parameter specifies the maximum size of the original message attached to an external DSN message. If the original message exceeds this size, only the headers of the original message are included in the DSN message. The default value is 10 megabytes (MB). When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 0 through 2147483647 bytes. If you specify a value of 0, only the original message headers are included in the external DSN message.
ExternalDsnReporting Authority	Optional	Microsoft.Exchange.Da ta.SmtpDomain	The *ExternalDsnReportingAuth ority* parameter specifies

			what the server name should be in the machine-readable part of the external DSN message. The default value is the authoritative domain specified during installation.
ExternalDsnSendHtml	Optional	System.Boolean	The *ExternalDsnSendHtml* parameter specifies whether external DSN messages should be sent by using HTML or whether messages should be sent in plain text. Valid input for this parameter is `$true` or `$false`.The default value is `$true`.
ExternalPostmasterAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *ExternalPostmasterAddress* parameter specifies the email address in the `From` header field of an external DSN message. The default value is `$null`. In the Transport service on a Mailbox server, the value of the external postmaster email address is `postmaster@<defaultacceptedomain>`. If an Edge Transport server hasn't

			yet been through the EdgeSync process, and the *ExternalPostmasterAddress* parameter is set to `$null`, the external postmaster email address on the Edge Transport server is `postmaster@<edgetransportserverfqdn>`. If an Edge Transport server has completed the EdgeSync process, and the *ExternalPostmasterAddress* parameter is set to `$null`, the external postmaster email address on the Edge Transport server is `postmaster@<defaultacceptedomain>`. To override the default behavior, you can specify an email address for the *ExternalPostMasterAddress* parameter.
GenerateCopyOfDSNFor	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *GenerateCopyOfDSNFor* parameter controls the

			non-delivery reports (NDRs) that are copied to a mailbox by specifying the DSN codes that you want to monitor. You must configure the list of monitored DSNs on one Mailbox server and locally on each Edge Transport server in your Exchange organization. On a Mailbox server, NDRs are copied to the mailbox assigned to the Exchange recipient. On Edge Transport servers, NDRs are copied to the mailbox assigned to the external postmaster address. DSN codes are entered as *x.y.z* and are separated by commas. By default, the following DSN codes are monitored: • 5.4.8 • 5.4.6 • 5.4.4 • 5.2.4 • 5.2.0 • 5.1.4 To enter multiple values

			and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. Although these DSN codes are monitored by default, the associated NDRs aren't copied to the Exchange recipient or to the external postmaster address if no mailbox is assigned to the Exchange recipient or to the external postmaster address. By default, no mailbox is assigned to the Exchange recipient or to the external postmaster address.

			To assign a mailbox to the Exchange recipient, use the **Set-OrganizationConfig** cmdlet with the *MicrosoftExchangeRecipientReplyRecipient* parameter. To assign a mailbox to the external postmaster address, create a new mailbox postmaster. The default email address policy of the Exchange organization should automatically add an SMTP address of postmaster@<*Authoritative domain*> to the mailbox.
HeaderPromotionModeSetting	Optional	Microsoft.Exchange.Data.HeaderPromotionMode	The *HeaderPromotionModeSetting* parameter specifies whether named properties are created for custom X-headers on messages received from outside the Exchange organization. You can use one of the following values: • `MustCreate` Exchange creates a named property for each new custom X-header.

			• `MayCreate` Exchange creates a named property for each new custom X-header on messages received from authenticated senders. No named properties are created for custom X-headers on messages received from unauthenticated senders. • `NoCreate` Exchange won't create any named properties based on custom X-headers on incoming messages.
HygieneSuite	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.HygieneSuiteEnum	This parameter is reserved for internal Microsoft use.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
InternalDelayDsnEnabled	Optional	System.Boolean	The *InternalDelayDsnEnabled* parameter specifies whether a delay DSN message should be created for messages sent to or from recipients or senders in the same Exchange organization that couldn't be immediately delivered. Valid input for this

			parameter is `$true` or `$false`. The default value is `$true`.
InternalDsnDefaultLanguage	Optional	System.Globalization.CultureInfo	The *InternalDsnDefaultLanguage* parameter specifies which Exchange server language should be used by default when you create internal DSN messages. The default value is the default Windows server language.
InternalDsnLanguageDetectionEnabled	Optional	System.Boolean	The *InternalDsnLanguageDetectionEnabled* parameter specifies whether the server should try to send an internal DSN message in the same language as the original message that generated the notification. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
InternalDsnMaxMessageAttachSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *InternalDsnMaxMessageAttachSize* parameter specifies the maximum size of the original message that generated

			an internal DSN message. If the original message exceeds this size, only the headers of the original message are included in the DSN message. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 0 through 2147483647 bytes. If you specify a value of 0, only the original message headers are included in the internal DSN message.
InternalDsnReporting Authority	Optional	Microsoft.Exchange.Data.SmtpDomain	The *InternalDsnReportingAuthority* parameter specifies what the server name should be in the internal DSN message. The default value is the authoritative domain specified during

			installation.
InternalDsnSendHtml	Optional	System.Boolean	The *InternalDsnSendHtml* parameter specifies whether internal DSN messages should be sent by using HTML or whether messages should be sent in plain text. Valid input for this parameter is `$true` or `$false`. The default is `$true`.
InternalSMTPServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *InternalSMTPServers* parameter specifies a list of internal SMTP server IP addresses or IP address ranges that should be ignored by Sender ID and connection filtering. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>".`

			... To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
JournalArchivingEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
JournalingReportNdrTo	Optional	Microsoft.Exchange.Data.SmtpAddress	The *JournalingReportNdrTo* parameter specifies the email address to which journal reports are sent if the journaling mailbox is unavailable. By default, if this parameter is left empty, Exchange continues to try to deliver the journal report to the journaling mailbox.
JournalReportDLMemberSubstitutionEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LegacyArchiveJournalingEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LegacyArchiveLiveJournalingEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.

LegacyJournalingMigrationEnabled	Optional	System.Boolean	The *LegacyJournalingMigrationEnabled* parameter specifies whether journal messages generated in Microsoft Exchange Server 2003 will be reformatted by the current version of Exchange. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
MaxAllowedAgentGeneratedMessageDepth	Optional	System.UInt32	This parameter is available only in on-premises Exchange 2013. The *MaxAllowedAgentGeneratedMessageDepth* parameter specifies how many times all agents can process any resulting copies of the same message. The default value is 3. Valid input for this parameter is an integer.
MaxAllowedAgentGeneratedMessageDepthPerAgent	Optional	System.UInt32	This parameter is available only in on-premises Exchange 2013.

			The *MaxAllowedAgentGeneratedMessageDepthPerAgent* parameter specifies how many times a single agent can process any resulting copies of the same message. The default value is 2. The value of the *MaxAllowedAgentGeneratedMessageDepth* parameter should be larger than the value of the *MaxAllowedAgentGeneratedMessageDepthPerAgent* parameter.
MaxDumpsterSizePerDatabase	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is available only in on-premises Exchange 2013. This parameter isn't used by Microsoft Exchange Server 2013. It's only used by Microsoft Exchange 2010 servers in a coexistence environment. The *MaxDumpsterSizePerDatabase* parameter specifies the maximum size of the transport dumpster on a

			Hub Transport server for each database. The default value is 18 MB. The valid input range for this parameter is from 0 through 2147483647 KB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. To enable the transport dumpster, the value of the *MaxDumpsterSizePerDatabase* parameter must be greater than 0, and the value of the *MaxDumpsterTime* parameter must be greater than 00:00:00. This parameter has no replacement in Exchange 2013.
MaxDumpsterTime	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is available only in on-premises Exchange 2013. This parameter isn't used

			by Microsoft Exchange Server 2013. It's only used by Microsoft Exchange 2010 servers in a coexistence environment. The *MaxDumpsterTime* parameter specifies how long an email message should remain in the transport dumpster on a Hub Transport server. The default value is seven days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 10 days for this parameter, use 10.00:00:00. The valid input range for this parameter is 00:00:00 to 24855.03:14:07. To enable the transport dumpster, the value of the *MaxDumpsterSizePerStorageGroup* parameter must be greater than 0, and the value of the *MaxDumpsterTime*

			parameter must be greater than 00:00:00. This parameter is replaced by the *SafetyNetHoldTime* parameter.
MaxReceiveSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxReceiveSize* parameter specifies the maximum message size that can be received by recipients in the organization. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 0 through 2097151 KB. If you enter a value of `Unlimited`, no limit is imposed on the message size that can be received

			by recipients in the organization.
MaxRecipientEnvelopeLimit	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxRecipientEnvelopeLimit* parameter specifies the maximum number of recipients in a message. The default value is 5000. The valid input range for this parameter is from 0 through 2147483647. If you enter a value of `Unlimited`, no limit is imposed on the number of recipients in a message. Exchange treats an unexpanded distribution group as one recipient.
MaxRetriesForLocalSiteShadow	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *MaxRetriesForLocalSiteShadow* parameter specifies the maximum number of attempts to make a shadow copy of the message in the local Active Directory site. Valid

			input for this parameter is an integer between 0 and 255. The default value is 2. The total number of attempts to create a shadow copy of the message is controlled by the *ShadowMessagePreferenceSetting* parameter: • If *ShadowMessagePreferenceSetting* is set to `LocalOnly`, the total number of attempts to make a shadow copy of the message is the value of the *MaxRetriesForLocalSiteShadow* parameter. • If *ShadowMessagePreferenceSetting* is set to `PreferRemote`, the total number of attempts to make a shadow copy of the message is the value of the *MaxRetriesForLocalSiteShadow* and *MaxRetriesForRemoteSiteShadow* parameters added together.

			• If *ShadowMessagePreferenceSetting* is set to `RemoteOnly`, the value of *MaxRetriesForLocalSiteShadow* is 0, and the *MaxRetriesForLocalSiteShadow* parameter has no effect on the total number of attempts to create a shadow copy of the message. If a shadow copy of the message isn't created after the specified number of attempts, accepting or rejecting the message is controlled by the *RejectMessageOnShadowFailure* parameter.
MaxRetriesForRemoteSiteShadow	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *MaxRetriesForRemoteSiteShadow* parameter specifies the maximum number of attempts to make a shadow copy of the message in a different Active Directory site. Valid

			input for this parameter is an integer between 0 and 255. The default value is 4. The total number of attempts to create a shadow copy of the message is controlled by the *ShadowMessagePreferenceSetting* parameter: • If *ShadowMessagePreferenceSetting* is set to `RemoteOnly`, the total number of attempts to make a shadow copy of the message is the value of the *MaxRetriesForRemoteSiteShadow* parameter. • If *ShadowMessagePreferenceSetting* is set to `PreferRemote`, the total number of attempts to make a shadow copy of the message is the value of the *MaxRetriesForLocalSiteShadow* and *MaxRetriesForRemoteSiteShadow* parameters added together.

			• If *ShadowMessagePreferenceSetting* is set to `LocalOnly`, the value of *MaxRetriesForRemoteSiteShadow* is 0, and the *MaxRetriesForRemoteSiteShadow* parameter has no effect on the total number of attempts to create a shadow copy of the message. If a shadow copy of the message isn't created after the specified number of attempts, accepting or rejecting the message is controlled by the *RejectMessageOnShadowFailure* parameter.
MaxSendSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxSendSize* parameter specifies the maximum message size that can be sent by senders in the organization. The default value is 10 MB. When you enter a value, qualify the value with one

			of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 0 through 2097151 KB. If you enter a value of `Unlimited`, no limit is imposed on the message size that can be sent by senders in the organization.
MigrationEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
OpenDomainRoutingEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
OrganizationFederatedMailbox	Optional	Microsoft.Exchange.Data.SmtpAddress	The *OrganizationFederatedMailbox* parameter specifies the SMTP address of the federated mailbox used for federated delivery with other organizations.
QueueDiagnosticsAggregationInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is available only in on-premises Exchange 2013.

			The *QueueDiagnosticsAggregationInterval* parameter specifies the polling interval that's used to retrieve message queue diagnostic information. The default value is `00:01:00` or one minute. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
RedirectDLMessagesForLegacyArchiveJournaling	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RedirectUnprovisionedUserMessagesForLegacyArchiveJournaling	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RejectMessageOnShadowFailure	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *RejectMessageOnShadowFailure* parameter accepts or rejects messages when a shadow copy of a message can't be created. Valid input for this parameter is `$true` or

			`$false`. The default value is `$true`. When this parameter is set to `$true`, messages are rejected with the SMTP code 450 4.5.1. When this parameter is set to `$false`, the message is accepted without making a shadow copy. The number of attempts to make a shadow copy of the message and where to make the shadow copy are controlled by the *MaxRetriesForLocalSiteShadow*, *MaxRetriesForRemoteSiteShadow*, and *ShadowMessagePreferenceSetting* parameter settings.
Rfc2231EncodingEnabled	Optional	System.Boolean	The *Rfc2231EncodingEnabled* parameter specifies whether the RFC 2231 encoding of MIME parameters for outbound messages is enabled in your organization. Valid input for this parameter is `$true` or `$false`. The

			default value is `$false`.
SafetyNetHoldTime	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is available only in on-premises Exchange 2013. The *SafetyNetHoldTime* parameter specifies how long a copy of a successfully processed message is retained in Safety Net. Unacknowledged shadow copies of messages auto-expire from Safety Net based on adding the values of the *SafetyNetHoldTime* parameter and the *MessageExpirationTimeout* parameter on the **Set-TransportService** cmdlet. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The default value is `2.00:00:00` or 2 days.
ShadowHeartbeatFrequency	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is available only in on-premises Exchange 2013. The

			ShadowHeartbeatFrequency parameter specifies the amount of time a server waits before establishing a connection to a primary server to query the discard status of shadow messages. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Valid input for this parameter is 00:00:01 to 1.00:00:00. The default value is `00:02:00` or 2 minutes.
ShadowHeartbeatRetryCount	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. This parameter isn't used by Microsoft Exchange Server 2013. It's only used by Microsoft Exchange 2010 servers in a coexistence environment. The *ShadowHeartbeatRetryCount* parameter specifies the number of time-outs a server waits before

			deciding that a primary server has failed and assumes ownership of shadow messages in the shadow queue for the primary server that's unreachable. Valid input for this parameter is an integer between 1 and 15. The default value is 12. This parameter is replaced by the *ShadowResubmitTimeSpan* parameter.
ShadowHeartbeatTimeoutInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is available only in on-premises Exchange 2013. This parameter isn't used by Microsoft Exchange Server 2013. It's only used by Microsoft Exchange 2010 servers in a coexistence environment. The *ShadowHeartbeatTimeoutInterval* parameter specifies the amount of time a server waits before establishing a connection to a primary server to query the discard status of shadow messages.

			To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Valid input for this parameter is 00:00:01 to 1.00:00:00. The default value is `00:15:00` or 15 minutes. This parameter is replaced by the *ShadowHeartbeatFrequency* parameter.
ShadowMessageAutoDiscardInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is available only in on-premises Exchange 2013. The *ShadowMessageAutoDiscardInterval* parameter specifies how long a server retains discard events for shadow messages. A primary server queues discard events until queried by the shadow server. However, if the shadow server doesn't query the primary server for the duration specified in this parameter, the primary

			server deletes the queued discard events. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Valid input for this parameter is `00:00:05` to `90.00:00:00`. The default value is `2.00:00:00` or 2 days.
ShadowMessagePreferenceSetting	Optional	Microsoft.Exchange.Data.ShadowMessagePreference	This parameter is available only in on-premises Exchange 2013. The *ShadowMessagePreferenceSetting* parameter specifies the preferred location for making a shadow copy of a message. Valid values are: • `LocalOnly`: A shadow copy of the message should only be made on a server in the local Active Directory site. • `RemoteOnly`: A shadow copy of the message should only be made on a server in a different Active Directory site. • `PreferRemote`: Try to make a shadow copy of the message in a

			different Active Directory site. If the operation fails, try make a shadow copy of the message on a server in the local Active Directory site. The default value is `PreferRemote`.
ShadowRedundancyEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *ShadowRedundancyEnabled* parameter specifies whether shadow redundancy is enabled in the organization. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
ShadowResubmitTimeSpan	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is available only in on-premises Exchange 2013. The *ShadowResubmitTimeSpan* parameter specifies amount of time a server waits before deciding that a primary server has failed and assumes ownership of shadow messages in the shadow

			queue for the primary server that's unreachable. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Valid input for this parameter is 00:00:01 to 1.00:00:00. The default value is 03:00:00 or 3 hours. This parameter replaces the *ShadowHeartbeatRetryCount* parameter.
SupervisionTags	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *SupervisionTags* parameter specifies the various tags that are used for transport supervision in the organization. When you install Exchange, two tags, `Allow` and `Reject`, are created by default.
TLSReceiveDomainSecureList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013.

			The *TLSReceiveDomainSecureList* parameter specifies the domains from which you want to receive domain secured email by using mutual Transport Layer Security (TLS) authentication. To fully support mutual TLS authentication, you must also perform the following steps: • Enable Domain Security (Mutual Auth TLS) and the TLS authentication method on the Receive connectors that receive messages from the domains that you specified with the *TLSReceiveDomainSecureList* parameter. • Specify the domains to which you want to send domain secured email by using the *TLSSendDomainSecureList* parameter. • Enable Domain Security (Mutual Auth TLS) on the Send connectors that send messages to

			the domains that you specified in the *TLSSendDomainSecureList* parameter. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. The wildcard character (*) isn't supported in the domains listed in the *TLSReceiveDomainSecureList* parameter or in the *TLSSendDomainSecureList* parameter. The default value for both parameters is an empty list ({}).

TLSSendDomainSecureList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *TLSSendDomainSecureList* parameter specifies the domains from which you want to send domain secured email by using mutual TLS authentication. To fully support mutual TLS authentication, you must also perform the following steps: • Enable Domain Security (Mutual Auth TLS) on the Send connectors that send messages to the domains that you specified in the *TLSSendDomainSecureList* parameter. • Specify the domains from which you want to receive domain secured email by using the *TLSReceiveDomainSecureList* parameter. • Enable Domain Security (Mutual Auth TLS) and the TLS authentication method on the Receive

			connectors that receive messages from the domains that you specified in the *TLSReceiveDomainSecureList* parameter. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. Multiple domains may be separated by commas. The wildcard character (*) isn't supported in the domains listed in the *TLSSendDomainSecureList* parameter or in the

			TLSReceiveSecureList parameter. The default values for both parameters are an empty list ({}).
TransportRuleAttachmentTextScanLimit	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is available only in on-premises Exchange 2013. The *TransportRuleAttachmentTextScanLimit* parameter specifies the maximum size of text to extract from attachments for scanning by attachment scanning predicates in transport rules and data loss prevention (DLP) policies. The default value is 150 kilobytes (KB). When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. If the amount of text in the attachment is larger than the value of this

			parameter, only the specified amount of text is scanned. For example, if a 5 megabyte attachment contains 300 kilobytes of text, and the value of *TransportRuleAttachmentTextScanLimit* is 150 kilobytes, only the first 150 kilobytes of text are extracted and scanned.
TransportRuleCollectionAddedRecipientsLimit	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
TransportRuleCollectionRegexCharsLimit	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportRuleLimit	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
TransportRuleMinProductVersion	Optional	System.Version	This parameter is reserved for internal Microsoft use.
TransportRuleRegexValidationTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
TransportRuleSizeLimit	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
VerifySecureSubmitEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *VerifySecureSubmitEnable*

			d parameter verifies that email clients submitting messages from mailboxes on Mailbox servers are using encrypted MAPI submission. The valid values for this parameter are `$true` or `$false`. The default value is `$false` If the *VerifySecureSubmitEnabled* parameter is set to `$true`, and Outlook 2010 or later is used to submit the message, the message is marked as secure. If a previous version of Outlook is used to submit the message, the message is marked as anonymous. If the *VerifySecureSubmitEnabled* parameter is set to `$false`, all MAPI message submissions are marked as secure. Messages submitted from mailboxes on the Mailbox server by using any MAPI client aren't checked for encrypted MAPI submission. If you use

			previous Outlook versions in your Exchange organization, you should set the *VerifySecureSubmitEnabled* parameter to `$false`.
VoicemailJournalingEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *VoicemailJournalingEnabled* parameter specifies whether Unified Messaging voice mail messages are journaled by the Journaling agent. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Xexch50Enabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *Xexch50Enabled* parameter specifies whether Xexch50 authentication should be enabled for backward compatibility with computers running Exchange 2003. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-TransportPipeline

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-TransportPipeline** cmdlet to view each transport agent and the event with which the transport agent is registered.

```
Get-TransportPipeline [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns a list of agents registered in the transport pipeline.

```
Get-TransportPipeline
```

EXAMPLE 2

This example returns a list of agents registered in the transport pipeline with full details for each transport event.

```
Get-TransportPipeline | Format-List
```

Detailed Description

The **Get-TransportPipeline** cmdlet enables you to view all the transport agents configured in the following locations:

- In the Transport service on a Mailbox server.
- In the Front End Transport service on a Client Access server.
- On an Edge Transport server in the perimeter network.

Note:

The associated transport service must be started, and at least one e-mail message must be sent through the server since the last service restart before the transport pipeline can be viewed. Only the transport events and agents that were involved in the processing of e-mail messages since the associated service was last restarted are returned.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport agents" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-TransportServer

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-TransportServer** cmdlet to view the transport configuration information for the Transport service on Mailbox servers or for Edge Transport servers.

```
Get-TransportServer [-Identity <TransportServerIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example provides different results depending on the server role on which it's run. When you run this command on an Edge Transport server, it provides a configuration summary for the local server. Otherwise, it displays a list of all Mailbox servers in your organization.

```
Get-TransportServer
```

EXAMPLE 2

This example retrieves the detailed transport configuration information for the Transport service on the Mailbox server named Mailbox01.

```
Get-TransportServer Mailbox01 | Format-List
```

Detailed Description

The **Get-TransportServer** cmdlet will be removed in a future version of Exchange. You should use the **Get-TransportService** cmdlet instead.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport service" or "Edge Transport server" entries in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't

			supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.TransportServerIdParameter	The *Identity* parameter specifies the server you want to view. When you use this parameter on a Mailbox server, the parameter returns the transport configuration of the Transport service on specified server. You can't use this parameter on an Edge Transport server.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-TransportServer

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-18*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-TransportServer** cmdlet to set the transport configuration options for the Transport service on Mailbox servers or for Edge Transport servers.

```
Set-TransportServer -Identity <ServerIdParameter> [-
ActiveUserStatisticsLogMaxAge <EnhancedTimeSpan>] [-
ActiveUserStatisticsLogMaxDirectorySize <ByteQuantifiedSize>] [-
ActiveUserStatisticsLogMaxFileSize <ByteQuantifiedSize>] [-
ActiveUserStatisticsLogPath <LocalLongFullPath>] [-AgentLogEnabled <$true
| $false>] [-AgentLogMaxAge <EnhancedTimeSpan>] [-AgentLogMaxDirectorySize
<Unlimited>] [-AgentLogMaxFileSize <Unlimited>] [-AgentLogPath
<LocalLongFullPath>] [-AntispamAgentsEnabled <$true | $false>] [-Confirm
[<SwitchParameter>]] [-ConnectivityLogEnabled <$true | $false>] [-
ConnectivityLogMaxAge <EnhancedTimeSpan>] [-
ConnectivityLogMaxDirectorySize <Unlimited>] [-ConnectivityLogMaxFileSize
<Unlimited>] [-ConnectivityLogPath <LocalLongFullPath>] [-
ContentConversionTracingEnabled <$true | $false>] [-
DelayNotificationTimeout <EnhancedTimeSpan>] [-
DeltaSyncClientCertificateThumbprint <String>] [-DnsLogEnabled <$true |
$false>] [-DnsLogMaxAge <EnhancedTimeSpan>] [-DnsLogMaxDirectorySize
<Unlimited>] [-DnsLogMaxFileSize <Unlimited>] [-DnsLogPath
<LocalLongFullPath>] [-DomainController <Fqdn>] [-
ExternalDNSAdapterEnabled <$true | $false>] [-ExternalDNSAdapterGuid
<Guid>] [-ExternalDNSProtocolOption <Any | UseUdpOnly | UseTcpOnly>] [-
ExternalDNSServers <MultiValuedProperty>] [-ExternalIPAddress <IPAddress>]
[-FlowControlLogEnabled <$true | $false>] [-FlowControlLogMaxAge
<EnhancedTimeSpan>] [-FlowControlLogMaxDirectorySize <Unlimited>] [-
FlowControlLogMaxFileSize <Unlimited>] [-FlowControlLogPath
<LocalLongFullPath>] [-HttpProtocolLogEnabled <$true | $false>] [-
HttpProtocolLogFilePath <LocalLongFullPath>] [-HttpProtocolLogLoggingLevel
<None | Verbose>] [-HttpProtocolLogMaxAge <EnhancedTimeSpan>] [-
HttpProtocolLogMaxDirectorySize <ByteQuantifiedSize>] [-
HttpProtocolLogMaxFileSize <ByteQuantifiedSize>] [-
HttpTransportSyncProxyServer <String>] [-InternalDNSAdapterEnabled <$true
| $false>] [-InternalDNSAdapterGuid <Guid>] [-InternalDNSProtocolOption
<Any | UseUdpOnly | UseTcpOnly>] [-InternalDNSServers
<MultiValuedProperty>] [-IntraOrgConnectorProtocolLoggingLevel <None |
Verbose>] [-IntraOrgConnectorSmtpMaxMessagesPerConnection <Int32>] [-
IrmLogEnabled <$true | $false>] [-IrmLogMaxAge <EnhancedTimeSpan>] [-
IrmLogMaxDirectorySize <Unlimited>] [-IrmLogMaxFileSize
<ByteQuantifiedSize>] [-IrmLogPath <LocalLongFullPath>] [-
JournalLogEnabled <$true | $false>] [-JournalLogMaxAge <EnhancedTimeSpan>]
[-JournalLogMaxDirectorySize <Unlimited>] [-JournalLogMaxFileSize
<Unlimited>] [-JournalLogPath <LocalLongFullPath>] [-
MaxActiveTransportSyncJobsPerProcessor <Int32>] [-
MaxConcurrentMailboxDeliveries <Int32>] [-MaxConcurrentMailboxSubmissions
<Int32>] [-MaxConnectionRatePerMinute <Int32>] [-
MaxNumberOfTransportSyncAttempts <Int32>] [-MaxOutboundConnections
<Unlimited>] [-MaxPerDomainOutboundConnections <Unlimited>] [-
MessageExpirationTimeout <EnhancedTimeSpan>] [-MessageRetryInterval
<EnhancedTimeSpan>] [-MessageTrackingLogEnabled <$true | $false>] [-
MessageTrackingLogMaxAge <EnhancedTimeSpan>] [-
MessageTrackingLogMaxDirectorySize <Unlimited>] [-
MessageTrackingLogMaxFileSize <ByteQuantifiedSize>] [-
MessageTrackingLogPath <LocalLongFullPath>] [-
MessageTrackingLogSubjectLoggingEnabled <$true | $false>] [-
OutboundConnectionFailureRetryInterval <EnhancedTimeSpan>] [-
PickupDirectoryMaxHeaderSize <ByteQuantifiedSize>] [-
PickupDirectoryMaxMessagesPerMinute <Int32>] [-
PickupDirectoryMaxRecipientsPerMessage <Int32>] [-PickupDirectoryPath
<LocalLongFullPath>] [-PipelineTracingEnabled <$true | $false>] [-
PipelineTracingPath <LocalLongFullPath>] [-PipelineTracingSenderAddress
<SmtpAddress>] [-PoisonMessageDetectionEnabled <$true | $false>] [-
PoisonThreshold <Int32>] [-ProcessingSchedulerLogEnabled <$true | $false>]
[-ProcessingSchedulerLogMaxAge <EnhancedTimeSpan>] [-
```

```
ProcessingSchedulerLogMaxDirectorySize <Unlimited>] [-
ProcessingSchedulerLogMaxFileSize <Unlimited>] [-
ProcessingSchedulerLogPath <LocalLongFullPath>] [-QueueLogMaxAge
<EnhancedTimeSpan>] [-QueueLogMaxDirectorySize <Unlimited>] [-
QueueLogMaxFileSize <Unlimited>] [-QueueLogPath <LocalLongFullPath>] [-
QueueMaxIdleTime <EnhancedTimeSpan>] [-ReceiveProtocolLogMaxAge
<EnhancedTimeSpan>] [-ReceiveProtocolLogMaxDirectorySize <Unlimited>] [-
ReceiveProtocolLogMaxFileSize <Unlimited>] [-ReceiveProtocolLogPath
<LocalLongFullPath>] [-RecipientValidationCacheEnabled <$true | $false>]
[-ReplayDirectoryPath <LocalLongFullPath>] [-ResourceLogEnabled <$true |
$false>] [-ResourceLogMaxAge <EnhancedTimeSpan>] [-
ResourceLogMaxDirectorySize <Unlimited>] [-ResourceLogMaxFileSize
<Unlimited>] [-ResourceLogPath <LocalLongFullPath>] [-
RootDropDirectoryPath <String>] [-RoutingTableLogMaxAge
<EnhancedTimeSpan>] [-RoutingTableLogMaxDirectorySize <Unlimited>] [-
RoutingTableLogPath <LocalLongFullPath>] [-SendProtocolLogMaxAge
<EnhancedTimeSpan>] [-SendProtocolLogMaxDirectorySize <Unlimited>] [-
SendProtocolLogMaxFileSize <Unlimited>] [-SendProtocolLogPath
<LocalLongFullPath>] [-ServerStatisticsLogMaxAge <EnhancedTimeSpan>] [-
ServerStatisticsLogMaxDirectorySize <ByteQuantifiedSize>] [-
ServerStatisticsLogMaxFileSize <ByteQuantifiedSize>] [-
ServerStatisticsLogPath <LocalLongFullPath>] [-SmtpEnableAllTlsVersions
<$true | $false>] [-TransientFailureRetryCount <Int32>] [-
TransientFailureRetryInterval <EnhancedTimeSpan>] [-
TransportMaintenanceLogEnabled <$true | $false>] [-
TransportMaintenanceLogMaxAge <EnhancedTimeSpan>] [-
TransportMaintenanceLogMaxDirectorySize <Unlimited>] [-
TransportMaintenanceLogMaxFileSize <Unlimited>] [-
TransportMaintenanceLogPath <LocalLongFullPath>] [-
TransportSyncAccountsPoisonAccountThreshold <Int32>] [-
TransportSyncAccountsPoisonDetectionEnabled <$true | $false>] [-
TransportSyncAccountsPoisonItemThreshold <Int32>] [-
TransportSyncAccountsSuccessivePoisonItemThreshold <Int32>] [-
TransportSyncEnabled <$true | $false>] [-TransportSyncExchangeEnabled
<$true | $false>] [-TransportSyncFacebookEnabled <$true | $false>] [-
TransportSyncHubHealthLogEnabled <$true | $false>] [-
TransportSyncHubHealthLogFilePath <LocalLongFullPath>] [-
TransportSyncHubHealthLogMaxAge <EnhancedTimeSpan>] [-
TransportSyncHubHealthLogMaxDirectorySize <ByteQuantifiedSize>] [-
TransportSyncHubHealthLogMaxFileSize <ByteQuantifiedSize>] [-
TransportSyncImapEnabled <$true | $false>] [-TransportSyncLinkedInEnabled
<$true | $false>] [-TransportSyncLogEnabled <$true | $false>] [-
TransportSyncLogFilePath <LocalLongFullPath>] [-
TransportSyncLogLoggingLevel <None | Error | Information | Verbose |
RawData | Debugging>] [-TransportSyncLogMaxAge <EnhancedTimeSpan>] [-
TransportSyncLogMaxDirectorySize <ByteQuantifiedSize>] [-
TransportSyncLogMaxFileSize <ByteQuantifiedSize>] [-
TransportSyncMaxDownloadItemsPerConnection <Int32>] [-
TransportSyncMaxDownloadSizePerConnection <ByteQuantifiedSize>] [-
TransportSyncMaxDownloadSizePerItem <ByteQuantifiedSize>] [-
TransportSyncPopEnabled <$true | $false>] [-
TransportSyncRemoteConnectionTimeout <EnhancedTimeSpan>] [-
UseDowngradedExchangeServerAuth <$true | $false>] [-WhatIf
[<SwitchParameter>]] [-WindowsLiveHotmailTransportSyncEnabled <$true |
$false>] [-WlmLogMaxAge <EnhancedTimeSpan>] [-WlmLogMaxDirectorySize
<Unlimited>] [-WlmLogMaxFileSize <Unlimited>] [-WlmLogPath
<LocalLongFullPath>]
```

Examples

EXAMPLE 1

This example sets the *DelayNotificationTimeout* parameter to 13 hours on server named Mailbox01.

```
Set-TransportServer Mailbox01 -DelayNotificationTimeout
13:00:00
```

EXAMPLE 2

This example sets the *TransientFailureRetryCount* parameter to 3 and sets the *TransientFailureRetryInterval* parameter to 30 seconds on server named Mailbox01.

```
Set-TransportServer Mailbox01 -TransientFailureRetryCount 3
-TransientFailureRetryInterval 00:00:30
```

EXAMPLE 3

This example sets the *ReceiveProtocolLogPath* parameter to C:\SMTP Protocol Logs\Receive.log on server Mailbox01.

```
Set-TransportServer Mailbox01 -ReceiveProtocolLogPath "C:
\SMTP Protocol Logs\Receive.log"
```

Detailed Description

The **Set-TransportServer** cmdlet will be removed in a future version of Exchange. You should use the **Set-TransportService** cmdlet instead.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport service" and "Edge Transport server" entries in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the server that you want to modify.
ActiveUserStatisticsLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ActiveUserStatisticsLogMaxAge* parameter specifies the maximum duration that the per user activity statistics log files are kept. Log files older than the specified value are deleted. The default value

			is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 10 days for this parameter, use 10.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of this parameter to 00:00:00 prevents the automatic removal of server statistics log files.
ActiveUserStatisticsLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ActiveUserStatisticsLogMaxDirectorySize* parameter specifies the cap on the size of the per user activity statistics log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The minimum value is 1 megabyte (MB). The default value is 250 MB. When you enter a value,

			qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ActiveUserStatisticsLogMaxFileSize* parameter must be less than or equal to the value of the *ActiveUserStatisticsLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the server statistics log directory.
ActiveUserStatisticsLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ActiveUserStatisticsLogMaxFileSize* parameter specifies the maximum file size for the per user activity statistics log files. When a log file reaches its maximum file size, a new log file is created. The

			default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ActiveUserStatisticsLogMaxFileSize* parameter must be less than or equal to the value of the *ActiveUserStatisticsLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the server statistics log files.
ActiveUserStatisticsLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ActiveUserStatisticsLogPath* parameter specifies the location of per user activity statistics log storage. The default location is %

			`ExchangeInstallPath%TransportRoles\Logs\Hub\ActiveUsersStats`. Setting the value of this parameter to `$null` disables server statistics logging.
AgentLogEnabled	Optional	System.Boolean	The *AgentLogEnabled* parameter specifies whether the agent log is enabled. The default value is `$true`.
AgentLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *AgentLogMaxAge* parameter specifies the maximum age for the agent log file. Log files older than the specified value are deleted. The default value is `7.00:00:00` or 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Setting the value of the *AgentLogMaxAge* parameter to 00:00:00 prevents the automatic removal of agent log files because of their age.
AgentLogMaxDirector	Optional	Microsoft.Exchange.Da	The

ySize		ta.Unlimited	*AgentLogMaxDirectorySize* parameter specifies the maximum size of all agent logs in the agent log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *AgentLogMaxFileSize* parameter must be less than or equal to the value of the *AgentLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the agent log directory.
AgentLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *AgentLogMaxFileSize* parameter specifies the maximum size of each

			agent log file. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *AgentLogMaxFileSize* parameter must be less than or equal to the value of the *AgentLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the agent log files.
AgentLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *AgentLogPath* parameter specifies the default agent log directory location. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\AgentLog`. Setting the value of this

			parameter to `$null` disables agent logging. However, setting this parameter to `$null` when the value of the *AgentLogEnabled* attribute is `$true` generates event log errors.
AntispamAgentsEnabled	Optional	System.Boolean	The *AntispamAgentsEnabled* parameter specifies whether anti-spam agents are installed on the server specified with the *Identity* parameter. The default value is `$false` for the Transport service on Mailbox servers and `$true` on Edge servers. **Important:** You set this parameter by using a script. You shouldn't modify this parameter manually.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a

			value with the *Confirm* switch.
ConnectivityLogEnabled	Optional	System.Boolean	The *ConnectivityLogEnabled* parameter specifies whether the connectivity log is enabled. The default value is `$true`.
ConnectivityLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConnectivityLogMaxAge* parameter specifies the maximum age for the connectivity log file. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 25 days for this parameter, use 25.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *ConnectivityLogMaxAge*

			parameter to 00:00:00 prevents the automatic removal of connectivity log files because of their age.
ConnectivityLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *ConnectivityLogMaxDirectorySize* parameter specifies the maximum size of all connectivity logs in the connectivity log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 1000 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ConnectivityLogMaxFileSize* parameter must be less than or equal to the value of the *ConnectivityLogMaxDirect*

			orySize parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the connectivity log directory.
ConnectivityLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ConnectivityLogMaxFileSize* parameter specifies the maximum size of each connectivity log file. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *ConnectivityLogMaxFileSize* parameter must be less than or equal to the value of the

			ConnectivityLogMaxDirectorySize parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the connectivity log files.
ConnectivityLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ConnectivityLogPath* parameter specifies the default connectivity log directory location. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\Connectivity`. Setting the value of this parameter to `$null` disables connectivity logging. However, setting this parameter to `$null` when the value of the *ConnectivityLogEnabled* attribute is `$true` generates event log errors.
ContentConversionTracingEnabled	Optional	System.Boolean	The *ContentConversionTracingEnabled* parameter specifies whether content conversion tracing is

			enabled. Content conversion tracing captures content conversion failures that occur in the Transport service on a Mailbox server or on the Edge server. The default value is `$false`. Content conversion tracing captures a maximum of 128 MB of content conversion failures. When the 128 MB limit is reached, no more content conversion failures are captured. Content conversion tracing captures the complete contents of e-mail messages to the path specified by the *PipelineTracingPath* parameter. Make sure that you restrict access to this directory. The permissions required on the directory specified by the *PipelineTracingPath* parameter are as follows: • Administrators: Full Control • Network Service: Full

			Control • System: Full Control
DelayNotificationTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *DelayNotificationTimeout* parameter specifies how long the server waits before it generates a delayed delivery status notification (DSN) message. The default value is 4 hours. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 3.5 hours for this parameter, use 03:30:00. The valid input range for this parameter is from 00:00:01 through 30.00:00:00. The value of the *DelayNotificationTimeout* parameter should always be greater than the value of the *TransientFailureRetryCount* parameter multiplied by the value of the *TransientFailureRetryInter*

			val parameter.
DeltaSyncClientCertificateThumbprint	Optional	System.String	This parameter is reserved for internal Microsoft use.
DnsLogEnabled	Optional	System.Boolean	The *DnsLogEnabled* parameter specifies whether the DNS log is enabled. The default value is `$false`.
DnsLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *DnsLogMaxAge* parameter specifies the maximum age for the DNS log file. Log files older than the specified value are deleted. The default value is `7.00:00:00` or 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Setting the value of the *DnsLogMaxAge* parameter to `00:00:00` prevents the automatic removal of DNS log files because of their age.
DnsLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *DnsLogMaxDirectorySize* parameter specifies the maximum size of all DNS

			logs in the DNS log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 100 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *DnsLogMaxFileSize* parameter must be less than or equal to the value of the *DnsLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the DNS log directory.
DnsLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *DnsLogMaxFileSize* parameter specifies the maximum size of each DNS log file. When a log file reaches its maximum file size, a new log file is

			created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *DnsLogMaxFileSize* parameter must be less than or equal to the value of the *DnsLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the DNS log files.
DnsLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *DnsLogPath* parameter specifies the DNS log directory location. The default value is blank (`$null`), which indicates no location is configured. If you enable DNS logging, you need to specify a local file path for the DNS log files by using this parameter. If the path

			contains spaces, enclose the entire path value in quotation marks (").
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
ExternalDNSAdapterEnabled	Optional	System.Boolean	The *ExternalDNSAdapterEnabled* parameter specifies one or more Domain Name System (DNS) servers that Exchange uses for external DNS lookups. When the *ExternalDNSAdapterEnabled* parameter is set to `$true`, DNS lookups of

			destinations outside the Exchange organization are performed by using the DNS settings of the external network adapter specified by the value of the *ExternalDNSAdapterGuid* parameter. If you want to specify a custom list of DNS servers used for external Exchange DNS lookups only, you must specify the DNS servers by using the *ExternalDNSServers* parameter, and you must also set the value of the *ExternalDNSAdapterEnabled* parameter to `$false`. The default value of the *ExternalDNSAdapterEnabled* parameter is `$true`.
ExternalDNSAdapterGuid	Optional	System.Guid	The *ExternalDNSAdapterGuid* parameter specifies the network adapter that has the DNS settings used for DNS lookups of destinations that exist outside the Exchange organization. The concept of an external network

			adapter and an internal network adapter is only applicable in a multi-homed Exchange server environment. When no particular network adapter is specified as the network adapter for external DNS lookups, the value of the *ExternalDNSAdapterGuid* parameter is `00000000-0000-0000-0000-000000000000`, and external DNS lookups are performed by using the DNS settings of any available network adapter. You may enter the GUID of a specific network adapter to use for external DNS lookups. The default value of the *ExternalDNSAdapterGuid* parameter is `00000000-0000-0000-0000-000000000000`. **Note:** If the value of the *ExternalDNSAdapterEnabled* parameter is set to `$false`, the value of the *ExternalDNSAdapterGuid* parameter is ignored, and the list of DNS servers from the *ExternalDNSServers*

			parameter is used.
ExternalDNSProtocolOption	Optional	Microsoft.Exchange.Data.ProtocolOption	The *ExternalDNSProtocolOption* parameter specifies which protocol to use when querying external DNS servers. The valid options for this parameter are `Any`, `UseTcpOnly`, and `UseUdpOnly`. The default value is `Any`.
ExternalDNSServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalDNSServers* parameter specifies the list of external DNS servers that the server queries when resolving a remote domain. DNS servers are specified by IP address. The default value is an empty list ({}). To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...`

			To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. **Note:** If the value of the *ExternalDNSAdapterEnabled* parameter is set to `$true`, the *ExternalDNSServers* parameter and its list of DNS servers isn't used.
ExternalIPAddress	Optional	System.Net.IPAddress	The *ExternalIPAddress* parameter specifies the IP address used in the `Received` message header field for every message that travels through the Edge server or the Transport service on a Mailbox server. The IP address in the `Received` header field is used for hop count and routing loop detection. The IP address specified by the *ExternalIPAddress* parameter overrides the external network adapter's actual IP address. Typically, you

			would want to set the value of the *ExternalIPAddress* parameter to match the value of your domain's public MX record. The default value of the *ExternalIPAddress* parameter is blank. This means the actual IP address of the external network adapter is used in the `Received` header field.
FlowControlLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
FlowControlLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
FlowControlLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
FlowControlLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
FlowControlLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
HttpProtocolLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
HttpProtocolLogFilePath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
HttpProtocolLogLoggingLevel	Optional	Microsoft.Exchange.Data.ProtocolLoggingLev	This parameter is reserved for internal Microsoft use.

		el	
HttpProtocolLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
HttpProtocolLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
HttpProtocolLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
HttpTransportSyncProxyServer	Optional	System.String	This parameter is reserved for internal Microsoft use.
InternalDNSAdapterEnabled	Optional	System.Boolean	The *InternalDNSAdapterEnabled* parameter specifies one or more DNS servers that Exchange uses for internal DNS lookups. When the *InternalDNSAdapterEnabled* parameter is set to `$true`, DNS lookups of destinations inside the Exchange organization are performed by using the DNS settings of the internal network adapter specified by the value of the *InternalDNSAdapterGuid* parameter. If you want to specify a custom list of DNS servers used for

			internal Exchange DNS lookups only, you must specify the DNS servers by using the *InternalDNSServers* parameter, and you must also set the value of the *InternalDNSAdapterEnabled* parameter to `$false`. The default value of the *InternalDNSAdapterEnabled* parameter is `$true`.
InternalDNSAdapterGuid	Optional	System.Guid	The *InternalDNSAdapterGuid* parameter specifies the network adapter that has the DNS settings used for DNS lookups of servers that exist inside the Exchange organization. The concept of an internal network adapter and an external network adapter is only applicable in a multi-homed Exchange server environment. When no particular network adapter is specified as the network adapter for external DNS lookups, the value of the *InternalDNSAdapterGuid* parameter is `00000000-`

			0000-0000-0000-000000000000, and internal DNS lookups are performed by using the DNS settings of any available network adapter. You may enter the GUID of a specific network adapter to use for internal DNS lookups. The default value of the *InternalDNSAdapterGuid* parameter is 00000000-0000-0000-0000-000000000000. **Note:** If the value of the *InternalDNSAdapterEnabled* parameter is set to $false, the value of the *InternalDNSAdapterGuid* parameter is ignored, and the list of DNS servers from the *InternalDNSServers* parameter is used.
InternalDNSProtocolOption	Optional	Microsoft.Exchange.Data.ProtocolOption	The *InternalDNSProtocolOption* parameter specifies which protocol to use when you query internal DNS servers. Valid options for this parameter are Any, UseTcpOnly, or UseUdpOnly. The default value is Any.

InternalDNSServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *InternalDNSServers* parameter specifies the list of DNS servers that should be used when resolving a domain name. DNS servers are specified by IP address. The default value is any empty list ({}). To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. **Note:** If the *InternalDNSAdapterGuid* parameter is set, and the value of the *InternalDNSAdapterEnabled* parameter is set to

			`$true`, the *InternalDNSServers* parameter and its list of DNS servers isn't used.
IntraOrgConnectorProtocolLoggingLevel	Optional	Microsoft.Exchange.Data.ProtocolLoggingLevel	The *IntraOrgConnectorProtocolLoggingLevel* parameter enables or disables SMTP protocol logging on the implicit and invisible intra-organization Send connectors that are used to transmit messages between Exchange servers in the Exchange organization. Valid values for this parameter are `None` or `Verbose`. The value `Verbose` enables protocol logging for the connector. The value `None` disables protocol logging for the connector. The default value is `None`. When the *IntraOrgConnectorProtocolLoggingLevel* parameter is set to `Verbose`, the information is written to the Send connector protocol log specified by the *SendProtocolLog* parameters.

IntraOrgConnectorSmtpMaxMessagesPerConnection	Optional	System.Int32	The *IntraOrgConnectorSmtpMaxMessagesPerConnection* parameter specifies the maximum number of messages per connection limit on the implicit and invisible intra-organization Send connectors that are used to transmit messages between Exchange servers in the Exchange organization.
IrmLogEnabled	Optional	System.Boolean	The *IrmLogEnabled* parameter enables logging of Information Rights Management (IRM) transactions. IRM logging is enabled by default. Values include: • `$true` Enable IRM logging • `$false` Disable IRM logging
IrmLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *IrmLogMaxAge* parameter specifies the maximum age for the IRM log file. Log files that are older than the specified value are deleted. The default value is 30 days. To specify a value, enter it

			as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The valid input range for this parameter is 00:00:00 to 24855.03:14:07. Setting the value of the *IrmLogMaxAge* parameter to 00:00:00 prevents the automatic removal of connectivity log files because of their age.
IrmLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This *IrmLogMaxDirectorySize* parameter specifies the maximum size of all IRM logs in the connectivity log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are

			treated as bytes. The value of the *IrmLogMaxFileSize* parameter must be less than or equal to the value of the *IrmLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the IRM log directory.
IrmLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This *IrmLogMaxFileSize* parameter specifies the maximum size of each IRM log file. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are

			treated as bytes. The value of the *IrmLogMaxFileSize* parameter must be less than or equal to the value of the *IrmLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the IRM log files.
IrmLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *IrmLogPath* parameter specifies the default IRM log directory location. The default location is `%ExchangeInstallPath%Logging\IRMLogs`. If you set the value of the *IrmLogPath* parameter to `$null`, you effectively disable IRM logging. However, if you set the value of the *IrmLogPath* parameter to `$null` when the value of the *IrmLogEnabled* attribute is `$true`, Exchange will log

			errors in the Application event log. The preferred way for disabling IRM logging is to set the *IrmLogEnabled* to `$false`.
JournalLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
JournalLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
JournalLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
JournalLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
JournalLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
MaxActiveTransportSyncJobsPerProcessor	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
MaxConcurrentMailboxDeliveries	Optional	System.Int32	The *MaxConcurrentMailboxDeliveries* parameter specifies the maximum number of delivery threads that the transport service can have open at the same time to deliver messages to mailboxes. The default value is 20. The valid input range for this parameter is from 1

			through 256. We recommend that you don't modify the default value unless Microsoft Customer Service and Support advises you to do this.
MaxConcurrentMailboxSubmissions	Optional	System.Int32	The *MaxConcurrentMailboxSubmissions* parameter specifies the maximum number of submission threads that the transport service can have open at the same time to send messages from mailboxes. The default value is 20. The valid input range for this parameter is from 1 through 256.
MaxConnectionRatePerMinute	Optional	System.Int32	The *MaxConnectionRatePerMinute* parameter specifies the maximum rate that connections are allowed to be opened with the transport service. If many connections are attempted with the transport service at the same time, the *MaxConnectionRatePerMi*

			nute parameter limits the rate that the connections are opened so that the server's resources aren't overwhelmed. The default value is 1200 connections per minute. The valid input range for this parameter is from 1 through 2147483647.
MaxNumberOfTransportSyncAttempts	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
MaxOutboundConnections	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxOutboundConnections* parameter specifies the maximum number of outbound connections that can be open at a time. The default value is 1000. The valid input range for this parameter is from 1 through 2147483647. If you enter a value of `unlimited`, no limit is imposed on the number of outbound connections. The value of the *MaxOutboundConnections* parameter must be greater than or equal to the value of the *MaxPerDomainOutbound*

			Connections parameter.
MaxPerDomainOutboundConnections	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxPerDomainOutboundConnections* parameter specifies the maximum number of concurrent connections to any single domain. The default value is 20. The valid input range for this parameter is from 1 through 2147483647. If you enter a value of `unlimited`, no limit is imposed on the number of outbound connections per domain. The value of the *MaxPerDomainOutboundConnections* parameter must be less than or equal to the value of the *MaxOutboundConnections* parameter.
MessageExpirationTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MessageExpirationTimeout* parameter specifies the maximum time that a particular message can remain in a queue. If a message remains in the queue for longer than this period of time, the

			message is returned to the sender as a permanent failure. The default value is 2 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 3 days for this parameter, use 3.00:00:00. The valid input range for this parameter is from 00:00:05 through 90.00:00:00.
MessageRetryInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MessageRetryInterval* parameter specifies the retry interval for individual messages after a connection failure with a remote server. The default value is 1 minute. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 2 minutes for this parameter, use 00:02:00.

			The valid input range for this parameter is from 00:00:01 through 1.00:00:00. We recommend that you don't modify the default value unless Customer Service and Support advises you to do this.
MessageTrackingLogEnabled	Optional	System.Boolean	The *MessageTrackingLogEnabled* parameter specifies whether message tracking is enabled. The default value is `$true`.
MessageTrackingLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MessageTrackingLogMaxAge* parameter specifies the message tracking log maximum file age. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 20 days for this parameter,

			use 20.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *MessageTrackingLogMaxAge* parameter to 00:00:00 prevents the automatic removal of message tracking log files because of their age.
MessageTrackingLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *MessageTrackingLogMaxDirectorySize* parameter specifies the maximum size of the message tracking log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The maximum size of the message tracking log directory is calculated as the total size of all log files that have the same name prefix. Other files that don't follow the name prefix convention aren't counted in the total directory size calculation.

			Renaming old log files or copying other files into the message tracking log directory could cause the directory to exceed its specified maximum size. For Mailbox servers, the maximum size of the message tracking log directory isn't the specified maximum size because the message tracking log files generated by the Transport service and the Mailbox Transport service have different name prefixes. Message tracking log files for the Transport service or for Edge servers begin with the name prefix *MSGTRK*. Message tracking log files for the Mailbox Transport service begin with the name prefix *MSGTRKM*. For Mailbox servers, the maximum size of the message tracking log directory is two times the specified value. The default value is 1000 MB.

			When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *MessageTrackingLogMaxFileSize* parameter must be less than or equal to the value of the *MessageTrackingLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the message tracking log directory.
MessageTrackingLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *MessageTrackingLogMaxFileSize* parameter specifies the maximum size of the message tracking log files. When a log file reaches its maximum file size, a new

			log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *MessageTrackingLogMaxFileSize* parameter must be less than or equal to the value of the *MessageTrackingLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 4294967296 bytes (4 GB). If you enter a value of `unlimited`, no size limit is imposed on the message tracking log files.
MessageTrackingLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *MessageTrackingLogPath* parameter specifies the location of the message tracking logs. The default location is %

			`ExchangeInstallPath%TransportRoles\Logs\MessageTracking`. Setting the value of this parameter to `$null` disables message tracking. However, setting this parameter to `$null` when the value of the *MessageTrackingLogEnabled* attribute is `$true` generates event log errors. The preferred method to disable message tracking is to use the *MessageTrackingLogEnabled* parameter.
MessageTrackingLogSubjectLoggingEnabled	Optional	System.Boolean	The *MessageTrackingLogSubjectLoggingEnabled* parameter specifies whether the message subject should be included in the message tracking log. The default value is `$true`.
OutboundConnectionFailureRetryInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *OutboundConnectionFailureRetryInterval* parameter specifies the retry interval for subsequent connection

			attempts to a remote server where previous connection attempts have failed. The previously failed connection attempts are controlled by the *TransientFailureRetryCount* and *TransientFailureRetryInterval* parameters. For the Transport service on a Mailbox server, the default value of the *OutboundConnectionFailureRetryInterval* parameter is 10 minutes. On an Edge server, the default value is 30 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 15 minutes for this parameter, use 00:15:00. The valid input range for this parameter is from 00:00:01 through 20.00:00:00.
PickupDirectoryMaxH	Optional	Microsoft.Exchange.Da	The

eaderSize		ta.ByteQuantifiedSize	*PickupDirectoryMaxHeaderSize* parameter specifies the maximum message header size that can be submitted to the Pickup directory. The default value is 64 KB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 32768 through 2147483647 bytes.
PickupDirectoryMaxMessagesPerMinute	Optional	System.Int32	The *PickupDirectoryMaxMessagesPerMinute* parameter specifies the maximum number of messages processed per minute by the Pickup directory and by the Replay directory. Each directory can independently process message files at the rate specified by the

			PickupDirectoryMaxMessagesPerMinute parameter. The default value is 100. The valid input range for this parameter is from 1 through 20000.
PickupDirectoryMaxRecipientsPerMessage	Optional	System.Int32	The *PickupDirectoryMaxRecipientsPerMessage* parameter specifies the maximum number of recipients that can be included on an e-mail message. The default value is 100. The valid input range for this parameter is from 1 through 10000.
PickupDirectoryPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *PickupDirectoryPath* parameter specifies the location of the Pickup directory. The Pickup directory is used by administrators and third-party applications to create and submit messages. The default location is `%ExchangeInstallPath%TransportRoles\Pickup`. If the value of the *PickupDirectoryPath*

			parameter is set to `$null`, the Pickup directory is disabled.
PipelineTracingEnabled	Optional	System.Boolean	The *PipelineTracingEnabled* parameter specifies whether to enable pipeline tracing. Pipeline tracing captures message snapshot files that record the changes made to the message by each transport agent configured in the transport service on the server. Pipeline tracing creates verbose log files that accumulate quickly. Pipeline tracing should only be enabled for a short time to provide in-depth diagnostic information that enables you to troubleshoot problems. In addition to troubleshooting, you can use pipeline tracing to validate changes that you make to the configuration of the transport service where you enable pipeline tracing. The default value

			is `$false`.
PipelineTracingPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *PipelineTracingPath* parameter specifies the location of the pipeline tracing logs. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\PipelineTracing`. The path must be local to the Exchange computer. Setting the value of this parameter to `$null` disables pipeline tracing. However, setting this parameter to `$null` when the value of the *PipelineTracingEnabled* attribute is `$true` generates event log errors. The preferred method to disable pipeline tracing is to use the *PipelineTracingEnabled* parameter. Pipeline tracing captures the complete contents of e-mail messages to the path specified by the *PipelineTracingPath* parameter. Make sure that you restrict access to this directory. The permissions

			required on the directory specified by the *PipelineTracingPath* parameter are as follows: • Administrators: Full Control • Network Service: Full Control • System: Full Control
PipelineTracingSenderAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PipelineTracingSenderAddress* parameter specifies the sender e-mail address that invokes pipeline tracing. Only messages from this address generate pipeline tracing output. The address can be either inside or outside the Exchange organization. Depending on your requirements, you may have to set this parameter to different sender addresses and send new messages to start the transport agents or routes that you want to test. The default value of this parameter is `$null`.
PoisonMessageDetectionEnabled	Optional	System.Boolean	The *PoisonMessageDetectionE*

			nabled parameter specifies whether poison messages should be detected. The default value is `$true`. Poison messages are messages determined to be potentially harmful to the Exchange system after a server failure. Poison messages are put in the poison message queue. We recommend that you don't modify the default value unless Customer Service and Support advises you to do this.
PoisonThreshold	Optional	System.Int32	The *PoisonThreshold* parameter specifies the number of times a message can be rejected before it's classified as a poison message. The default value is 2. We recommend that you don't modify the default value unless Customer Service and Support advises you to do this. The valid input range for this parameter is from 1 through 10.

ProcessingSchedulerLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ProcessingSchedulerLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
ProcessingSchedulerLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ProcessingSchedulerLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ProcessingSchedulerLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
QueueLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *QueueLogMaxAge* parameter specifies the maximum age of the queue log files. Log files that are older than the specified value are deleted. The default value is 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 10 days for this parameter, use 10.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting

			the value of the *QueueLogMaxAge* parameter to 00:00:00 prevents the automatic removal of queue log files because of their age.
QueueLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *QueueLogMaxDirectorySize* parameter specifies the maximum size of the queue log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The default value is 200 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *QueueLogMaxFileSize* parameter must be less than or equal to the value of the *QueueLogMaxDirectorySize* parameter. The valid

			input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the queue log directory.
QueueLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *QueueLogMaxFileSize* parameter specifies the maximum size of the queue log files. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *QueueLogMaxFileSize* parameter must be less than or equal to the value of the *QueueLogMaxDirectorySize* parameter. The valid

			input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the queue log files.
QueueLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *QueueLogPath* parameter specifies the path of the queue log directory. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\QueueViewer`. Setting the value of this parameter to `$null` disables queue logging.
QueueMaxIdleTime	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *QueueMaxIdleTime* parameter specifies the period of time an empty delivery queue can exist before the queue is removed. The default value is 3 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 5

			minutes for this parameter, use 00:05:00. The valid input range for this parameter is from 00:00:05 through 01:00:00. We recommend that you don't modify the default value unless Customer Service and Support advises you to do this.
ReceiveProtocolLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ReceiveProtocolLogMaxAge* parameter specifies the maximum age of the Receive connector protocol log file. Log files that are older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 20 days for this parameter, use 20.00:00:00. The valid input range for this parameter is from 00:00:00 through

			24855.03:14:07. Setting the value of the *ReceiveProtocolLogMaxAge* parameter to 00:00:00 prevents the automatic removal of Receive connector protocol log files because of their age.
ReceiveProtocolLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *ReceiveProtocolLogMaxDirectorySize* parameter specifies the maximum size of the Receive connector protocol log directory shared by all the Receive connectors that exist on the server. When the maximum directory size is reached, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the

			ReceiveProtocolLogMaxFileSize parameter must be less than or equal to the value of the *ReceiveProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Receive connector protocol log directory.
ReceiveProtocolLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ReceiveProtocolLogMaxFileSize* parameter specifies the maximum size of the Receive connector protocol log files shared by all the Receive connectors that exist on the server. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes)

			• MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ReceiveProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *ReceiveProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Receive connector protocol log files.
ReceiveProtocolLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ReceiveProtocolLogPath* parameter specifies the path of the protocol log directory for all the Receive connectors that exist on the server. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\ProtocolLog\SmtpReceive`. Setting the value of this parameter to

			`$null` disables protocol logging for all Receive connectors on the server. However, setting this parameter to `$null` when the value of the *ProtocolLoggingLevel* attribute for any Receive connector on the server is `verbose` generates event log errors. The preferred method of disabling protocol logging is to use the **Set-ReceiveConnector** cmdlet to set the *ProtocolLoggingLevel* to `None` on each Receive connector.
RecipientValidationCacheEnabled	Optional	System.Boolean	The *RecipientValidationCacheEnabled* parameter specifies whether the recipient addresses used by transport agents, such as the Recipient Filtering agent, are cached. The default value is `$true` on Edge servers and `$false` for the Transport service on Mailbox servers.
ReplayDirectoryPath	Optional	Microsoft.Exchange.Da	The *ReplayDirectoryPath*

		ta.LocalLongFullPath	parameter specifies the path of the Replay directory. The Replay directory is used to resubmit exported messages and to receive messages from foreign gateway servers. The default location is `%ExchangeInstallPath%TransportRoles\Replay`. If the value of the *ReplayDirectoryPath* parameter is set to `$null`, the Replay directory is disabled.
ResourceLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ResourceLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
ResourceLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ResourceLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ResourceLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
RootDropDirectoryPath	Optional	System.String	The *RootDropDirectoryPath* parameter specifies the top-level location of the

			Drop directory used by all Foreign connectors defined in the Transport service on a Mailbox server. The value of the *RootDropDirectoryPath* parameter may be a local path, or a Universal Naming Convention (UNC) path to a remote server. By default, the *RootDropDirectoryPath* parameter is blank. This indicates the value of *RootDropDirectoryPath* is the Exchange installation folder. The *RootDropDirectoryPath* parameter is used with the *DropDirectory* parameter in the **Set-ForeignConnector** cmdlet to specify the location for outgoing messages going to the address spaces defined on the Foreign connector.
RoutingTableLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *RoutingTableLogMaxAge* parameter specifies the maximum routing table log age. Log files older than the specified value

			are deleted. The default value is 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 5 days for this parameter, use 5.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *RoutingTableLogMaxAge* parameter to 00:00:00 prevents the automatic removal of routing table log files because of their age.
RoutingTableLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *RoutingTableLogMaxDirectorySize* parameter specifies the maximum size of the routing table log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The default value is 50 MB.

			When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the routing table log directory.
RoutingTableLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *RoutingTableLogPath* parameter specifies the directory location where routing table log files should be stored. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Routing`. Setting the value of the *RoutingTableLogPath* parameter to `$null` disables routing table logging.
SendProtocolLogMaxA	Optional	Microsoft.Exchange.Da	The *SendProtocolLogMaxAge*

ge		ta.EnhancedTimeSpan	parameter specifies the Send connector protocol log file maximum age. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 25 days for this parameter, use 25.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *SendProtocolLogMaxAge* parameter to 00:00:00 prevents the automatic removal of Send connector protocol log files because of their age.
SendProtocolLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *SendProtocolLogMaxDirectorySize* parameter specifies the maximum size of the Send connector protocol log directory.

			When the maximum directory size is reached, the server deletes the oldest log files first. The minimum value is 1 MB. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *SendProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *SendProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Send connector protocol log directory.

SendProtocolLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *SendProtocolLogMaxFileSize* parameter specifies the maximum size of the Send connector protocol log files shared by all the Send connectors that exist on a server. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *SendProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *SendProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807

			bytes. If you enter a value of `unlimited`, no size limit is imposed on the Send connector protocol log files.
SendProtocolLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *SendProtocolLogPath* parameter specifies the location of protocol log storage for the Send connectors. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\ProtocolLog\SmtpSend`. Setting the value of this parameter to `$null` disables protocol logging for all Send connectors on the server. However, setting this parameter to `$null` when the value of the *ProtocolLoggingLevel* or *IntraOrgConnectorProtocolLoggingLevel* attribute for any Send connector on the server is `verbose` generates event log errors. The preferred method of disabling protocol logging is to use the **Set-SendConnector** cmdlet to set the *ProtocolLoggingLevel*

			parameter to None on each Send connector and to set the *IntraOrgConnectorProtocolLoggingLevel* parameter to None.
ServerStatisticsLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ServerStatisticsLogMaxAge* parameter specifies the maximum duration that the server statistics log files are kept. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 60 days for this parameter, use 60.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of this parameter to 00:00:00 prevents the automatic removal of server statistics log files.

ServerStatisticsLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ServerStatisticsLogMaxDirectorySize* parameter specifies the cap on the size of the server statistics log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The minimum value is 1 MB. The default value is 250 MB. When you enter a value, qualify the value with one of the following: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ServerStatisticsLogMaxFileSize* parameter must be less than or equal to the value of the *ServerStatisticsLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value

			of `unlimited`, no size limit is imposed on the server statistics log directory.
ServerStatisticsLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ServerStatisticsLogMaxFileSize* parameter specifies the maximum file size for the server statistics log files. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *ServerStatisticsLogMaxFileSize* parameter must be less than or equal to the value of the *ServerStatisticsLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through

			9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the server statistics log files.
ServerStatisticsLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ServerStatisticsLogPath* parameter specifies the location of server statistics log storage. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\ServerStats`. Setting the value of this parameter to `$null` disables server statistics logging.
SmtpEnableAllTlsVersions	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransientFailureRetryCount	Optional	System.Int32	The *TransientFailureRetryCount* parameter specifies the maximum number of immediate connection retries attempted when the server encounters a connection failure with a remote server. The default value is 6. The valid input range for this parameter is from 0 through 15.

			When the value of this parameter is set to 0, the server doesn't immediately attempt to retry an unsuccessful connection, and the next connection attempt is controlled by the *OutboundConnectionFailureRetryInterval* parameter.
TransientFailureRetryInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *TransientFailureRetryInterval* parameter controls the connection interval between each connection attempt specified by the *TransientFailureRetryCount* parameter. For the Transport service on a Mailbox server, the default value of the *TransientFailureRetryInterval* parameter is 5 minutes. On an Edge server, the default value is 10 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.

			For example, to specify 8 minutes for this parameter, use 00:08:00. The valid input range for this parameter is from 00:00:01 through 12:00:00.
TransportMaintenanceLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportMaintenanceLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
TransportMaintenanceLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
TransportMaintenanceLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
TransportMaintenanceLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
TransportSyncAccountsPoisonAccountThreshold	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
TransportSyncAccountsPoisonDetectionEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncAccountsPoisonItemThreshold	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
TransportSyncAccountsSuccessivePoisonItemThreshold	Optional	System.Int32	This parameter is reserved for internal Microsoft use.

TransportSyncEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncExchangeEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncFacebookEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncHubHealthLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncHubHealthLogFilePath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
TransportSyncHubHealthLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
TransportSyncHubHealthLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncHubHealthLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncImapEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncLinkedInEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncLogFilePath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
TransportSyncLogLoggingLevel	Optional	Microsoft.Exchange.Data.SyncLoggingLevel	This parameter is reserved for internal Microsoft use.

TransportSyncLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
TransportSyncLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncMaxDownloadItemsPerConnection	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
TransportSyncMaxDownloadSizePerConnection	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncMaxDownloadSizePerItem	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncPopEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncRemoteConnectionTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
UseDowngradedExchangeServerAuth	Optional	System.Boolean	The *UseDowngradedExchangeServerAuth* parameter specifies whether the Generic Security Services application programming interface (GSSAPI) authentication method is used on connections where Transport Layer Security (TLS) is disabled.

			Normally, TLS is required for connections between the Transport services on Mailbox servers in your organization. On TLS secured connections, Kerberos authentication is used by default. However, there may be scenarios where you need to disable TLS between specific Transport services in your organization. When you do that, you need to set this parameter to `$true` to provide an alternative authentication method. The default value is `$false`. You shouldn't set this value to `$true` unless it's absolutely required. If you set this parameter to `$true`, you also need to create a specific Receive connector to service the non-TLS connections. This Receive connector must have remote IP address ranges specified to ensure that it's only used for non-TLS connections. You also must set the *SuppressXAnonymousTls*

			attribute of the Receive connector to `$true`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsLiveHotmailTransportSyncEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
WlmLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
WlmLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
WlmLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
WlmLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-TransportService

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-TransportService** cmdlet to view the transport configuration information for the Transport service on Mailbox servers or for Edge Transport servers.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-TransportService [-Identity <TransportServerIdParameter>] [-
DomainController <Fqdn>]
```

Examples

Example 1

This example provides different results depending on the server role on which it's run. When you run this command on an Edge Transport server, it provides a configuration summary for the local server. Otherwise, it displays a list of all Mailbox servers in your organization.

```
Get-TransportService
```

Example 2

This example retrieves the detailed transport configuration information for the Transport service on the Mailbox server named Mailbox1.

```
Get-TransportService Mailbox1 | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Transport service" or "Edge Transport server" entries in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.TransportServerIdParameter	The *Identity* parameter specifies the server you want to view. When you use this parameter on a Mailbox server, the parameter returns the transport configuration of the Transport service on the specified server.

			You can't use this parameter on an Edge Transport server.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-TransportService

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-26*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-TransportService** cmdlet to set the transport configuration options for the Transport service on Mailbox servers or for Edge Transport servers.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-TransportService -Identity <ServerIdParameter> [-
ActiveUserStatisticsLogMaxAge <EnhancedTimeSpan>] [-
ActiveUserStatisticsLogMaxDirectorySize <ByteQuantifiedSize>] [-
ActiveUserStatisticsLogMaxFileSize <ByteQuantifiedSize>] [-
ActiveUserStatisticsLogPath <LocalLongFullPath>] [-AgentLogEnabled <$true
| $false>] [-AgentLogMaxAge <EnhancedTimeSpan>] [-AgentLogMaxDirectorySize
<Unlimited>] [-AgentLogMaxFileSize <Unlimited>] [-AgentLogPath
<LocalLongFullPath>] [-AntispamAgentsEnabled <$true | $false>] [-Confirm
[<SwitchParameter>]] [-ConnectivityLogEnabled <$true | $false>] [-
ConnectivityLogMaxAge <EnhancedTimeSpan>] [-
ConnectivityLogMaxDirectorySize <Unlimited>] [-ConnectivityLogMaxFileSize
<Unlimited>] [-ConnectivityLogPath <LocalLongFullPath>] [-
ContentConversionTracingEnabled <$true | $false>] [-
DelayNotificationTimeout <EnhancedTimeSpan>] [-
DeltaSyncClientCertificateThumbprint <String>] [-DnsLogEnabled <$true |
$false>] [-DnsLogMaxAge <EnhancedTimeSpan>] [-DnsLogMaxDirectorySize
<Unlimited>] [-DnsLogMaxFileSize <Unlimited>] [-DnsLogPath
<LocalLongFullPath>] [-DomainController <Fqdn>] [-
ExternalDNSAdapterEnabled <$true | $false>] [-ExternalDNSAdapterGuid
<Guid>] [-ExternalDNSProtocolOption <Any | UseUdpOnly | UseTcpOnly>] [-
ExternalDNSServers <MultiValuedProperty>] [-ExternalIPAddress <IPAddress>]
[-FlowControlLogEnabled <$true | $false>] [-FlowControlLogMaxAge
<EnhancedTimeSpan>] [-FlowControlLogMaxDirectorySize <Unlimited>] [-
```

```
FlowControlLogMaxFileSize <Unlimited>] [-FlowControlLogPath
<LocalLongFullPath>] [-HttpProtocolLogEnabled <$true | $false>] [-
HttpProtocolLogFilePath <LocalLongFullPath>] [-HttpProtocolLogLoggingLevel
<None | Verbose>] [-HttpProtocolLogMaxAge <EnhancedTimeSpan>] [-
HttpProtocolLogMaxDirectorySize <ByteQuantifiedSize>] [-
HttpProtocolLogMaxFileSize <ByteQuantifiedSize>] [-
HttpTransportSyncProxyServer <String>] [-InternalDNSAdapterEnabled <$true
| $false>] [-InternalDNSAdapterGuid <Guid>] [-InternalDNSProtocolOption
<Any | UseUdpOnly | UseTcpOnly>] [-InternalDNSServers
<MultiValuedProperty>] [-IntraOrgConnectorProtocolLoggingLevel <None |
Verbose>] [-IntraOrgConnectorSmtpMaxMessagesPerConnection <Int32>] [-
IrmLogEnabled <$true | $false>] [-IrmLogMaxAge <EnhancedTimeSpan>] [-
IrmLogMaxDirectorySize <Unlimited>] [-IrmLogMaxFileSize
<ByteQuantifiedSize>] [-IrmLogPath <LocalLongFullPath>] [-
JournalLogEnabled <$true | $false>] [-JournalLogMaxAge <EnhancedTimeSpan>]
[-JournalLogMaxDirectorySize <Unlimited>] [-JournalLogMaxFileSize
<Unlimited>] [-JournalLogPath <LocalLongFullPath>] [-
MaxActiveTransportSyncJobsPerProcessor <Int32>] [-
MaxConcurrentMailboxDeliveries <Int32>] [-MaxConcurrentMailboxSubmissions
<Int32>] [-MaxConnectionRatePerMinute <Int32>] [-
MaxNumberOfTransportSyncAttempts <Int32>] [-MaxOutboundConnections
<Unlimited>] [-MaxPerDomainOutboundConnections <Unlimited>] [-
MessageExpirationTimeout <EnhancedTimeSpan>] [-MessageRetryInterval
<EnhancedTimeSpan>] [-MessageTrackingLogEnabled <$true | $false>] [-
MessageTrackingLogMaxAge <EnhancedTimeSpan>] [-
MessageTrackingLogMaxDirectorySize <Unlimited>] [-
MessageTrackingLogMaxFileSize <ByteQuantifiedSize>] [-
MessageTrackingLogPath <LocalLongFullPath>] [-
MessageTrackingLogSubjectLoggingEnabled <$true | $false>] [-
OutboundConnectionFailureRetryInterval <EnhancedTimeSpan>] [-
PickupDirectoryMaxHeaderSize <ByteQuantifiedSize>] [-
PickupDirectoryMaxMessagesPerMinute <Int32>] [-
PickupDirectoryMaxRecipientsPerMessage <Int32>] [-PickupDirectoryPath
<LocalLongFullPath>] [-PipelineTracingEnabled <$true | $false>] [-
PipelineTracingPath <LocalLongFullPath>] [-PipelineTracingSenderAddress
<SmtpAddress>] [-PoisonMessageDetectionEnabled <$true | $false>] [-
PoisonThreshold <Int32>] [-ProcessingSchedulerLogEnabled <$true | $false>]
[-ProcessingSchedulerLogMaxAge <EnhancedTimeSpan>] [-
ProcessingSchedulerLogMaxDirectorySize <Unlimited>] [-
ProcessingSchedulerLogMaxFileSize <Unlimited>] [-
ProcessingSchedulerLogPath <LocalLongFullPath>] [-QueueLogMaxAge
<EnhancedTimeSpan>] [-QueueLogMaxDirectorySize <Unlimited>] [-
QueueLogMaxFileSize <Unlimited>] [-QueueLogPath <LocalLongFullPath>] [-
QueueMaxIdleTime <EnhancedTimeSpan>] [-ReceiveProtocolLogMaxAge
<EnhancedTimeSpan>] [-ReceiveProtocolLogMaxDirectorySize <Unlimited>] [-
ReceiveProtocolLogMaxFileSize <Unlimited>] [-ReceiveProtocolLogPath
<LocalLongFullPath>] [-RecipientValidationCacheEnabled <$true | $false>]
[-ReplayDirectoryPath <LocalLongFullPath>] [-ResourceLogEnabled <$true |
$false>] [-ResourceLogMaxAge <EnhancedTimeSpan>] [-
ResourceLogMaxDirectorySize <Unlimited>] [-ResourceLogMaxFileSize
<Unlimited>] [-ResourceLogPath <LocalLongFullPath>] [-
RootDropDirectoryPath <String>] [-RoutingTableLogMaxAge
<EnhancedTimeSpan>] [-RoutingTableLogMaxDirectorySize <Unlimited>] [-
RoutingTableLogPath <LocalLongFullPath>] [-SendProtocolLogMaxAge
<EnhancedTimeSpan>] [-SendProtocolLogMaxDirectorySize <Unlimited>] [-
SendProtocolLogMaxFileSize <Unlimited>] [-SendProtocolLogPath
<LocalLongFullPath>] [-ServerStatisticsLogMaxAge <EnhancedTimeSpan>] [-
ServerStatisticsLogMaxDirectorySize <ByteQuantifiedSize>] [-
ServerStatisticsLogMaxFileSize <ByteQuantifiedSize>] [-
ServerStatisticsLogPath <LocalLongFullPath>] [-SmtpEnableAllTlsVersions
<$true | $false>] [-TransientFailureRetryCount <Int32>] [-
TransientFailureRetryInterval <EnhancedTimeSpan>] [-
TransportMaintenanceLogEnabled <$true | $false>] [-
TransportMaintenanceLogMaxAge <EnhancedTimeSpan>] [-
TransportMaintenanceLogMaxDirectorySize <Unlimited>] [-
TransportMaintenanceLogMaxFileSize <Unlimited>] [-
TransportMaintenanceLogPath <LocalLongFullPath>] [-
TransportSyncAccountsPoisonAccountThreshold <Int32>] [-
TransportSyncAccountsPoisonDetectionEnabled <$true | $false>] [-
TransportSyncAccountsPoisonItemThreshold <Int32>] [-
TransportSyncAccountsSuccessivePoisonItemThreshold <Int32>] [-
TransportSyncEnabled <$true | $false>] [-TransportSyncExchangeEnabled
<$true | $false>] [-TransportSyncFacebookEnabled <$true | $false>] [-
TransportSyncHubHealthLogEnabled <$true | $false>] [-
TransportSyncHubHealthLogFilePath <LocalLongFullPath>] [-
TransportSyncHubHealthLogMaxAge <EnhancedTimeSpan>] [-
TransportSyncHubHealthLogMaxDirectorySize <ByteQuantifiedSize>] [-
TransportSyncHubHealthLogMaxFileSize <ByteQuantifiedSize>] [-
```

```
TransportSyncImapEnabled <$true | $false>] [-TransportSyncLinkedInEnabled
<$true | $false>] [-TransportSyncLogEnabled <$true | $false>] [-
TransportSyncLogFilePath <LocalLongFullPath>] [-
TransportSyncLogLoggingLevel <None | Error | Information | Verbose |
RawData | Debugging>] [-TransportSyncLogMaxAge <EnhancedTimeSpan>] [-
TransportSyncLogMaxDirectorySize <ByteQuantifiedSize>] [-
TransportSyncLogMaxFileSize <ByteQuantifiedSize>] [-
TransportSyncMaxDownloadItemsPerConnection <Int32>] [-
TransportSyncMaxDownloadSizePerConnection <ByteQuantifiedSize>] [-
TransportSyncMaxDownloadSizePerItem <ByteQuantifiedSize>] [-
TransportSyncPopEnabled <$true | $false>] [-
TransportSyncRemoteConnectionTimeout <EnhancedTimeSpan>] [-
UseDowngradedExchangeServerAuth <$true | $false>] [-WhatIf
[<SwitchParameter>]] [-WindowsLiveHotmailTransportSyncEnabled <$true |
$false>] [-WlmLogMaxAge <EnhancedTimeSpan>] [-WlmLogMaxDirectorySize
<Unlimited>] [-WlmLogMaxFileSize <Unlimited>] [-WlmLogPath
<LocalLongFullPath>]
```

Examples

EXAMPLE 1

This example sets the *DelayNotificationTimeout* parameter to 13 hours for the Transport service on a Mailbox server named Mailbox01.

```
Set-TransportService Mailbox01 -DelayNotificationTimeout
13:00:00
```

EXAMPLE 2

This example sets the *TransientFailureRetryCount* parameter to 3 and sets the *TransientFailureRetryInterval* parameter to 30 seconds for the Transport service on a Mailbox server named Mailbox01.

```
Set-TransportService Mailbox01 -TransientFailureRetryCount
3 -TransientFailureRetryInterval 00:00:30
```

EXAMPLE 3

This example sets the *ReceiveProtocolLogPath* parameter to C:\SMTP Protocol Logs\Receive.log for the Transport service on a Mailbox server named Mailbox01.

```
Set-TransportService Mailbox01 -ReceiveProtocolLogPath "C:
\SMTP Protocol Logs\Receive.log"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport service" and "Edge Transport server" entries in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the server that you want to modify.
ActiveUserStatisticsLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ActiveUserStatisticsLogMaxAge* parameter specifies the maximum duration that the per user activity statistics log files are kept. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 10 days for this parameter, use 10.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of this parameter to 00:00:00 prevents the automatic removal of server

			statistics log files.
ActiveUserStatisticsLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ActiveUserStatisticsLogMaxDirectorySize* parameter specifies the cap on the size of the per user activity statistics log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The minimum value is 1 megabyte (MB). The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ActiveUserStatisticsLogMaxFileSize* parameter must be less than or equal to the value of the *ActiveUserStatisticsLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1

			through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the server statistics log directory.
ActiveUserStatisticsLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ActiveUserStatisticsLogMaxFileSize* parameter specifies the maximum file size for the per user activity statistics log files. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *ActiveUserStatisticsLogMaxFileSize* parameter must be less than or equal to the value of the *ActiveUserStatisticsLogMaxDirectorySize* parameter.

			The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the server statistics log files.
ActiveUserStatisticsLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ActiveUserStatisticsLogPath* parameter specifies the location of per user activity statistics log storage. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\ActiveUsersStats`. Setting the value of this parameter to `$null` disables server statistics logging.
AgentLogEnabled	Optional	System.Boolean	The *AgentLogEnabled* parameter specifies whether the agent log is enabled. The default value is `$true`.
AgentLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *AgentLogMaxAge* parameter specifies the maximum age for the agent log file. Log files older than the specified

			value are deleted. The default value is `7.00:00:00` or 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Setting the value of the *AgentLogMaxAge* parameter to 00:00:00 prevents the automatic removal of agent log files because of their age.
AgentLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *AgentLogMaxDirectorySize* parameter specifies the maximum size of all agent logs in the agent log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes)

			Unqualified values are treated as bytes. The value of the *AgentLogMaxFileSize* parameter must be less than or equal to the value of the *AgentLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the agent log directory.
AgentLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *AgentLogMaxFileSize* parameter specifies the maximum size of each agent log file. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *AgentLogMaxFileSize* parameter must be less

			than or equal to the value of the *AgentLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the agent log files.
AgentLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *AgentLogPath* parameter specifies the default agent log directory location. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\AgentLog`. Setting the value of this parameter to `$null` disables agent logging. However, setting this parameter to `$null` when the value of the *AgentLogEnabled* attribute is `$true` generates event log errors.
AntispamAgentsEnabled	Optional	System.Boolean	The *AntispamAgentsEnabled* parameter specifies whether anti-spam agents are installed on the server specified with the *Identity* parameter. The default value is `$false` for the

			Transport service on Mailbox servers and `$true` on Edge Transport servers. **Important:** You set this parameter by using a script. You shouldn't modify this parameter manually.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ConnectivityLogEnabled	Optional	System.Boolean	The *ConnectivityLogEnabled* parameter specifies whether the connectivity log is enabled. The default value is `$true`.
ConnectivityLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ConnectivityLogMaxAge* parameter specifies the maximum age for the connectivity log file. Log files older than the specified value are deleted. The default value

			is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 25 days for this parameter, use 25.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *ConnectivityLogMaxAge* parameter to 00:00:00 prevents the automatic removal of connectivity log files because of their age.
ConnectivityLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *ConnectivityLogMaxDirectorySize* parameter specifies the maximum size of all connectivity logs in the connectivity log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 1000 MB.

			When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ConnectivityLogMaxFileSize* parameter must be less than or equal to the value of the *ConnectivityLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the connectivity log directory.
ConnectivityLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ConnectivityLogMaxFileSize* parameter specifies the maximum size of each connectivity log file. When a log file reaches its maximum file size, a new log file is created. The

			default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *ConnectivityLogMaxFileSize* parameter must be less than or equal to the value of the *ConnectivityLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the connectivity log files.
ConnectivityLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ConnectivityLogPath* parameter specifies the default connectivity log directory location. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\Connectivity`.

			Setting the value of this parameter to `$null` disables connectivity logging. However, setting this parameter to `$null` when the value of the *ConnectivityLogEnabled* attribute is `$true` generates event log errors.
ContentConversionTracingEnabled	Optional	System.Boolean	The *ContentConversionTracingEnabled* parameter specifies whether content conversion tracing is enabled. Content conversion tracing captures content conversion failures that occur in the Transport service on a Mailbox server or on the Edge Transport server. The default value is `$false`. Content conversion tracing captures a maximum of 128 MB of content conversion failures. When the 128 MB limit is reached, no more content conversion failures are captured. Content conversion

			tracing captures the complete contents of email messages to the path specified by the *PipelineTracingPath* parameter. Make sure that you restrict access to this directory. The permissions required on the directory specified by the *PipelineTracingPath* parameter are as follows: • Administrators: Full Control • Network Service: Full Control • System: Full Control
DelayNotificationTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *DelayNotificationTimeout* parameter specifies how long the server waits before it generates a delayed delivery status notification (DSN) message. The default value is 4 hours. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 3.5

			hours for this parameter, use 03:30:00. The valid input range for this parameter is from 00:00:01 through 30.00:00:00. The value of the *DelayNotificationTimeout* parameter should always be greater than the value of the *TransientFailureRetryCount* parameter multiplied by the value of the *TransientFailureRetryInterval* parameter.
DeltaSyncClientCertificateThumbprint	Optional	System.String	This parameter is reserved for internal Microsoft use.
DnsLogEnabled	Optional	System.Boolean	The *DnsLogEnabled* parameter specifies whether the DNS log is enabled. The default value is `$false`.
DnsLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *DnsLogMaxAge* parameter specifies the maximum age for the DNS log file. Log files older than the specified value are deleted. The default value is `7.00:00:00` or 7 days.

			To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. Setting the value of the *DnsLogMaxAge* parameter to `00:00:00` prevents the automatic removal of DNS log files because of their age.
DnsLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *DnsLogMaxDirectorySize* parameter specifies the maximum size of all DNS logs in the DNS log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 100 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the

			DnsLogMaxFileSize parameter must be less than or equal to the value of the *DnsLogMaxDirectorySize* parameter. If you enter a value of `unlimited`, no size limit is imposed on the DNS log directory.
DnsLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *DnsLogMaxFileSize* parameter specifies the maximum size of each DNS log file. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *DnsLogMaxFileSize* parameter must be less than or equal to the value of the *DnsLogMaxDirectorySize*

			parameter. If you enter a value of `unlimited`, no size limit is imposed on the DNS log files.
DnsLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *DnsLogPath* parameter specifies the DNS log directory location. The default value is blank (`$null`), which indicates no location is configured. If you enable DNS logging, you need to specify a local file path for the DNS log files by using this parameter. If the path contains spaces, enclose the entire path value in quotation marks (").
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active

			Directory Lightweight Directory Services (AD LDS) to read and write data.
ExternalDNSAdapterEnabled	Optional	System.Boolean	The *ExternalDNSAdapterEnabled* parameter specifies one or more Domain Name System (DNS) servers that Exchange uses for external DNS lookups. When the *ExternalDNSAdapterEnabled* parameter is set to `$true`, DNS lookups of destinations outside the Exchange organization are performed by using the DNS settings of the external network adapter specified by the value of the *ExternalDNSAdapterGuid* parameter. If you want to specify a custom list of DNS servers used for external Exchange DNS lookups only, you must specify the DNS servers by using the *ExternalDNSServers* parameter, and you must

			also set the value of the *ExternalDNSAdapterEnabled* parameter to `$false`. The default value of the *ExternalDNSAdapterEnabled* parameter is `$true`.
ExternalDNSAdapterGuid	Optional	System.Guid	The *ExternalDNSAdapterGuid* parameter specifies the network adapter that has the DNS settings used for DNS lookups of destinations that exist outside the Exchange organization. The concept of an external network adapter and an internal network adapter is only applicable in a multi-homed Exchange server environment. When no particular network adapter is specified as the network adapter for external DNS lookups, the value of the *ExternalDNSAdapterGuid* parameter is `00000000-0000-0000-0000-000000000000`, and external DNS lookups are performed by using the DNS settings of any

			available network adapter. You may enter the GUID of a specific network adapter to use for external DNS lookups. The default value of the *ExternalDNSAdapterGuid* parameter is `00000000-0000-0000-0000-000000000000`. **Note:** If the value of the *ExternalDNSAdapterEnabled* parameter is set to `$false`, the value of the *ExternalDNSAdapterGuid* parameter is ignored, and the list of DNS servers from the *ExternalDNSServers* parameter is used.
ExternalDNSProtocolOption	Optional	Microsoft.Exchange.Data.ProtocolOption	The *ExternalDNSProtocolOption* parameter specifies which protocol to use when querying external DNS servers. The valid options for this parameter are `Any`, `UseTcpOnly`, and `UseUdpOnly`. The default value is `Any`.
ExternalDNSServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExternalDNSServers* parameter specifies the list of external DNS servers that the server queries when resolving a

			remote domain. DNS servers are specified by IP address. The default value is an empty list ({}). To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. **Note:** If the value of the *ExternalDNSAdapterEnabled* parameter is set to `$true`, the *ExternalDNSServers* parameter and its list of DNS servers isn't used.
ExternalIPAddress	Optional	System.Net.IPAddress	The *ExternalIPAddress* parameter specifies the IP address used in the

			`Received` message header field for every message that travels through the Edge Transport server or the Transport service on a Mailbox server. The IP address in the `Received` header field is used for hop count and routing loop detection. The IP address specified by the *ExternalIPAddress* parameter overrides the external network adapter's actual IP address. Typically, you would want to set the value of the *ExternalIPAddress* parameter to match the value of your domain's public MX record. The default value of the *ExternalIPAddress* parameter is blank. This means the actual IP address of the external network adapter is used in the `Received` header field.
FlowControlLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
FlowControlLogMaxA	Optional	Microsoft.Exchange.Da	This parameter is reserved

ge		ta.EnhancedTimeSpan	for internal Microsoft use.
FlowControlLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
FlowControlLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
FlowControlLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
HttpProtocolLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
HttpProtocolLogFilePath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
HttpProtocolLogLoggingLevel	Optional	Microsoft.Exchange.Data.ProtocolLoggingLevel	This parameter is reserved for internal Microsoft use.
HttpProtocolLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
HttpProtocolLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
HttpProtocolLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
HttpTransportSyncProxyServer	Optional	System.String	This parameter is reserved for internal Microsoft use.
InternalDNSAdapterEnabled	Optional	System.Boolean	The *InternalDNSAdapterEnabled* parameter specifies one or more DNS servers that Exchange uses for

			internal DNS lookups. When the *InternalDNSAdapterEnabled* parameter is set to `$true`, DNS lookups of destinations inside the Exchange organization are performed by using the DNS settings of the internal network adapter specified by the value of the *InternalDNSAdapterGuid* parameter. If you want to specify a custom list of DNS servers used for internal Exchange DNS lookups only, you must specify the DNS servers by using the *InternalDNSServers* parameter, and you must also set the value of the *InternalDNSAdapterEnabled* parameter to `$false`. The default value of the *InternalDNSAdapterEnabled* parameter is `$true`.
InternalDNSAdapterGuid	Optional	System.Guid	The *InternalDNSAdapterGuid* parameter specifies the network adapter that has the DNS settings used for

			DNS lookups of servers that exist inside the Exchange organization. The concept of an internal network adapter and an external network adapter is only applicable in a multi-homed Exchange server environment. When no particular network adapter is specified as the network adapter for external DNS lookups, the value of the *InternalDNSAdapterGuid* parameter is `00000000-0000-0000-0000-000000000000`, and internal DNS lookups are performed by using the DNS settings of any available network adapter. You may enter the GUID of a specific network adapter to use for internal DNS lookups. The default value of the *InternalDNSAdapterGuid* parameter is `00000000-0000-0000-0000-000000000000`. **Note:** If the value of the *InternalDNSAdapterEnabled* parameter is set to

			`$false`, the value of the *InternalDNSAdapterGuid* parameter is ignored, and the list of DNS servers from the *InternalDNSServers* parameter is used.
InternalDNSProtocolOption	Optional	Microsoft.Exchange.Data.ProtocolOption	The *InternalDNSProtocolOption* parameter specifies which protocol to use when you query internal DNS servers. Valid options for this parameter are `Any`, `UseTcpOnly`, or `UseUdpOnly`.The default value is `Any`.
InternalDNSServers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *InternalDNSServers* parameter specifies the list of DNS servers that should be used when resolving a domain name. DNS servers are specified by IP address. The default value is any empty list ({}). To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following

			syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. **Note:** If the *InternalDNSAdapterGuid* parameter is set, and the value of the *InternalDNSAdapterEnabled* parameter is set to `$true`, the *InternalDNSServers* parameter and its list of DNS servers isn't used.
IntraOrgConnectorProtocolLoggingLevel	Optional	Microsoft.Exchange.Data.ProtocolLoggingLevel	The *IntraOrgConnectorProtocolLoggingLevel* parameter enables or disables SMTP protocol logging on the implicit and invisible intra-organization Send connectors that are used to transmit messages between Exchange servers in the Exchange organization. Valid values for this

			parameter are `None` or `Verbose`. The value `Verbose` enables protocol logging for the connector. The value `None` disables protocol logging for the connector. The default value is `None`. When the *IntraOrgConnectorProtocolLoggingLevel* parameter is set to `Verbose`, the information is written to the Send connector protocol log specified by the *SendProtocolLog* parameters.
IntraOrgConnectorSmtpMaxMessagesPerConnection	Optional	System.Int32	The *IntraOrgConnectorSmtpMaxMessagesPerConnection* parameter specifies the maximum number of messages per connection limit on the implicit and invisible intra-organization Send connectors that are used to transmit messages between Exchange servers in the Exchange organization.
IrmLogEnabled	Optional	System.Boolean	The *IrmLogEnabled* parameter enables

			logging of Information Rights Management (IRM) transactions. IRM logging is enabled by default. Values include: • `$true` Enable IRM logging • `$false` Disable IRM logging
IrmLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *IrmLogMaxAge* parameter specifies the maximum age for the IRM log file. Log files that are older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The valid input range for this parameter is 00:00:00 to 24855.03:14:07. Setting the value of the *IrmLogMaxAge* parameter to 00:00:00 prevents the automatic removal of connectivity log files because of their age.
IrmLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This *IrmLogMaxDirectorySize* parameter specifies the

			maximum size of all IRM logs in the connectivity log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *IrmLogMaxFileSize* parameter must be less than or equal to the value of the *IrmLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the IRM log directory.

IrmLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This *IrmLogMaxFileSize* parameter specifies the maximum size of each IRM log file. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *IrmLogMaxFileSize* parameter must be less than or equal to the value of the *IrmLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the IRM log files.

IrmLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *IrmLogPath* parameter specifies the default IRM log directory location. The default location is `%ExchangeInstallPath%Logging\IRMLogs`. If you set the value of the *IrmLogPath* parameter to `$null`, you effectively disable IRM logging. However, if you set the value of the *IrmLogPath* parameter to `$null` when the value of the *IrmLogEnabled* attribute is `$true`, Exchange will log errors in the Application event log. The preferred way for disabling IRM logging is to set the *IrmLogEnabled* to `$false`.
JournalLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
JournalLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
JournalLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
JournalLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
JournalLogPath	Optional	Microsoft.Exchange.Da	This parameter is reserved

		ta.LocalLongFullPath	for internal Microsoft use.
MaxActiveTransportSyncJobsPerProcessor	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
MaxConcurrentMailboxDeliveries	Optional	System.Int32	The *MaxConcurrentMailboxDeliveries* parameter specifies the maximum number of delivery threads that the transport service can have open at the same time to deliver messages to mailboxes. The default value is 20. The valid input range for this parameter is from 1 through 256. We recommend that you don't modify the default value unless Microsoft Customer Service and Support advises you to do this.
MaxConcurrentMailboxSubmissions	Optional	System.Int32	The *MaxConcurrentMailboxSubmissions* parameter specifies the maximum number of submission threads that the transport service can have open at the same time to send messages from mailboxes.

			The default value is 20. The valid input range for this parameter is from 1 through 256.
MaxConnectionRatePerMinute	Optional	System.Int32	The *MaxConnectionRatePerMinute* parameter specifies the maximum rate that connections are allowed to be opened with the transport service. If many connections are attempted with the transport service at the same time, the *MaxConnectionRatePerMinute* parameter limits the rate that the connections are opened so that the server's resources aren't overwhelmed. The default value is 1200 connections per minute. The valid input range for this parameter is from 1 through 2147483647.
MaxNumberOfTransportSyncAttempts	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
MaxOutboundConnections	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxOutboundConnections* parameter specifies the

			maximum number of outbound connections that can be open at a time. The default value is 1000. The valid input range for this parameter is from 1 through 2147483647. If you enter a value of `unlimited`, no limit is imposed on the number of outbound connections. The value of the *MaxOutboundConnections* parameter must be greater than or equal to the value of the *MaxPerDomainOutboundConnections* parameter.
MaxPerDomainOutboundConnections	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxPerDomainOutboundConnections* parameter specifies the maximum number of concurrent connections to any single domain. The default value is 20. The valid input range for this parameter is from 1 through 2147483647. If you enter a value of `unlimited`, no limit is imposed on the number of outbound

			connections per domain. The value of the *MaxPerDomainOutboundConnections* parameter must be less than or equal to the value of the *MaxOutboundConnections* parameter.
MessageExpirationTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MessageExpirationTimeout* parameter specifies the maximum time that a particular message can remain in a queue. If a message remains in the queue for longer than this period of time, the message is returned to the sender as a permanent failure. The default value is 2 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 3 days for this parameter, use 3.00:00:00. The valid input range for this parameter is from 00:00:05 through

			90.00:00:00.
MessageRetryInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MessageRetryInterval* parameter specifies the retry interval for individual messages after a connection failure with a remote server. The default value is 1 minute. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 2 minutes for this parameter, use 00:02:00. The valid input range for this parameter is from 00:00:01 through 1.00:00:00. We recommend that you don't modify the default value unless Customer Service and Support advises you to do this.
MessageTrackingLogEnabled	Optional	System.Boolean	The *MessageTrackingLogEnabled* parameter specifies whether message tracking is enabled. The default value is `$true`.

MessageTrackingLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MessageTrackingLogMaxAge* parameter specifies the message tracking log maximum file age. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 20 days for this parameter, use 20.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *MessageTrackingLogMaxAge* parameter to 00:00:00 prevents the automatic removal of message tracking log files because of their age.
MessageTrackingLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *MessageTrackingLogMaxDirectorySize* parameter specifies the maximum

			size of the message tracking log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The maximum size of the message tracking log directory is calculated as the total size of all log files that have the same name prefix. Other files that don't follow the name prefix convention aren't counted in the total directory size calculation. Renaming old log files or copying other files into the message tracking log directory could cause the directory to exceed its specified maximum size. For Mailbox servers, the maximum size of the message tracking log directory isn't the specified maximum size because the message tracking log files generated by the Transport service and the Mailbox Transport service have different name

			prefixes. Message tracking log files for the Transport service or for Edge Transport servers begin with the name prefix *MSGTRK*. Message tracking log files for the Mailbox Transport service begin with the name prefix *MSGTRKM*. For Mailbox servers, the maximum size of the message tracking log directory is two times the specified value. The default value is 1000 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *MessageTrackingLogMaxFileSize* parameter must be less than or equal to the value of the *MessageTrackingLogMax*

			DirectorySize parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the message tracking log directory.
MessageTrackingLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *MessageTrackingLogMaxFileSize* parameter specifies the maximum size of the message tracking log files. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *MessageTrackingLogMaxFileSize* parameter must be less than or equal to the

			value of the *MessageTrackingLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 4294967296 bytes (4 GB). If you enter a value of `unlimited`, no size limit is imposed on the message tracking log files.
MessageTrackingLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *MessageTrackingLogPath* parameter specifies the location of the message tracking logs. The default location is `%ExchangeInstallPath%TransportRoles\Logs\MessageTracking`. Setting the value of this parameter to `$null` disables message tracking. However, setting this parameter to `$null` when the value of the *MessageTrackingLogEnabled* attribute is `$true` generates event log errors. The preferred method to disable message tracking is to use the

			MessageTrackingLogEnabled parameter.
MessageTrackingLogSubjectLoggingEnabled	Optional	System.Boolean	The *MessageTrackingLogSubjectLoggingEnabled* parameter specifies whether the message subject should be included in the message tracking log. The default value is `$true`.
OutboundConnectionFailureRetryInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *OutboundConnectionFailureRetryInterval* parameter specifies the retry interval for subsequent connection attempts to a remote server where previous connection attempts have failed. The previously failed connection attempts are controlled by the *TransientFailureRetryCount* and *TransientFailureRetryInterval* parameters. For the Transport service on a Mailbox server, the default value of the *OutboundConnectionFailureRetryInterval*

			parameter is 10 minutes. On an Edge Transport server, the default value is 30 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 15 minutes for this parameter, use 00:15:00. The valid input range for this parameter is from 00:00:01 through 20.00:00:00.
PickupDirectoryMaxHeaderSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *PickupDirectoryMaxHeaderSize* parameter specifies the maximum message header size that can be submitted to the Pickup directory. The default value is 64 KB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are

			treated as bytes. The valid input range for this parameter is from 32768 through 2147483647 bytes.
PickupDirectoryMaxMessagesPerMinute	Optional	System.Int32	The *PickupDirectoryMaxMessagesPerMinute* parameter specifies the maximum number of messages processed per minute by the Pickup directory and by the Replay directory. Each directory can independently process message files at the rate specified by the *PickupDirectoryMaxMessagesPerMinute* parameter. The default value is 100. The valid input range for this parameter is from 1 through 20000.
PickupDirectoryMaxRecipientsPerMessage	Optional	System.Int32	The *PickupDirectoryMaxRecipientsPerMessage* parameter specifies the maximum number of recipients that can be included on an email message. The default value is 100. The valid

			input range for this parameter is from 1 through 10000.
PickupDirectoryPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *PickupDirectoryPath* parameter specifies the location of the Pickup directory. The Pickup directory is used by administrators and third-party applications to create and submit messages. The default location is `%ExchangeInstallPath%TransportRoles\Pickup`. If the value of the *PickupDirectoryPath* parameter is set to `$null`, the Pickup directory is disabled.
PipelineTracingEnabled	Optional	System.Boolean	The *PipelineTracingEnabled* parameter specifies whether to enable pipeline tracing. Pipeline tracing captures message snapshot files that record the changes made to the message by each transport agent configured in the transport service on the

			server. Pipeline tracing creates verbose log files that accumulate quickly. Pipeline tracing should only be enabled for a short time to provide in-depth diagnostic information that enables you to troubleshoot problems. In addition to troubleshooting, you can use pipeline tracing to validate changes that you make to the configuration of the transport service where you enable pipeline tracing. The default value is `$false`.
PipelineTracingPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *PipelineTracingPath* parameter specifies the location of the pipeline tracing logs. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\PipelineTracing`. The path must be local to the Exchange computer. Setting the value of this parameter to `$null` disables pipeline tracing. However, setting this parameter to `$null` when the value of the

			PipelineTracingEnabled attribute is `$true` generates event log errors. The preferred method to disable pipeline tracing is to use the *PipelineTracingEnabled* parameter. Pipeline tracing captures the complete contents of email messages to the path specified by the *PipelineTracingPath* parameter. Make sure that you restrict access to this directory. The permissions required on the directory specified by the *PipelineTracingPath* parameter are as follows: • Administrators: Full Control • Network Service: Full Control • System: Full Control
PipelineTracingSenderAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PipelineTracingSenderAddress* parameter specifies the sender email address that invokes pipeline tracing. Only messages from this address

			generate pipeline tracing output. The address can be either inside or outside the Exchange organization. Depending on your requirements, you may have to set this parameter to different sender addresses and send new messages to start the transport agents or routes that you want to test. The default value of this parameter is `$null`.
PoisonMessageDetectionEnabled	Optional	System.Boolean	The *PoisonMessageDetectionEnabled* parameter specifies whether poison messages should be detected. The default value is `$true`. Poison messages are messages determined to be potentially harmful to the Exchange system after a server failure. Poison messages are put in the poison message queue. We recommend that you don't modify the default value unless Customer Service and Support

			advises you to do this.
PoisonThreshold	Optional	System.Int32	The *PoisonThreshold* parameter specifies the number of times a message can be rejected before it's classified as a poison message. The default value is 2. We recommend that you don't modify the default value unless Customer Service and Support advises you to do this. The valid input range for this parameter is from 1 through 10.
ProcessingSchedulerLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ProcessingSchedulerLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
ProcessingSchedulerLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ProcessingSchedulerLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ProcessingSchedulerLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
QueueLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *QueueLogMaxAge* parameter specifies the maximum age of the queue log files. Log files

			that are older than the specified value are deleted. The default value is 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 10 days for this parameter, use 10.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *QueueLogMaxAge* parameter to 00:00:00 prevents the automatic removal of queue log files because of their age.
QueueLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *QueueLogMaxDirectorySize* parameter specifies the maximum size of the queue log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The default value is 200 MB.

			When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *QueueLogMaxFileSize* parameter must be less than or equal to the value of the *QueueLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the queue log directory.
QueueLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *QueueLogMaxFileSize* parameter specifies the maximum size of the queue log files. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB.

			When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *QueueLogMaxFileSize* parameter must be less than or equal to the value of the *QueueLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the queue log files.
QueueLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *QueueLogPath* parameter specifies the path of the queue log directory. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\QueueViewer`. Setting the value of this

			parameter to `$null` disables queue logging.
QueueMaxIdleTime	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *QueueMaxIdleTime* parameter specifies the period of time an empty delivery queue can exist before the queue is removed. The default value is 3 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 5 minutes for this parameter, use 00:05:00. The valid input range for this parameter is from 00:00:05 through 01:00:00. We recommend that you don't modify the default value unless Customer Service and Support advises you to do this.
ReceiveProtocolLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ReceiveProtocolLogMaxAge* parameter specifies the maximum age of the Receive connector protocol log file. Log files

			that are older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 20 days for this parameter, use 20.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *ReceiveProtocolLogMaxAge* parameter to 00:00:00 prevents the automatic removal of Receive connector protocol log files because of their age.
ReceiveProtocolLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *ReceiveProtocolLogMaxDirectorySize* parameter specifies the maximum size of the Receive connector protocol log directory shared by all the Receive connectors that exist on the server. When

			the maximum directory size is reached, the server deletes the oldest log files first. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ReceiveProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *ReceiveProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Receive connector protocol log directory.
ReceiveProtocolLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ReceiveProtocolLogMaxFil*

			eSize parameter specifies the maximum size of the Receive connector protocol log files shared by all the Receive connectors that exist on the server. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ReceiveProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *ReceiveProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value

			of `unlimited`, no size limit is imposed on the Receive connector protocol log files.
ReceiveProtocolLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ReceiveProtocolLogPath* parameter specifies the path of the protocol log directory for all the Receive connectors that exist on the server. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\ProtocolLog\SmtpReceive`. Setting the value of this parameter to `$null` disables protocol logging for all Receive connectors on the server. However, setting this parameter to `$null` when the value of the *ProtocolLoggingLevel* attribute for any Receive connector on the server is `verbose` generates event log errors. The preferred method of disabling protocol logging is to use the **Set-ReceiveConnector** cmdlet to set the *ProtocolLoggingLevel* to

			`None` on each Receive connector.
RecipientValidationCacheEnabled	Optional	System.Boolean	The *RecipientValidationCacheEnabled* parameter specifies whether the recipient addresses used by transport agents, such as the Recipient Filtering agent, are cached. The default value is `$true` on Edge Transport servers and `$false` for the Transport service on Mailbox servers.
ReplayDirectoryPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ReplayDirectoryPath* parameter specifies the path of the Replay directory. The Replay directory is used to resubmit exported messages and to receive messages from foreign gateway servers. The default location is `%ExchangeInstallPath%TransportRoles\Replay`. If the value of the *ReplayDirectoryPath* parameter is set to `$null`, the Replay directory is disabled.

ResourceLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ResourceLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
ResourceLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ResourceLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
ResourceLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
RootDropDirectoryPath	Optional	System.String	The *RootDropDirectoryPath* parameter specifies the top-level location of the Drop directory used by all Foreign connectors defined in the Transport service on a Mailbox server. The value of the *RootDropDirectoryPath* parameter may be a local path, or a Universal Naming Convention (UNC) path to a remote server. By default, the *RootDropDirectoryPath* parameter is blank. This indicates the value of *RootDropDirectoryPath* is the Exchange installation

			folder. The *RootDropDirectoryPath* parameter is used with the *DropDirectory* parameter in the **Set-ForeignConnector** cmdlet to specify the location for outgoing messages going to the address spaces defined on the Foreign connector.
RoutingTableLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *RoutingTableLogMaxAge* parameter specifies the maximum routing table log age. Log files older than the specified value are deleted. The default value is 7 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 5 days for this parameter, use 5.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the

			RoutingTableLogMaxAge parameter to 00:00:00 prevents the automatic removal of routing table log files because of their age.
RoutingTableLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *RoutingTableLogMaxDirectorySize* parameter specifies the maximum size of the routing table log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The default value is 50 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit

			is imposed on the routing table log directory.
RoutingTableLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *RoutingTableLogPath* parameter specifies the directory location where routing table log files should be stored. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Routing`. Setting the value of the *RoutingTableLogPath* parameter to `$null` disables routing table logging.
SendProtocolLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *SendProtocolLogMaxAge* parameter specifies the Send connector protocol log file maximum age. Log files older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 25 days for this parameter,

			use 25.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *SendProtocolLogMaxAge* parameter to 00:00:00 prevents the automatic removal of Send connector protocol log files because of their age.
SendProtocolLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *SendProtocolLogMaxDirectorySize* parameter specifies the maximum size of the Send connector protocol log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The minimum value is 1 MB. The default value is 250 MB. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes)

			Unqualified values are treated as bytes. The value of the *SendProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *SendProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Send connector protocol log directory.
SendProtocolLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	The *SendProtocolLogMaxFileSize* parameter specifies the maximum size of the Send connector protocol log files shared by all the Send connectors that exist on a server. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB. When you enter a value, qualify the value with one

			of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *SendProtocolLogMaxFileSize* parameter must be less than or equal to the value of the *SendProtocolLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the Send connector protocol log files.
SendProtocolLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *SendProtocolLogPath* parameter specifies the location of protocol log storage for the Send connectors. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\ProtocolLog\SmtpSend`. Setting the

			value of this parameter to `$null` disables protocol logging for all Send connectors on the server. However, setting this parameter to `$null` when the value of the *ProtocolLoggingLevel* or *IntraOrgConnectorProtocolLoggingLevel* attribute for any Send connector on the server is `Verbose` generates event log errors. The preferred method of disabling protocol logging is to use the **Set-SendConnector** cmdlet to set the *ProtocolLoggingLevel* parameter to `None` on each Send connector and to set the *IntraOrgConnectorProtocolLoggingLevel* parameter to `None`.
ServerStatisticsLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ServerStatisticsLogMaxAge* parameter specifies the maximum duration that the server statistics log files are kept. Log files older than the specified value are deleted. The

			default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 60 days for this parameter, use 60.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of this parameter to 00:00:00 prevents the automatic removal of server statistics log files.
ServerStatisticsLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ServerStatisticsLogMaxDirectorySize* parameter specifies the cap on the size of the server statistics log directory. When the maximum directory size is reached, the server deletes the oldest log files first. The minimum value is 1 MB. The default value is 250 MB. When you enter a value, qualify the value with one

			of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *ServerStatisticsLogMaxFileSize* parameter must be less than or equal to the value of the *ServerStatisticsLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the server statistics log directory.
ServerStatisticsLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ServerStatisticsLogMaxFileSize* parameter specifies the maximum file size for the server statistics log files. When a log file reaches its maximum file size, a new log file is created. The default value is 10 MB.

			When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *ServerStatisticsLogMaxFileSize* parameter must be less than or equal to the value of the *ServerStatisticsLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the server statistics log files.
ServerStatisticsLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *ServerStatisticsLogPath* parameter specifies the location of server statistics log storage. The default location is `%ExchangeInstallPath%TransportRoles\Logs\Hub\ServerStats`.

			Setting the value of this parameter to `$null` disables server statistics logging.
SmtpEnableAllTlsVersions	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransientFailureRetryCount	Optional	System.Int32	The *TransientFailureRetryCount* parameter specifies the maximum number of immediate connection retries attempted when the server encounters a connection failure with a remote server. The default value is 6. The valid input range for this parameter is from 0 through 15. When the value of this parameter is set to 0, the server doesn't immediately attempt to retry an unsuccessful connection, and the next connection attempt is controlled by the *OutboundConnectionFailureRetryInterval* parameter.
TransientFailureRetryInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *TransientFailureRetryInterval* parameter controls the

			connection interval between each connection attempt specified by the *TransientFailureRetryCount* parameter. For the Transport service on a Mailbox server, the default value of the *TransientFailureRetryInterval* parameter is 5 minutes. On an Edge Transport server, the default value is 10 minutes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 8 minutes for this parameter, use 00:08:00. The valid input range for this parameter is from 00:00:01 through 12:00:00.
TransportMaintenanceLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportMaintenanceLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
TransportMaintenanc	Optional	Microsoft.Exchange.Da	This parameter is reserved

eLogMaxDirectorySize		ta.Unlimited	for internal Microsoft use.
TransportMaintenanceLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
TransportMaintenanceLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
TransportSyncAccountsPoisonAccountThreshold	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
TransportSyncAccountsPoisonDetectionEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncAccountsPoisonItemThreshold	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
TransportSyncAccountsSuccessivePoisonItemThreshold	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
TransportSyncEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncExchangeEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncFacebookEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncHubHealthLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncHubHealthLogFilePath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
TransportSyncHubHea	Optional	Microsoft.Exchange.Da	This parameter is reserved

lthLogMaxAge		ta.EnhancedTimeSpan	for internal Microsoft use.
TransportSyncHubHealthLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncHubHealthLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncImapEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncLinkedInEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncLogFilePath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
TransportSyncLogLoggingLevel	Optional	Microsoft.Exchange.Data.SyncLoggingLevel	This parameter is reserved for internal Microsoft use.
TransportSyncLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
TransportSyncLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncMaxDownloadItemsPerConnection	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
TransportSyncMaxDownloadSizePerConnec	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.

tion			
TransportSyncMaxDownloadSizePerItem	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncPopEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncRemoteConnectionTimeout	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
UseDowngradedExchangeServerAuth	Optional	System.Boolean	The *UseDowngradedExchangeServerAuth* parameter specifies whether the Generic Security Services application programming interface (GSSAPI) authentication method is used on connections where Transport Layer Security (TLS) is disabled. Normally, TLS is required for connections between the Transport services on Mailbox servers in your organization. On TLS secured connections, Kerberos authentication is used by default. However, there may be scenarios where you need to disable TLS between specific Transport services in your organization. When you

			do that, you need to set this parameter to `$true` to provide an alternative authentication method. The default value is `$false`. You shouldn't set this value to `$true` unless it's absolutely required. If you set this parameter to `$true`, you also need to create a specific Receive connector to service the non-TLS connections. This Receive connector must have remote IP address ranges specified to ensure that it's only used for non-TLS connections. You also must set the *SuppressXAnonymousTls* attribute of the Receive connector to `$true`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.
WindowsLiveHotmailTransportSyncEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
WlmLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
WlmLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
WlmLogMaxFileSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
WlmLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-X400AuthoritativeDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-X400AuthoritativeDomain** cmdlet to view the configuration information for the X.400 authoritative domains configured in your organization. For more information about how to

configure an X.400 authoritative domain, see Set-X400AuthoritativeDomain.

```
Get-X400AuthoritativeDomain [-Identity
<X400AuthoritativeDomainIdParameter>] [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example displays detailed information about the X.400 authoritative domain Europe Sales X.400 Domain.

```
Get-X400AuthoritativeDomain "Europe Sales X.400 Domain" |
Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "X.400 domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance

			of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.X400AuthoritativeDomainIdParameter	The *Identity* parameter specifies a string value for the X.400 authoritative domain. Enter either the GUID or the name of the remote domain.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-X400AuthoritativeDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-X400AuthoritativeDomain** cmdlet to create and specify the X.400 authoritative domain for the organization. The X.400 authoritative domain defines the standard fields for the namespace appended to the recipient identity for all mailboxes assigned an X.400 address.

```
New-X400AuthoritativeDomain -Name <String> -X400DomainName <X400Domain> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]] [-X400ExternalRelay <$true | $false>]
```

Examples

EXAMPLE 1

This example creates the X.400 authoritative domain Sales in the private domain Contoso, which is under the administrative domain Fabrikam.

```
New-X400AuthoritativeDomain -Name Sales -X400DomainName
"C=US;A=Fabrikam;P=Contoso;O=Sales"
```

EXAMPLE 2

This example creates an external relay domain for the X.400 namespace Europe organizational unit (OU) under the Sales organization in the private domain Contoso, which is under the administrative domain Fabrikam.

```
New-X400AuthoritativeDomain -Name "Sales Europe" -
X400DomainName
"C=US;A=Fabrikam;P=Contoso;O=Sales;OU1=Europe" -
X400ExternalRelay: $true
```

Detailed Description

X.400 domain names can only include the following ASCII characters:

- A to Z
- a to z
- 0–9
- These punctuation and special characters: (space) ' () + , - . / : = ?

You can use the following X.400 attributes (one each per address):

- **Name** **country** **Abbreviation** C **Maximum character length** 2
- **Name** **administrative domain** **Abbreviation** A **Maximum character length** 16
- **Name** **private domain** **Abbreviation** P **Maximum character length** 16
- **Name** **organization name** **Abbreviation** O **Maximum character length** 64
- **Name** **organizational unit name** **Abbreviation** Ou1-4 **Maximum character length** 32

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "X.400 domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies a unique name for an X.400 authoritative domain object. When you specify a name that includes spaces, you must enclose the name in quotation marks ("), for example, "Display Name". The *Name* parameter can't exceed 64 characters.
X400DomainName	Required	Microsoft.Exchange.Data.X400Domain	The *X400DomainName* parameter specifies the X.400 namespace, which can only include the X.400 organizational components. Specifically, only the following attribute types are supported: • **Label** (Abbreviation) • **C** (Country) • **A** (ADMD) • **P** (PRMD) • **O** (Organization) • **OU1** (Organization unit 1) • **OU2** (Organization unit 2) • **OU3** (Organization unit 3)

			• **OU4** (Organization unit 4) The address attributes must be separated by semicolons, and the address must be enclosed in quotation marks ("), for example, `"C=US;A=ATT;P=Contoso;O=Sales"`
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge

			Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
X400ExternalRelay	Optional	System.Boolean	The *X400ExternalRelay* parameter specifies this authoritative domain as an external relay domain. If you set the *X400ExternalRelay* parameter to `$true`, Microsoft Exchange routes email to the external address and doesn't treat resolution failures to this subdomain as an error.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-X400AuthoritativeDomain

Exchange Management Shell > Exchange 2013 cmdlets > Mail flow cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-X400AuthoritativeDomain** cmdlet to remove an X.400 authoritative domain. When you remove an X.400 authoritative domain, the X.400 authoritative domain object is deleted from Active Directory.

```
Remove-X400AuthoritativeDomain -Identity
<X400AuthoritativeDomainIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the X.400 authoritative domain object for the X.400 authoritative domain Sales.

```
Remove-X400AuthoritativeDomain -Identity Sales
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "X.400 domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.X400AuthoritativeDomainIdParameter	The *Identity* parameter specifies a string value for the X.400 authoritative domain. Enter either the GUID or the name of the X.400 authoritative domain.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't

			supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-X400AuthoritativeDomain

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-X400AuthoritativeDomain** cmdlet to edit an existing X.400 authoritative domain for your organization. The X.400 authoritative domain defines the standard fields for the namespace appended to the recipient identity for all mailboxes assigned an X.400 address.

```
Set-X400AuthoritativeDomain -Identity <X400AuthoritativeDomainIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Name <String>]
[-WhatIf [<SwitchParameter>]] [-X400DomainName <X400Domain>] [-
X400ExternalRelay <$true | $false>]
```

Examples

EXAMPLE 1

This example makes the following changes to an existing X.400 authoritative domain:

- It changes the domain name from Sales to Sales and Marketing.
- It updates the organizational attribute to **Sales and Marketing**.

```
Set-X400AuthoritativeDomain Sales -X400DomainName
"C=US;A=att,P=Contoso;O=Sales and Marketing" -Name "Sales
and Marketing"
```

Detailed Description

X.400 domain names can include only the following ASCII characters:

- A to Z
- a to z
- 0-9
- These punctuation marks and special characters: (space) ' () + , - . / : = ?

You can use the following X.400 attributes (1 each per address):

- **Name** **country** **Abbreviation** C **Maximum character length** 2
- **Name** **administrative domain** **Abbreviation** A **Maximum character length** 16
- **Name** **private domain** **Abbreviation** P **Maximum character length** 16
- **Name** **organization name** **Abbreviation** O **Maximum character length** 64
- **Name** **organizational unit name** **Abbreviation** Ou1-4 **Maximum character length** 32

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "X.400

domains" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.X400AuthoritativeDomainIdParameter	The *Identity* parameter specifies the display name of the X.400 authoritative domain.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the

			local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Name	Optional	System.String	The *Name* parameter specifies a unique name for an X.400 authoritative domain object. When you specify a name that includes spaces, you must enclose the name in quotation marks ("), for example, "Display Name". The *Name* parameter must contain a maximum of 64 characters.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
X400DomainName	Optional	Microsoft.Exchange.Data.X400Domain	The *X400DomainName* parameter specifies the X.400 namespace that can

			only include the X.400 organizational components. Specifically, only the following attribute types are supported: • **Label** (Abbreviation) • **C** (Country) • **A** (ADMD) • **P** (PRMD) • **O** (Organization) • **OU1** (Organization unit 1) • **OU2** (Organization unit 2) • **OU3** (Organization unit 3) • **OU4** (Organization unit 4) The address attributes must be separated by semicolons and the address must be enclosed in quotation marks ("), for example, `"C=US;A=att;P=Contoso;O=Sales"`.
X400ExternalRelay	Optional	System.Boolean	The *X400ExternalRelay* parameter specifies whether this authoritative domain is an external relay domain. If you set the *X400ExternalRelay*

			parameter to `$true`, Exchange routes e-mail to the external address and doesn't treat resolution failures to this subdomain as errors.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Mailbox cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-14*

Mailbox cmdlets

Connect-Mailbox

Disable-Mailbox

Enable-Mailbox

Get-Mailbox

New-Mailbox

Remove-Mailbox

Search-Mailbox

Set-Mailbox

Mailbox configuration cmdlets

Get-MailboxAutoReplyConfiguration

Set-MailboxAutoReplyConfiguration

Get-MailboxFolder

New-MailboxFolder

New-MailMessage

Get-MessageCategory

Disable-ServiceEmailChannel

Enable-ServiceEmailChannel

Mailbox import and export cmdlets

Note:

These cmdlets are available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use these cmdlets, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

Get-MailboxExportRequest

New-MailboxExportRequest

Remove-MailboxExportRequest

Resume-MailboxExportRequest

Set-MailboxExportRequest

Suspend-MailboxExportRequest

Get-MailboxExportRequestStatistics

Get-MailboxImportRequest

New-MailboxImportRequest

Remove-MailboxImportRequest

Resume-MailboxImportRequest

Set-MailboxImportRequest

Suspend-MailboxImportRequest

Get-MailboxImportRequestStatistics

Mailbox permission cmdlets

Add-MailboxFolderPermission

Get-MailboxFolderPermission

Remove-MailboxFolderPermission

Set-MailboxFolderPermission

Add-MailboxPermission

Get-MailboxPermission

Remove-MailboxPermission

Mailbox reporting cmdlets

Export-MailboxDiagnosticLogs

Get-MailboxFolderStatistics

Get-MailboxStatistics

Mailbox restore cmdlets

Get-MailboxRestoreRequest

New-MailboxRestoreRequest

Remove-MailboxRestoreRequest

Resume-MailboxRestoreRequest

Set-MailboxRestoreRequest

Suspend-MailboxRestoreRequest

Get-MailboxRestoreRequestStatistics

Apps for Outlook cmdlets

Disable-App

Enable-App

Get-App

New-App

Remove-App

Set-App

Calendar cmdlets

Get-CalendarDiagnosticAnalysis

Get-CalendarDiagnosticLog

Get-CalendarNotification

Set-CalendarNotification

Get-CalendarProcessing

Set-CalendarProcessing

Get-MailboxCalendarFolder

Set-MailboxCalendarFolder

Get-ResourceConfig

Set-ResourceConfig

Inbox rule cmdlets

Disable-InboxRule

Enable-InboxRule

Get-InboxRule

New-InboxRule

Remove-InboxRule

Set-InboxRule

User photo cmdlets

Export-RecipientDataProperty

Import-RecipientDataProperty

Get-UserPhoto

Remove-UserPhoto

Set-UserPhoto

Disable-App

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-App** cmdlet to disable (turn off) a specific app for a specific user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-App -Identity <AppIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the Bing Maps app for user Tony.

```
Disable-App -Identity 7a774f0c-7a6f-11e0-85ad-07fb4824019b
-Mailbox Tony
```

For more information, see Manage user access to apps for Outlook.

EXAMPLE 2

This example disables the administrator-installed app FinanceTestApp for user Tony.

```
Disable-App -Identity <GUID for FinanceTestApp> -Mailbox
Tony
```

For more information, see Manage user access to apps for Outlook.

Detailed Description

The **Disable-App** cmdlet requires that the specified app has already been installed (for example, that the app has been installed with the **New-App** cmdlet, or that it's a default app for Microsoft Outlook).

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Apps for Outlook" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AppIdParameter	The *Identity* parameter specifies the GUID of the app.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the mailbox or mail user. You can use the following values:

			• GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias You can't use this parameter in conjunction with the *Identity* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-App

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-App** cmdlet to enable (turn on) a specific app for a specific user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-App -Identity <AppIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the default Bing Maps app installed for user Tony.

```
Enable-App -Identity 7a774f0c-7a6f-11e0-85ad-07fb4824019b -
Mailbox Tony
```

For more information, see Manage user access to apps for Outlook.

EXAMPLE 2

This example enables the administrator-installed app FinanceTestApp for user Tony.

```
Enable-App -Identity <GUID for FinanceTestApp> -Mailbox
Tony
```

For more information, see Manage user access to apps for Outlook.

Detailed Description

The **Enable-App** cmdlet requires that the specified app has already been installed (for example, that it has been installed with the **New-App** cmdlet, or that it's a default app for Microsoft Outlook).

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Apps for Outlook" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AppIdParameter	The *Identity* parameter specifies the GUID of the app.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Mailbox	Optional	Microsoft.Exchange.Co	The *Mailbox* parameter

		nfiguration.Tasks.MailboxIdParameter	specifies the identity of the mailbox or mail user. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-App

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-App** cmdlet to return information about the installed app.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-App [-Identity <AppIdParameter>] [-DomainController <Fqdn>] [-Mailbox
<MailboxIdParameter>] [-Organization <OrganizationIdParameter>] [-
OrganizationApp <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns the list of apps installed for user Tony. The Exchange Management Shell returns the name of the app, whether the app is enabled, and the app version number.

```
Get-App -Mailbox Tony
```

EXAMPLE 2

This example displays the version of the Bing Maps app for the currently logged on user.

```
Get-App -Identity 7a774f0c-7a6f-11e0-85ad-07fb4824019b
```

EXAMPLE 3

This example displays the apps installed by administrators for the entire organization.

```
Get-App -OrganizationApp $true
```

For information about installing or removing apps for Outlook, see Install or remove apps for Outlook for your organization.

Detailed Description

The **Get-App** cmdlet returns information about all installed apps or the details of a specific installed app.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Apps for Outlook" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.AppIdParameter	The *Identity* parameter specifies the GUID of the app.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the mailbox or mail user. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name

			(UPN) • Legacy Exchange DN • SMTP address • Alias You can't use this parameter in conjunction with the *Identity* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
OrganizationApp	Optional	System.Management.Automation.SwitchParameter	The *OrganizationApp* parameter specifies the apps installed for the organization (not bound to a specific user). This is set to `$false` by default.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-App

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-App** cmdlet to install apps for Outlook.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-App [-FileData <Byte[]>] <COMMON PARAMETERS>


New-App [-Etoken <String>] [-MarketplaceAssetID <String>] [-
MarketplaceQueryMarket <String>] [-MarketplaceServicesUrl <String>]
<COMMON PARAMETERS>


New-App [-FileStream <Stream>] <COMMON PARAMETERS>


New-App [-Url <Uri>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DefaultStateForUser
<Enabled | Disabled | AlwaysEnabled>] [-DomainController <Fqdn>] [-
DownloadOnly <SwitchParameter>] [-Enabled <$true | $false>] [-Mailbox
<MailboxIdParameter>] [-OrganizationApp <SwitchParameter>] [-ProvidedTo
<Everyone | SpecificUsers>] [-UserList <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example installs the Finance Test app manifest file that has been copied to the local hard disk.

```
$Data=Get-Content -Path "C:\Apps\FinanceTestApp.xml" -
Encoding Byte -ReadCount 0

New-App -FileData $Data
```

For more information, see Install or remove apps for Outlook for your organization.

EXAMPLE 2

This example installs the Contoso CRM app manifest.xml from a URL on the Contoso corporate network. The Exchange server must be able to reach the target URL. This app is installed as an organization app and made available to a specific set of users in the organization, and is enabled for those users by default.

```
New-App -OrganizationApp -Url https://Server01.Contoso.com/
apps/ContosoCRMApp/manifest.xml -ProvidedTo SpecificUsers -
UserList "user1,user2,user3,user4,user5" -
DefaultStateForUser Enabled
```

For more information, see Install or remove apps for Outlook for your organization.

Detailed Description

If the app is enabled for the entire organization, users can activate the new app when viewing mail or calendar items within Microsoft Outlook or Microsoft Office Outlook Web App. If an installed app isn't enabled, users can enable the app from Outlook Web App Options. Similarly, if an app is installed, an administrator can enable the app from the Exchange Administration Center or by using the **Enable-App** or **Set-App** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Apps for Outlook" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DefaultStateForUser	Optional	Microsoft.Exchange.Management.Extension.DefaultStateForUser	The *DefaultStateForUser* parameter specifies the default initial state of the organization app for the provided users. This parameter is set to `Disabled` by default. If set to `AlwaysEnabled`, users

			can't disable the app for themselves. You must use the *OrganizationApp* parameter when you use this parameter. The possible values are: • `Enabled` • `Disabled` • `AlwaysEnabled`
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
DownloadOnly	Optional	System.Management.Automation.SwitchParameter	The *DownloadOnly* switch specifies whether to get the app manifest file and prompt the user for confirmation before committing to actual installation. This is set to `$false` by default. If the *DownloadOnly* parameter is set to `$true`, the cmdlet only downloads the app manifest file (and displays the app properties)

			without installing the app.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the app is made available to users in the organization. By default, this is set to `$true`. **Caution:** This setting overrides the *ProvidedTo*, *UserList*, and *DefaultStateForUser* settings.This setting doesn't prevent users from installing their own instance of the app if the user has install permissions. This parameter may be set to the following values: • `$true` The app is enabled for the specified users in the organization. This makes the app available for the specified users. • `$false` The app isn't enabled for any users in the organization. This hides the app from all users in the organization.
Etoken	Optional	System.String	This parameter is reserved for internal Microsoft use.
FileData	Optional	System.Byte[]	The *FileData* parameter specifies the location of the app manifest file. You must specify only one

			source location for the app manifest file. You can specify the app manifest file by using the *MarketplaceServicesUrl*, *Url*, or *FileData* parameter. If you use this parameter, use the **Get-Content** cmdlet and this cmdlet together as shown in the example.
FileStream	Optional	System.IO.Stream	The *FileStream* parameter can't be used in the Exchange Management Shell. It's used to support the app uploader and is only used by the Exchange admin center.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the mailbox or mail user. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias

MarketplaceAssetID	Optional	System.String	The *MarketplaceAssetID* parameter specifies the office store identifier for the app. This parameter is required if the *MarketplaceServicesUrl* parameter is specified.
MarketplaceQueryMarket	Optional	System.String	The *MarketplaceQueryMarket* parameter specifies the locale that an app is filed under at the office marketplace. For example, an app for the United States market in English uses the value `en-us`. If not specified, this value is set to `en-us`.
MarketplaceServicesUrl	Optional	System.String	The *MarketplaceServicesUrl* parameter specifies the full services URL for the app. You must specify only one source location for the app manifest file. You can specify the app manifest file by using the *MarketplaceServicesUrl*, *Url*, or *FileData* parameter.
OrganizationApp	Optional	System.Management.	The *OrganizationApp*

		Automation.SwitchParameter	parameter specifies the apps that are installed for the organization (not bound to a specific user). This is set to `$false` by default.
ProvidedTo	Optional	Microsoft.Exchange.Data.ApplicationLogic.Extension.ClientExtensionProvidedTo	The *ProvidedTo* parameter specifies the availability of an app in your organization. By default, new apps are available to all users in your organization. You must use the *OrganizationApp* parameter when you use this parameter. The following are the possible values: • `Everyone` This app is provided to every user in the organization. Every user sees this app listed in the **installed apps** list in Outlook Web App Options. When apps in the installed apps list display as enabled, users can use the features of this app in their email. All users are blocked from installing their own instances of this app, including but not limited to users with install apps permissions.

			• `SpecificUsers` This app is provided to only the users specified using the *UserList* parameter. Users that aren't specified don't see this organizational app in their management view, nor will it activate in their mail or calendar items. Specified users are also blocked from installing their own instance of this app. Users that aren't listed aren't blocked from installing their own instance of this app.
Url	Optional	System.Uri	The *Url* parameter specifies the full URL location of the app manifest file you want to install. You must specify only one source location for the app manifest file. You can specify the app manifest file by using the *MarketplaceServicesUrl*, *Url*, or *FileData* parameter.
UserList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UserList* parameter specifies the list of users that are granted access to the organizational app. Use the *UserList* parameter to specify the users that you want to use

			the app.You must use the *OrganizationApp* parameter when you use this parameter. You can use the following values: • GUID • DN • *Domain\account* • UPN • Legacy Exchange DN • SMTP address • Alias
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-App

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-App** cmdlet to uninstall an app.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-App -Identity <AppIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>] [-OrganizationApp
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the Finance Test app installed for user Tony.

```
Remove-App -Identity <GUID for FinanceTestApp> -Mailbox
Tony
```

For more information, see Install or remove apps for Outlook for your organization.

Detailed Description

The **Remove-App** cmdlet requires that the specified app has already been installed (for example, that the app has been installed with the **New-App** cmdlet. Apps installed by default can't be uninstalled, but they can be disabled.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Apps for Outlook" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.AppIdParameter	specifies the GUID of the app.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$false`. You must include a colon (:) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the mailbox or mail user. You can use the following values: • GUID • Distinguished name (DN)

			• *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias
OrganizationApp	Optional	System.Management.Automation.SwitchParameter	The *OrganizationApp* parameter specifies that the scope of the app is organizational. This is set to `$false` by default. This parameter is required if the targeted app is installed for the organization.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-App

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-App** cmdlet to set configuration properties on an app object.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-App -Identity <AppIdParameter> [-Confirm [<SwitchParameter>]] [-
DefaultStateForUser <Enabled | Disabled | AlwaysEnabled>] [-
DomainController <Fqdn>] [-Enabled <$true | $false>] [-Organization
<OrganizationIdParameter>] [-OrganizationApp <SwitchParameter>] [-
ProvidedTo <Everyone | SpecificUsers>] [-UserList <MultiValuedProperty>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the organization app FinanceTestApp, which was installed to everyone in the organization, to be provided to specific users on the finance team and to be enabled by default.

```
$a= Get-DistributionGroupMember FinanceTeam

Set-App -OrganizationApp -Identity 7a774f0c-7a6f-11e0-85ad-
07fb4824019b -ProvidedTo SpecificUsers -UserList $a -
DefaultStateForUser Enabled
```

For more information, see Manage user access to apps for Outlook.

EXAMPLE 2

This example disables the organization app FinanceTestApp across the organization and hides it from end user view.

```
Set-App -OrganizationApp -Identity 7a774f0c-7a6f-11e0-85ad-07fb4824019b -Enabled:$false
```

For more information, see Manage user access to apps for Outlook.

Detailed Description

The **Set-App** cmdlet can only be used when configuring the availability of an organization app. This task requires that the specified app has already been installed (for example, that the app has been installed with the **New-App** cmdlet, or that it's a default app for Microsoft Outlook).

Default apps in Microsoft Office Outlook Web App and apps that you've installed for use by users in your organization are known as organization apps. Organization apps can't be removed by end users, but can be enabled or disabled. If an app is an organization app (scope default or organization), the delete control on the toolbar is disabled for end users. Administrators are able to remove organization apps. Administrators can't remove default apps, but they can disable them for the entire organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Apps for Outlook" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AppIdParameter	The *Identity* parameter specifies the GUID of the app.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DefaultStateForUser	Optional	Microsoft.Exchange.Management.Extension.DefaultStateForUser	The *DefaultStateForUser* parameter specifies the default initial state of the organization app for the provided users. This parameter is set to `Disabled` by default. If set to `AlwaysEnabled`, users can't disable the app for themselves. You must use the *OrganizationApp* parameter when you use this parameter. The possible values are: • `Enabled` • `Disabled` • `AlwaysEnabled`
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the app is made available to users in the organization. By default, this is set to `$true`. **Caution:**

			This setting overrides the *ProvidedTo*, *UserList*, and *DefaultStateForUser* settings. This setting doesn't prevent users from installing their own instance of the app if the users have install permissions. The following are the possible values: • `$true` App is enabled for the specified users in the organization. This makes the app available for the specified users. • `$false` App isn't enabled for any users in the organization. This hides the app from user view for all users in the organization.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
OrganizationApp	Optional	System.Management.Automation.SwitchParameter	The *OrganizationApp* parameter specifies that the scope of the app is organizational. This is set to `$false` by default.
ProvidedTo	Optional	Microsoft.Exchange.Data.ApplicationLogic.Extension.ClientExtensionProvidedTo	The *ProvidedTo* parameter specifies the availability of an app in your organization. By default, new apps are available to all users in your organization. You must use the *OrganizationApp*

			parameter when you use this parameter. The following are the possible values: • `Everyone` This app is provided to every user in the organization. Every user sees this app listed in the installed apps list in Outlook Web App Options. When apps in the installed apps list display as enabled, the users are able to use the features of this app in their email. All users are blocked from installing their own instances of this app, including but not limited to users with install apps permissions. • `Specificusers` This app is provided to only the users specified using the *UserList* parameter. Users that aren't specified don't see this organizational app in their management view, nor does it activate in their mail or calendar items. Specified users are also blocked from installing their own instance of this app. Users that aren't listed aren't blocked from installing their own instance of this app.
UserList	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *UserList* parameter specifies the list of users

		y	granted access to the organizational app. Use the *UserList* parameter to specify the users that you want to use the app.You must use the *OrganizationApp* parameter when you use this parameter. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP Address • Alias
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-CalendarDiagnosticAnalysis

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-22*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-CalendarDiagnosticAnalysis** cmdlet to troubleshoot calendar-related reliability issues. You can use this cmdlet to analyze calendar log data captured in your calendar diagnostic log files.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-CalendarDiagnosticAnalysis -CalendarLogs <CalendarLog[]> <COMMON
PARAMETERS>

Get-CalendarDiagnosticAnalysis -LogLocation <String[]> <COMMON PARAMETERS>

COMMON PARAMETERS: [-DetailLevel <Basic | Advanced>] [-GlobalObjectId
<String>] [-OutputAs <HTML | CSV | XML>]
```

Examples

EXAMPLE 1

This example reads the logs for a user with alias Tony into memory, analyzes the logs, and produces the output in a CSV file.

This command collects the logs for a meeting with a specified MeetingID:

```
$logs = Get-CalendarDiagnosticLog -Identity Tony -
MeetingID
040000008200E00074C5B7101A82E008000000009421DCCD5046CD01000
0000000000000010000000010B0349F6B17454685E17D9F9512E71F
```

This command returns a detailed analysis in a CSV file.

```
Get-CalendarDiagnosticAnalysis -CalendarLogs $logs -
DetailLevel Advanced > analysis.csv
```

This command returns a basic analysis in the Exchange Management Shell output.

```
Get-CalendarDiagnosticAnalysis -CalendarLogs $logs
```

Detailed Description

You run the **Get-CalendarDiagnosticAnalysis** cmdlet to analyze calendar data you've retrieved using the **Get-CalendarDiagnosticLog** cmdlet. For more information, see Get-CalendarDiagnosticLog.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Calendar diagnostics" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
CalendarLogs	Required	Microsoft.Exchange.Management.StoreTasks.CalendarLog[]	The *CalendarLogs* parameter specifies the Message ID of the calendar item you want to analyze. You can specify the logs you want to analyze by using the *CalendarLogs* parameter or the *LogLocation* parameter, but not both.
LogLocation	Required	System.String[]	The *LogLocation* parameter specifies the location of the calendar log files you want to

			analyze. You can specify the logs you want to analyze by using the *CalendarLogs* parameter or the *LogLocation* parameter, but not both. If you use this parameter and the logs you want to analyze are located in the C:\logs directory on the computer you're running the cmdlet on, use `"C:\logs"` as the parameter value. If the logs you want to analyze are located on a share, use the format `"\\ServerName\ShareFolder"` where *ServerName**ShareFolder* is the location of the log files on your network.
DetailLevel	Optional	Microsoft.Exchange.Management.StoreTasks.AnalysisDetailLevel	The *DetailLevel* parameter specifies the level of detail you want to see in the analysis output. The default value is `Basic` and returns a summary of the analysis. Specify `Advanced` to see a

			detailed report which includes all of the properties from the calendar logs. `Advanced` should be used only when detailed debugging is necessary.
GlobalObjectId	Optional	System.String	The *GlobalObjectId* parameter specifies the Global ID of the calendar item you want to analyze.
OutputAs	Optional	Microsoft.Exchange.Management.StoreTasks.OutputType	The *OutputAs* parameter specifies the file format you want to output into the Logging directory. The default value is `HTML`. The other output options are csv, and `XML`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-CalendarDiagnosticLog

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-CalendarDiagnosticLog** cmdlet to collect a range of calendar logs. The Calendar Diagnostic logs track important calendar-related event data for each mailbox and can be used to troubleshoot calendar issues that occur in mailboxes. The logs track all calendar items and meeting messages.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-CalendarDiagnosticLog -Identity <MailboxIdParameter> <COMMON
PARAMETERS>


Get-CalendarDiagnosticLog -Identity <MailboxIdParameter> -LogLocation
<String> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Credential <PSCredential>] [-DomainController <Fqdn>]
[-EndDate <ExDateTime>] [-Identity <MailboxIdParameter>] [-Latest
<SwitchParameter>] [-MeetingID <String>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>] [-StartDate <ExDateTime>] [-
Subject <String>]
```

Examples

EXAMPLE 1

This example retrieves the Calendar Diagnostic logs for Tony Smith's mailbox by using the subject Weekly development meeting.

```
Get-CalendarDiagnosticLog -Identity Tony -Subject "Weekly development meeting"
```

EXAMPLE 2

This example retrieves the Calendar Diagnostic logs for Tony Smith's mailbox from 6/1/2012 to 6/30/2012.

```
Get-CalendarDiagnosticLog -Identity Tony -StartDate "6/1/2012 6:00:00 AM" -EndDate "6/30/2012 5:00:00 PM"
```

EXAMPLE 3

This example retrieves the Calendar Diagnostic log data only for the most recent calendar item in Tony Smith's mailbox with a message subject of "Weekly development meeting".

```
Get-CalendarDiagnosticLog -Identity Tony -Subject "Weekly development meeting" -Latest
```

Detailed Description

After you run the **Get-CalendarDiagnosticLog** cmdlet, you can analyze the calendar data using the **Get-CalendarDiagnosticAnalysis** cmdlet. For more information, see Get-CalendarDiagnosticAnalysis.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Calendar diagnostics" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox user's name. This is the name that appears in Active Directory Users and Computers. This is also the user name that appears in **Recipient Properties** on the **User Information** tab. You can use the following values: • Alias • Display name • *Domain\Account* • SMTP address • Distinguished name

			(DN) • Object GUID • User principal name (UPN) • LegacyExchangeDN
LogLocation	Required	System.String	The *LogLocation* parameter specifies the location of the log files. The log files are located in the Exchange Logging directory.
Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the user name and password to use to access Active Directory Domain Services (AD DS). This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
EndDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *EndDate* parameter specifies the end date of the date range. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
Latest	Optional	System.Management.Automation.SwitchParameter	The *Latest* switch specifies whether to return calendar log data for only the most recent calendar

			item.
MeetingID	Optional	System.String	This parameter is reserved for internal Microsoft use.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the calendar diagnostic information is read from a domain controller in the user's domain. If you use this parameter, multiple reads might be necessary to get the information.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
StartDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *StartDate* parameter specifies the start date of the date range. Use the short date format

			defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
Subject	Optional	System.String	The *Subject* parameter specifies the subject of the calendar item or meeting request. You can't use this parameter in conjunction with the *MeetingID* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-CalendarNotification

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-CalendarNotification** cmdlet to return a list of all calendar notification settings for a user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-CalendarNotification -Identity <MailboxIdParameter> [-Credential
<PSCredential>] [-DomainController <Fqdn>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns the calendar notification settings for the user Tony Smith using the user's alias.

```
Get-CalendarNotification -Identity "TonySmith"
```

EXAMPLE 2

This example returns the calendar notification settings for the user Tony Smith.

```
Get-CalendarNotification -Identity tony@contoso.com -
ReadFromDomainController
```

EXAMPLE 3

This example returns the calendar notification settings for the user Tony Smith using the user's domain and name.

```
Get-CalendarNotification -Identity "contoso\tonysmith"
```

Detailed Description

The **Get-CalendarNotification** cmdlet retrieves and displays the rules used to trigger the calendar agenda notification, reminder notification, or update notification.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Text messaging settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox ID for the user's mailbox.
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user credentials used to run the command. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that retrieves data from Active Directory.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	The *ReadFromDomainController* parameter specifies whether the command should return data from the domain controller.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the amount of data returned.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-CalendarNotification

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Set-CalendarNotification** cmdlet to set text message notifications for calendar events for a user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-CalendarNotification -Identity <MailboxIdParameter> [-
CalendarUpdateNotification <$true | $false>] [-
CalendarUpdateSendDuringWorkHour <$true | $false>] [-Confirm
[<SwitchParameter>]] [-DailyAgendaNotification <$true | $false>] [-
DailyAgendaNotificationSendTime <TimeSpan>] [-DomainController <Fqdn>] [-
IgnoreDefaultScope <SwitchParameter>] [-MeetingReminderNotification <$true
| $false>] [-MeetingReminderSendDuringWorkHour <$true | $false>] [-
NextDays <Int32>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables calendar updates to be sent in text messages to the user Tony Smith.

```
Set-CalendarNotification -Identity "tony@contoso.com" -
CalendarUpdateNotification $true
```

EXAMPLE 2

This example enables calendar updates and meeting reminders to be sent in text messages to the user Tony Smith.

```
Set-CalendarNotification -Identity "TonySmith" -
CalendarUpdateNotification $true -
MeetingReminderNotification $true -
MeetingReminderSendDuringWorkHour $true
```

EXAMPLE 3

This example enables a daily agenda to be sent in text messages to the user Tony Smith.

```
Set-CalendarNotification -Identity contoso\tonysmith -
DailyAgendaNotification $true
```

Detailed Description

Users can receive text message notifications of changes to calendar events and daily agendas. Use the **Set-CalendarNotification** cmdlet to configure these notifications for a user.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Text messaging settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox ID for the user.
CalendarUpdateNotification	Optional	System.Boolean	The *CalendarUpdateNotification* parameter specifies whether calendar notifications are enabled for the user.
CalendarUpdateSendDuringWorkHour	Optional	System.Boolean	The *CalendarUpdateSendDuringWorkHour* parameter specifies whether calendar notifications are sent during working hours.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value

			with the *Confirm* switch.
DailyAgendaNotification	Optional	System.Boolean	The *DailyAgendaNotification* parameter specifies whether a daily agenda should be sent to the user's mobile phone.
DailyAgendaNotificationSendTime	Optional	System.TimeSpan	The *DailyAgendaNotificationSendTime* parameter specifies the time to send the daily agenda. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, a time span of 2 days and 8 hours is shown: 02.08:00:00.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter isn't implemented yet.
MeetingReminderNotification	Optional	System.Boolean	The *MeetingReminderNotification* parameter specifies whether meeting reminders are sent to the user's mobile phone.
MeetingReminderSendDuringWorkHour	Optional	System.Boolean	The *MeetingReminderSendDuringWorkHour* parameter specifies whether meeting reminders are only sent during working hours.
NextDays	Optional	System.Int32	The *NextDays* parameter specifies how many days should be sent in the daily agenda.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-CalendarProcessing

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-06-10*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-CalendarProcessing** cmdlet to view the calendar processing options for resource mailboxes, which include the Calendar Attendant, resource booking assistant, and calendar configuration. Note that the settings returned by this cmdlet are editable only on resource mailboxes.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-CalendarProcessing -Identity <MailboxIdParameter> [-DomainController
```

```
<Fqdn>] [-ReadFromDomainController <SwitchParameter>] [-ResultSize
<Unlimited>]
```

Examples

EXAMPLE 1

This example shows the calendar processing options for the resource mailbox Room 212.

```
Get-CalendarProcessing -Identity "Room 212" | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Calendar processing" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the resource mailbox. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account`

			Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainControl*

			ler switch specifies that information should be read from a domain controller in the user's domain. If you run the command `Set-AdServerSettings -ViewEntireForest $true` to include all objects in the forest and you don't use the *ReadFromDomainControl ler* switch, it's possible that information will be read from a global catalog that has outdated information. When you use the *ReadFromDomainControl ler* switch, multiple reads might be necessary to get the information. You don't have to specify a value with this switch. **Note:** By default, the recipient scope is set to the domain that hosts your Exchange servers.
ResultSize	Optional	Microsoft.Exchange.Da ta.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that

			match the query, use `unlimited` for the value of this parameter. The default value is `1000`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-CalendarProcessing

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-09-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-CalendarProcessing** cmdlet to modify calendar processing options for resource mailboxes, which include the Calendar Attendant, resource booking assistant, and calendar configuration. Note that this cmdlet is effective only on resource mailboxes.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-CalendarProcessing -Identity <MailboxIdParameter> [-
AddAdditionalResponse <$true | $false>] [-AdditionalResponse <String>] [-
AddNewRequestsTentatively <$true | $false>] [-AddOrganizerToSubject <$true
| $false>] [-AllBookInPolicy <$true | $false>] [-AllowConflicts <$true |
$false>] [-AllowRecurringMeetings <$true | $false>] [-AllRequestInPolicy
<$true | $false>] [-AllRequestOutOfPolicy <$true | $false>] [-
AutomateProcessing <None | AutoUpdate | AutoAccept>] [-BookingWindowInDays
<Int32>] [-BookInPolicy <RecipientIdParameter[]>] [-Confirm
[<SwitchParameter>]] [-ConflictPercentageAllowed <Int32>] [-
DeleteAttachments <$true | $false>] [-DeleteComments <$true | $false>] [-
DeleteNonCalendarItems <$true | $false>] [-DeleteSubject <$true | $false>]
[-DomainController <Fqdn>] [-EnableResponseDetails <$true | $false>] [-
EnforceSchedulingHorizon <$true | $false>] [-ForwardRequestsToDelegates
<$true | $false>] [-IgnoreDefaultScope <SwitchParameter>] [-
MaximumConflictInstances <Int32>] [-MaximumDurationInMinutes <Int32>] [-
OrganizerInfo <$true | $false>] [-ProcessExternalMeetingMessages <$true |
$false>] [-RemoveForwardedMeetingNotifications <$true | $false>] [-
```

```
RemoveOldMeetingMessages <$true | $false>] [-RemovePrivateProperty <$true
| $false>] [-RequestInPolicy <RecipientIdParameter[]>] [-
RequestOutOfPolicy <RecipientIdParameter[]>] [-ResourceDelegates
<RecipientIdParameter[]>] [-ScheduleOnlyDuringWorkHours <$true | $false>]
[-TentativePendingApproval <$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example automates the processing of calendar requests to the resource mailbox Conf 212.

```
Set-CalendarProcessing -Identity "Conf 212" -
AutomateProcessing AutoAccept -DeleteComments $true -
AddOrganizerToSubject $true -AllowConflicts $false
```

EXAMPLE 2

This example disables automatic processing for the resource mailbox Car 53.

```
Set-CalendarProcessing -Identity "Car 53" -
AutomateProcessing None
```

EXAMPLE 3

This example allows the Calendar Attendant to approve in-policy requests from all users.

```
Set-CalendarProcessing -Identity "5th Floor Conference
Room" -AutomateProcessing AutoAccept -AllBookInPolicy $true
```

EXAMPLE 4

This example allows all users to submit in-policy requests, but the request is still subject to approval by a delegate.

```
Set-CalendarProcessing -Identity "5th Floor Conference
Room" -AutomateProcessing AutoAccept -AllRequestInPolicy
$true
```

EXAMPLE 5

This example allows the Calendar Attendant to accept out-of-policy requests from David Pelton. The request is still subject to approval by a delegate.

```
Set-CalendarProcessing -Identity "Room 221" -
AutomateProcessing AutoAccept -RequestOutOfPolicy
DavidPelton@contoso.com
```

EXAMPLE 6

This example allows a list of users to submit in-policy meeting requests to the equipment mailbox for Car 53.

```
Set-CalendarProcessing -Identity "Car 53" -
AutomateProcessing AutoAccept -BookInPolicy
"ayla@contoso.com","tony@contoso.com"
```

EXAMPLE 7

This example rejects meeting requests from any user who isn't a member of the Exchange organization.

```
Set-CalendarProcessing -Identity "Room 221" -
ProcessExternalMeetingMessages $false
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Calendar processing" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the resource mailbox. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name`

			(DN) Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
AddAdditionalResponse	Optional	System.Boolean	The *AddAdditionalResponse* parameter specifies whether additional information would be sent from the resource mailbox when responding to meeting requests. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.

			This parameter is used only on resource mailboxes where the *AutomateProcessing* parameter is set to `AutoAccept`.
AdditionalResponse	Optional	System.String	The *AdditionalResponse* parameter specifies the additional information to be included in responses to meeting requests. This parameter is meaningful only when the *AddAdditionalRepsponse* parameter is set to `$true`.
AddNewRequestsTentatively	Optional	System.Boolean	The *AddNewRequestsTentatively* parameter specifies whether to have the Calendar Attendant put new calendar items tentatively on the calendar. If the *AddNewRequestsTentatively* parameter is set to $false, only existing calendar items are updated by the Calendar Attendant Valid input for this parameter is `$true` or `$false`. The default value

			is `$true`.
AddOrganizerToSubject	Optional	System.Boolean	The *AddOrganizerToSubject* parameter specifies whether the meeting organizer's name is used as the subject of the meeting request. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. This parameter is used only on resource mailboxes where the *AutomateProcessing* parameter is set to `AutoAccept`.
AllBookInPolicy	Optional	System.Boolean	The *AllBookInPolicy* parameter specifies whether to automatically approve in-policy requests from all users. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllowConflicts	Optional	System.Boolean	The *AllowConflicts* parameter specifies whether to allow conflicting meeting requests. Valid input for

			this parameter is `$true` or `$false`. The default value is `$false`.
AllowRecurringMeetings	Optional	System.Boolean	The *AllowRecurringMeetings* parameter specifies whether to allow recurring meetings. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
AllRequestInPolicy	Optional	System.Boolean	The *AllRequestInPolicy* parameter specifies whether to allow all users to submit in-policy requests. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. These requests are subject to approval by a resource mailbox delegate unless the *AllBookInPolicy* parameter is set to `$true`.
AllRequestOutOfPolicy	Optional	System.Boolean	The *AllRequestOutOfPolicy* parameter specifies whether to allow all users to submit out-of-policy requests. Valid input for this parameter is `$true` or

			`$false`. The default value is `$false`. Out-of-policy requests are subject to approval by a resource mailbox delegate.
AutomateProcessing	Optional	Microsoft.Exchange.Data.Storage.CalendarProcessingFlags	The *AutomateProcessing* parameter enables or disables calendar processing on the mailbox. This parameter takes the following values: • `None` Both the resource booking attendant and the Calendar Attendant are disabled on the mailbox. • `AutoUpdate` Only the Calendar Attendant processes meeting requests and responses. • `AutoAccept` Both the Calendar Attendant and resource booking attendant are enabled on the mailbox. This means that the Calendar Attendant updates the calendar, and then the resource booking assistant accepts the meeting based upon the policies. The default value on a resource mailbox is `AutoAccept`. The default

			value on a user mailbox is `AutoUpdate`, but you can't change the value on a user mailbox.
BookingWindowInDays	Optional	System.Int32	The *BookingWindowInDays* parameter specifies the maximum number of days in advance that the resource can be reserved. Valid input is an integer from 0 through 1080. The default value is 180 days. The value 0 means today.
BookInPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *BookInPolicy* parameter specifies a comma-separated list of users who are allowed to submit in-policy meeting requests to the resource mailbox. Any in-policy meeting requests from these users are automatically approved. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name`

			Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
ConflictPercentageAllowed	Optional	System.Int32	The *ConflictPercentageAllowed* parameter specifies the maximum percentage of meeting conflicts for new recurring meeting requests. Valid input for this parameter is an integer from 0 through 100. The default value is 0. If a new recurring meeting request conflicts with existing reservations for the resource more than the percentage specified by this parameter, the recurring meeting request is automatically declined. When the value is 0, no conflicts are permitted for new recurring meeting requests.
DeleteAttachments	Optional	System.Boolean	The *DeleteAttachments* parameter specifies whether to remove attachments from all incoming messages. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.

			This parameter is used only on resource mailboxes where the *AutomateProcessing* parameter is set to `AutoAccept`.
DeleteComments	Optional	System.Boolean	The *DeleteComments* parameter specifies whether to remove or keep any text in the message body of incoming meeting requests. Valid input for this parameter is `$true` or `$false`. This parameter is used only on resource mailboxes where the *AutomateProcessing* parameter is set to `AutoAccept`.
DeleteNonCalendarItems	Optional	System.Boolean	The *DeleteNonCalendarItems* parameter specifies whether to remove or keep all non-calendar items received by the resource mailbox. Valid input for this parameter is `$true` or `$false`. The

			default value is `$true`.
DeleteSubject	Optional	System.Boolean	The *DeleteSubject* parameter specifies whether to remove or keep the subject of incoming meeting requests. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. This parameter is used only on resource mailboxes where the *AutomateProcessing* parameter is set to `AutoAccept`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EnableResponseDetails	Optional	System.Boolean	The *EnableResponseDetails* parameter specifies whether to include the

			reasons for accepting or declining a meeting in the response email message. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, the reasons for accepting or declining a meeting in the response email message are included.
EnforceSchedulingHorizon	Optional	System.Boolean	The *EnforceSchedulingHorizon* parameter controls the behavior of recurring meetings that extend beyond the date specified by the *BookingWindowInDays* parameter. Valid values for this parameter are: • `$true` A recurring meeting request is automatically declined if the meetings start on or before the date specified by the *BookingWindowInDays* parameter, and the meetings extend beyond the specified date. • `$false` A recurring meeting request is

			automatically accepted if the meetings start on or before the date specified by the *BookingWindowInDays* parameter, and the meetings extend beyond the specified date. However, the number of meetings is automatically reduced so meetings won't occur after the specified date. The default value is `$true`.
ForwardRequestsToDelegates	Optional	System.Boolean	The *ForwardRequestsToDelegates* parameter specifies whether to forward incoming meeting requests to the delegates defined for the resource mailbox. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* switch tells the command to ignore the default recipient scope setting for the Exchange Management Shell session, and to use the entire forest as the scope. This allows the command

			to access Active Directory objects that aren't currently available in the default scope. Using the *IgnoreDefaultScope* switch introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
MaximumConflictInstances	Optional	System.Int32	The *MaximumConflictInstances* parameter specifies the maximum number of conflicts for new recurring meeting requests when the *AllowRecurringMeetings* parameter is set to `$true`. Valid input for this parameter is an integer from 0 through `INT32`

			(2147483647). The default value is 0. If a new recurring meeting request conflicts with existing reservations for the resource more than the number of times specified by the *MaximumConflictInstances* parameter value, the recurring meeting request is automatically declined. When the value is 0, no conflicts are permitted for new recurring meeting requests.
MaximumDurationInMinutes	Optional	System.Int32	The *MaximumDurationInMinutes* parameter specifies the maximum duration in minutes for meeting requests. Valid input for this parameter is an integer from 0 through `INT32` (2147483647). The default value is 1440 (24 hours). When the value is set to 0, the maximum duration of a meeting is unlimited. For recurring meetings, the value of this

			parameter applies to the length of an individual meeting instance.
OrganizerInfo	Optional	System.Boolean	The *OrganizerInfo* parameter specifies whether to have mailboxes send organizer information when a meeting request is declined because of conflicts. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, resource mailboxes send organizer information when a meeting request is declined because of conflicts.
ProcessExternalMeetingMessages	Optional	System.Boolean	The *ProcessExternalMeetingMessages* parameter specifies whether to process meeting requests that originate outside the Exchange organization. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.

			By default, meeting requests that originate outside of the organization are rejected.
RemoveForwardedMeetingNotifications	Optional	System.Boolean	The *RemoveForwardedMeetingNotifications* parameter specifies whether forwarded meeting notifications are moved to the Deleted Items folder after they're processed by the Calendar Attendant. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
RemoveOldMeetingMessages	Optional	System.Boolean	The *RemoveOldMeetingMessages* parameter specifies whether the Calendar Attendant removes old and redundant updates and responses. Valid input for this parameter is `$true` or `$false`. The default value is `$true`.
RemovePrivateProperty	Optional	System.Boolean	The *RemovePrivateProperty* parameter specifies

			whether to clear the private flag for incoming meeting requests. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. By default, the private flag for incoming meeting requests is cleared. To ensure the private flag that was sent by the organizer in the original request remains as specified, set tthis parameter to `$false`.
RequestInPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *RequestInPolicy* parameter specifies a comma-separated list of users who are allowed to submit in-policy meeting requests to the resource mailbox. All in-policy meeting requests from these users are subject to approval by a resource mailbox delegate. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com

			/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
RequestOutOfPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *RequestOutOfPolicy* parameter specifies a comma-separated list of users who are allowed to submit out-of-policy requests. Out-of-policy requests are subject to approval by a

			resource mailbox delegate. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com

ResourceDelegates	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ResourceDelegates* parameter specifies a comma-separated list of users who are resource mailbox delegates. Resource mailbox delegates can approve or reject requests sent to the resource mailbox. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/

			ou=AdministrativeGroup/ cn=Recipients/ cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
ScheduleOnlyDuring WorkHours	Optional	System.Boolean	The *ScheduleOnlyDuringWork Hours* parameter specifies whether to allow meetings to be scheduled outside of the working hours that are defined for the resource mailbox. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If set to `$true`, meeting requests for times outside the working hours of the resource mailbox will be rejected. You configure the working hours of the resource mailbox by using the *WorkDays*, *WorkingHoursStartTime*, *WorkingHoursEndTime* and *WorkingHoursTimeZone*

			parameters on the **Set-MailboxCalendarConfiguration** cmdlet.
TentativePendingApproval	Optional	System.Boolean	The *TentativePendingApproval* parameter specifies whether to mark pending requests as tentative on the calendar. Valid input for this parameter is `$true` or `$false`. The default value is `$true`. If set to `$false`, pending requests are marked as free.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-InboxRule

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-InboxRule** cmdlet to disable an Inbox rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-InboxRule -Identity <InboxRuleIdParameter> [-
AlwaysDeleteOutlookRulesBlob <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-Mailbox <MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the Inbox rule MoveAnnoncements in the mailbox Joe@Contoso.com.

```
Disable-InboxRule -Identity "MoveAnnouncements" -Mailbox
"Joe@Contoso.com"
```

Detailed Description

Important:
When you create, modify, remove, enable, or disable an Inbox rule on Microsoft Exchange Server 2013, any client-side rules created by Microsoft Outlook are removed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Inbox rules" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.InboxRuleIdParameter	The *Identity* parameter specifies the identity of the Inbox rule to be disabled.
AlwaysDeleteOutlookRulesBlob	Optional	System.Management.Automation.SwitchParameter	The *AlwaysDeleteOutlookRulesBlob* parameter suppresses a warning that end users or administrators get if they use Outlook Web App or Windows PowerShell to modify Inbox rules.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress the confirmation prompt produced if rules created by Outlook exist on the mailbox. When taking an action using Inbox rules on Exchange 2013, any client-side rules are removed.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the mailbox to which the Inbox rule belongs.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes

			would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-InboxRule

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-InboxRule** cmdlet to enable an Inbox rule. Inbox rules are used to process messages in the Inbox based on conditions specified and take actions such as moving a message to a specified folder or deleting a message.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-InboxRule -Identity <InboxRuleIdParameter> [-
AlwaysDeleteOutlookRulesBlob <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-Mailbox <MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the Inbox rule Move To Junk Mail for User 1. Values for parameters such as the *Mailbox* or *InboxRule* parameter that contain spaces must be enclosed in quotation marks (").

```
Enable-InboxRule "Move To Junk Mail" -Mailbox "User 1"
```

Detailed Description

Important:
When you create, modify, remove, enable, or disable an Inbox rule on Microsoft Exchange Server 2013, any client-side rules created by Microsoft Outlook are removed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Inbox rules" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.InboxRuleIdParameter	The *Identity* parameter specifies the display name or GUID of the Inbox rule.
AlwaysDeleteOutlookRulesBlob	Optional	System.Management.Automation.SwitchParameter	The *AlwaysDeleteOutlookRulesBlob* parameter suppresses a warning that end users or administrators get if they use Outlook Web App or Windows PowerShell to modify Inbox rules.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress the confirmation prompt produced if rules created by Outlook exist on the mailbox. When taking an action using Inbox rules on Exchange 2013, any client-side rules are removed.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the mailbox where the Inbox rule is located. You can use one of the

			following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-InboxRule

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-InboxRule** cmdlet to view Inbox rule properties. Inbox rules are used to process messages in the Inbox based on conditions specified and take actions such as moving a message to a specified folder or deleting a message.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-InboxRule [-Identity <InboxRuleIdParameter>] [-DescriptionTimeFormat
<String>] [-DescriptionTimeZone <ExTimeZoneValue>] [-DomainController
<Fqdn>] [-IncludeHidden <SwitchParameter>] [-Mailbox <MailboxIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves all Inbox rules for the mailbox Joe@Contoso.com.

```
Get-InboxRule -Mailbox Joe@Contoso.com
```

EXAMPLE 2

This example retrieves the Inbox rule `ReceivedLastYear` from the mailbox joe@contoso.com on which the *ReceivedBeforeDate* parameter was set when the rule was created. The *DescriptionTimeFormat* and *DescriptionTimeZone* parameters are used in this example to specify formatting of the time and the time zone used in the rule's **Description** property.

```
Get-InboxRule "ReceivedLastYear" -Mailbox joe@contoso.com -
DescriptionTimeFormat "mm/dd/yyyy" -DescriptionTimeZone
"Pacific Standard Time"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Inbox

rules" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
DescriptionTimeFormat	Optional	System.String	The *DescriptionTimeFormat* parameter specifies the format in which time values are returned in the rule description. You must use standard time value format settings, for example: *mm/dd/yyyy*, where *mm* is the 2-digit month, *dd* is the 2-digit day, and *yyyy* is the 4-digit year.
DescriptionTimeZone	Optional	Microsoft.Exchange.Data.Storage.Management.ExTimeZoneValue	The *DescriptionTimeZone* parameter specifies the format in which the time zone for time values is returned in the rule description. Use a valid Microsoft Windows time zone name. You can use the Windows PowerShell command-line interface to retrieve time zone names from the registry, for example: `$timezone = Get-ChildItem "HKLM :\Software\Microsoft\Windows NT`

			`\CurrentVersion\Timezones" | For-Each {Get-ItemProperty $_.PSPath}; $timezone | Format-Table pschildname,display -auto`
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.InboxRuleIdParameter	The *Identity* parameter specifies the identity of an Inbox rule.
IncludeHidden	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the mailbox to which the Inbox rule belongs. You can use one of the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN

			• SmtpAddress • Alias

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-InboxRule

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-InboxRule** cmdlet to remove an Inbox rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-InboxRule -Identity <InboxRuleIdParameter> [-
AlwaysDeleteOutlookRulesBlob <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-Mailbox <MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the Inbox rule ProjectA-MoveToFolderA from the mailbox Joe@Contoso.com.

```
Remove-InboxRule -Mailbox Joe@Contoso.com -Identity
"ProjectA-MoveToFolderA"
```

EXAMPLE 2

This example removes all Inbox rules from the mailbox Joe@Contoso.com.

```
Get-InboxRule -Mailbox "Joe@Contoso.com" | Remove-InboxRule
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Inbox rules" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.InboxRuleIdParameter	The *Identity* parameter specifies the name of the Inbox rule to be removed.
AlwaysDeleteOutlookRulesBlob	Optional	System.Management.Automation.SwitchParameter	The *AlwaysDeleteOutlookRulesBlob* parameter suppresses a warning that end users or administrators get if they use Outlook Web App or Windows PowerShell to modify Inbox rules.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a

			value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress the confirmation prompt produced if rules created by Microsoft Office Outlook exist on the mailbox. When taking an action using Inbox rules on Microsoft Exchange Server 2013, any client-side rules are removed.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the mailbox to which the Inbox rule belongs. You can use one of the following values: • GUID • Distinguished name

			(DN) • *Domain\Name* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-InboxRule

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-28*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-InboxRule** cmdlet to create an Inbox rule for a mailbox. Inbox rules are used to process messages in the Inbox based on conditions specified and take actions such as moving a message to a specified folder or deleting a message.

You must have adequate permissions on the mailbox to create an Inbox rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-InboxRule -Name <String> [-ApplyCategory <MultiValuedProperty>] [-
BodyContainsWords <MultiValuedProperty>] [-CopyToFolder
<MailboxFolderIdParameter>] [-DeleteMessage <$true | $false>] [-
ExceptIfBodyContainsWords <MultiValuedProperty>] [-
ExceptIfFlaggedForAction <String>] [-ExceptIfFrom <RecipientIdParameter[]
>] [-ExceptIfFromAddressContainsWords <MultiValuedProperty>] [-
ExceptIfHasAttachment <$true | $false>] [-ExceptIfHasClassification
<MessageClassificationIdParameter[]>] [-ExceptIfHeaderContainsWords
<MultiValuedProperty>] [-ExceptIfMessageTypeMatches <AutomaticReply |
AutomaticForward | Encrypted | Calendaring | CalendaringResponse |
PermissionControlled | Voicemail | Signed | ApprovalRequest | ReadReceipt
| NonDeliveryReport>] [-ExceptIfMyNameInCcBox <$true | $false>] [-
ExceptIfMyNameInToBox <$true | $false>] [-ExceptIfMyNameInToOrCcBox <$true
| $false>] [-ExceptIfMyNameNotInToBox <$true | $false>] [-
ExceptIfReceivedAfterDate <ExDateTime>] [-ExceptIfReceivedBeforeDate
<ExDateTime>] [-ExceptIfRecipientAddressContainsWords
<MultiValuedProperty>] [-ExceptIfSentOnlyToMe <$true | $false>] [-
ExceptIfSentTo <RecipientIdParameter[]>] [-ExceptIfSubjectContainsWords
<MultiValuedProperty>] [-ExceptIfSubjectOrBodyContainsWords
<MultiValuedProperty>] [-ExceptIfWithImportance <Low | Normal | High>] [-
ExceptIfWithinSizeRangeMaximum <ByteQuantifiedSize>] [-
ExceptIfWithinSizeRangeMinimum <ByteQuantifiedSize>] [-
ExceptIfWithSensitivity <Normal | Personal | Private |
CompanyConfidential>] [-FlaggedForAction <String>] [-ForwardAsAttachmentTo
<RecipientIdParameter[]>] [-ForwardTo <RecipientIdParameter[]>] [-From
<RecipientIdParameter[]>] [-FromAddressContainsWords
<MultiValuedProperty>] [-HasAttachment <$true | $false>] [-
HasClassification <MessageClassificationIdParameter[]>] [-
HeaderContainsWords <MultiValuedProperty>] [-MarkAsRead <$true | $false>]
[-MarkImportance <Low | Normal | High>] [-MessageTypeMatches
<AutomaticReply | AutomaticForward | Encrypted | Calendaring |
CalendaringResponse | PermissionControlled | Voicemail | Signed |
ApprovalRequest | ReadReceipt | NonDeliveryReport>] [-MoveToFolder
<MailboxFolderIdParameter>] [-MyNameInCcBox <$true | $false>] [-
MyNameInToBox <$true | $false>] [-MyNameInToOrCcBox <$true | $false>] [-
MyNameNotInToBox <$true | $false>] [-Priority <Int32>] [-ReceivedAfterDate
<ExDateTime>] [-ReceivedBeforeDate <ExDateTime>] [-
RecipientAddressContainsWords <MultiValuedProperty>] [-RedirectTo
<RecipientIdParameter[]>] [-SendTextMessageNotificationTo
<MultiValuedProperty>] [-SentOnlyToMe <$true | $false>] [-SentTo
<RecipientIdParameter[]>] [-StopProcessingRules <$true | $false>] [-
SubjectContainsWords <MultiValuedProperty>] [-SubjectOrBodyContainsWords
<MultiValuedProperty>] [-WithImportance <Low | Normal | High>] [-
WithinSizeRangeMaximum <ByteQuantifiedSize>] [-WithinSizeRangeMinimum
<ByteQuantifiedSize>] [-WithSensitivity <Normal | Personal | Private |
CompanyConfidential>] <COMMON PARAMETERS>


New-InboxRule -FromMessageId <MailboxStoreObjectIdParameter> -ValidateOnly
<SwitchParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-AlwaysDeleteOutlookRulesBlob <SwitchParameter>] [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
ExceptIfFromSubscription <AggregationSubscriptionIdentity[]>] [-Force
<SwitchParameter>] [-FromSubscription <AggregationSubscriptionIdentity[]>]
```

```
[-Mailbox <MailboxIdParameter>] [-Organization <OrganizationIdParameter>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example raises the message importance to `High` if the mailbox owner is in the To field. In addition, the message is flagged for action.

```
New-InboxRule "CheckActionRequired" -MyNameInToBox $true -
FlaggedForAction Any -MarkImportance "High"
```

Detailed Description

The **New-InboxRule** cmdlet exposes rule predicates and actions, in addition to the parameters such as a rule name and the mailbox identity, required to create an Inbox rule.

Important:
When you create, modify, remove, enable, or disable an Inbox rule on Microsoft Exchange Server 2013, any client-side rules created by Microsoft Outlook are removed.

Predicate parameters used for conditions, such as *SubjectOrBodyContainsWords*, also have an exception parameter. When conditions specified in an exception are matched, the rule isn't applied to the message. Exception parameters begin with *ExceptIf*. For example, the exception parameter for *SubjectOrBodyContainsWords* is *ExceptIfSubjectOrBodyContainsWords*.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Inbox rules" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
FromMessageId	Required	Microsoft.Exchange.Configuration.Tasks.MailboxStoreObjectIdParameter	The *FromMessageId* parameter specifies a message ID to create an Inbox rule based on the properties of that message. You must specify the Base64-

			encoded StoreObjectId of the message. Valid values include one of the following: • MailboxId\StoreObjectId • StoreObjectId When you create an Inbox rule by specifying the *FromMessageId* parameter, the following properties of that message are used to create the rule: • **Subject** The message subject is added to the **SubjectContainsWords** property of the rule. • **From** The message sender is added to the **From** property of the rule. • **To** and **Cc**recipients Recipients in the To and Cc fields of the message are added to the **SentTo** property of the rule.
Name	Required	System.String	The *Name* parameter specifies a name for the Inbox rule being created.
ValidateOnly	Required	System.Management.Automation.SwitchParameter	The *ValidateOnly* switch tells the cmdlet to evaluate the conditions

			and requirements necessary to perform the operation and then reports whether the operation will succeed or fail. No changes are made when the *ValidateOnly* switch is used.
AlwaysDeleteOutlookRulesBlob	Optional	System.Management.Automation.SwitchParameter	The *AlwaysDeleteOutlookRulesBlob* parameter suppresses a warning that end users or administrators get if they use Outlook Web App or Windows PowerShell to modify Inbox rules.
ApplyCategory	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ApplyCategory* parameter specifies one or more categories to apply to a message.
BodyContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BodyContainsWords* parameter specifies one or more words or phrases to check the message body for. If phrases contain a space, you must enclose it in quotation marks ("). Use a comma to separate phrases.
Confirm	Optional	System.Management.	The *Confirm* switch can be

		Automation.SwitchParameter	used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
CopyToFolder	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *CopyToFolder* parameter specifies the name of an existing mailbox folder to copy the message to.
DeleteMessage	Optional	System.Boolean	The *DeleteMessage* parameter specifies whether the message is sent to the Deleted Items folder. If set to `$true`, the message is sent to the Deleted Items folder.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory.
ExceptIfBodyContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfBodyContainsWords* parameter specifies one or more words or phrases to check the message body for. If the message body contains the specified words or phrases, the Inbox rule action isn't applied.
ExceptIfFlaggedForAction	Optional	System.String	The *ExceptIfFlaggedForAction* parameter specifies one or more message flags for which to check. You can use the following values: • `Any` • `Call` • `DoNotForward` • `FollowUp` • `ForYourInformation` • `Forward` • `NoResponseNecessary` • `Read` • `Reply` • `ReplyToAll` • `Review` If the requested action in the message matches the requested action specified in this parameter, the Inbox rule action isn't applied.
ExceptIfFrom	Optional	Microsoft.Exchange.Configuration.Tasks.Reci	The *ExceptIfFrom* parameter specifies a

		pientIdParameter[]	recipient to check in the From field of the message. If the message sender matches the senders specified in this parameter, the Inbox rule action isn't applied.
ExceptIfFromAddressContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfFromAddressContainsWords* parameter specifies one or more words to check in the sender's address. If the sender's address contains the specified words, the Inbox rule action isn't applied to the message.
ExceptIfFromSubscription	Optional	Microsoft.Exchange.Transport.Sync.Common.Subscription.AggregationSubscriptionIdentity[]	This parameter is available only in the cloud-based service. The *ExceptIfFromSubscription* parameter specifies an exception when a message is received from a POP, IMAP, or Hotmail subscription.
ExceptIfHasAttachment	Optional	System.Boolean	The *ExceptIfHasAttachment* parameter specifies whether the rule should

			apply to messages with attachments. Valid values include `$true` and `$false`. We recommend that you set the value to `$true`. If the *ExceptIfHasAttachment* parameter is set to `$true` and the message has an attachment, the Inbox rule action isn't applied.
ExceptIfHasClassification	Optional	Microsoft.Exchange.Configuration.Tasks.MessageClassificationIdParameter[]	The *ExceptIfHasClassification* parameter checks the message for the specified classification name. Use the **Get-MessageClassification** cmdlet to retrieve a list of classifications defined in the organization. If the message classification matches the classification specified in this parameter, the Inbox rule action isn't applied.
ExceptIfHeaderContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfHeaderContainsWords* parameter specifies one or more words to check for in the specified message header.

			If the message header contains the words specified in this parameter, the Inbox rule action isn't applied.
ExceptIfMessageTypeMatches	Optional	Microsoft.Exchange.Data.Storage.InboxRuleMessageType	The *ExceptIfMessageTypeMatches* parameter specifies a message type. You can use the following values: • `AutomaticReply` • `AutomaticForward` • `Encrypted` • `Calendaring` • `CalendaringResponse` • `PermissionControlled` • `Voicemail` • `Signed` • `ApprovalRequest` • `ReadReceipt` • `NonDeliveryReport` If the message type equals one of the preceding values, the Inbox rule action isn't applied.
ExceptIfMyNameInCcBox	Optional	System.Boolean	The *ExceptIfMyNameInCcBox* parameter specifies that the Cc field of messages be checked for the mailbox owner's address. This parameter accepts `$true` or `$false`. If the name of the recipient whose Inbox is being edited is in the Cc

			box, the Inbox rule action isn't applied.
ExceptIfMyNameInToBox	Optional	System.Boolean	The *ExceptIfMyNameInToBox* parameter specifies that the To field of messages be checked for the mailbox owner's address. If the name of the recipient whose Inbox is being edited is in the To box, the Inbox rule action isn't applied. This parameter accepts `$true` or `$false`.
ExceptIfMyNameInToOrCcBox	Optional	System.Boolean	The *ExceptIfMyNameInToOrCcBox* parameter specifies that the To or Cc fields of messages be checked for the mailbox owner's address. If the name of the recipient whose Inbox is being edited is in the To or Cc box, the Inbox rule action isn't applied. This parameter accepts `$true` or `$false`.
ExceptIfMyNameNotIn	Optional	System.Boolean	The *ExceptIfMyNameNotInToB*

ToBox			*ox* parameter matches messages where the mailbox owner isn't addressed in the To field of the message. If the name of the recipient whose Inbox is being edited is not in the To box, the Inbox rule action isn't applied. This parameter accepts `$true` or `$false`.
ExceptIfReceivedAfterDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *ExceptIfReceivedAfterDate* parameter specifies to check for messages received after the specified date. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and

			time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**. If the message was received after the date specified in this parameter, the Inbox rule action isn't applied.
ExceptIfReceivedBeforeDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *ExceptIfReceivedBeforeDate* parameter specifies to check for messages received before the specified date. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in

			quotation marks ("), for example, **"10/05/2010 5:00 PM"**. If the message was received before the time specified in this parameter, the Inbox rule action isn't applied.
ExceptIfRecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfRecipientAddressContainsWords* parameter specifies one or more words to check in the message recipient's address. If the recipient address contains one or more words specified in this parameter, the Inbox rule action isn't applied.
ExceptIfSentOnlyToMe	Optional	System.Boolean	The *ExceptIfSentOnlyToMe* parameter specifies to check for messages where the mailbox owner is the only recipient. This parameter accepts `$true` or `$false`. If the message is only sent to the recipient whose Inbox rule is being edited, the Inbox rule action isn't

			applied.
ExceptIfSentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfSentTo* parameter specifies to check message recipients for the recipient specified. If the message was sent to the recipient specified in this parameter, the Inbox rule action isn't applied.
ExceptIfSubjectContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfSubjectContainsWords* parameter specifies one or more words or phrases to check in the message subject. If the message subject contains one or more words or phrases specified in this parameter, the Inbox rule action isn't applied.
ExceptIfSubjectOrBodyContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfSubjectOrBodyContainsWords* parameter specifies one or more words or phrases to check in the message subject or body. If the message subject or body contains one or more of the words or

			phrases specified in this parameter, the Inbox rule action isn't applied.
ExceptIfWithImportance	Optional	Microsoft.Exchange.Data.Storage.Importance	The *ExceptIfWithImportance* parameter specifies a message importance level. Valid values include one of the following: • `Low` • `Normal` • `High` If the message importance matches the importance specified in this parameter, the Inbox rule action isn't applied.
ExceptIfWithinSizeRangeMaximum	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ExceptIfWithinSizeRangeMaximum* parameter specifies the maximum message size. When using this parameter, you must also specify the minimum message size value using the *ExceptIfWithinSizeRangeMinimum* parameter. If the message size exceeds the maximum value specified in this parameter, the Inbox rule action isn't applied.

ExceptIfWithinSizeRangeMinimum	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ExceptIfWithinSizeRangeMinimum* parameter specifies the minimum message size. When using this parameter, you must also specify a maximum message size value using the *ExceptIfWithinSizeRangeMaximum* parameter. If the message size is smaller than the minimum value specified in this parameter, the Inbox rule action isn't applied.
ExceptIfWithSensitivity	Optional	Microsoft.Exchange.Data.Storage.Sensitivity	The *ExceptIfWithSensitivity* parameter specifies a message sensitivity level. You can use the following values: • `Normal` • `Personal` • `Private` • `CompanyConfidential` If the sensitivity matches the sensitivity value specified in this parameter, the Inbox rule action isn't applied.
FlaggedForAction	Optional	System.String	The *FlaggedForAction*

			parameter specifies one or more message flags for which to check. Values include: • `Any` • `Call` • `DoNotForward` • `FollowUp` • `ForYourInformation` • `Forward` • `NoResponseNecessary` • `Read` • `Reply` • `ReplyToAll` • `Review`
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether the command suppresses the confirmation prompt produced if rules created by Microsoft Outlook exist on the mailbox. When taking an action using Inbox rules on Exchange 2013, any client-side rules are removed.
ForwardAsAttachmentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ForwardAsAttachmentTo* parameter specifies a recipient to forward the message to as an attachment.
ForwardTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ForwardTo* parameter specifies a recipient to forward the message to.
From	Optional	Microsoft.Exchange.Co	The *From* parameter

		nfiguration.Tasks.RecipientIdParameter[]	specifies the identity of the sender as a rule condition.
FromAddressContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *FromAddressContainsWords* parameter specifies one or more words to check in the From field of the message.
FromSubscription	Optional	Microsoft.Exchange.Transport.Sync.Common.Subscription.AggregationSubscriptionIdentity[]	This parameter is available only in the cloud-based service. The *FromSubscription* parameter specifies the condition when a message is received from a POP, IMAP, or Hotmail subscription.
HasAttachment	Optional	System.Boolean	The *HasAttachment* parameter specifies whether to check for messages with attachments.
HasClassification	Optional	Microsoft.Exchange.Configuration.Tasks.MessageClassificationIdParameter[]	The *HasClassification* parameter checks messages for the specified message classification.
HeaderContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *HeaderContainsWords* parameter specifies one

			or more words to match in the specified message header.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the mailbox for which the rule is being created. The following values can be used: • `GUID` • `Distinguished name (DN)` • *Domain\Account* • `User principal name (UPN)` • `LegacyExchangeDN` • `SmtpAddress` • `Alias`
MarkAsRead	Optional	System.Boolean	The *MarkAsRead* parameter specifies whether to mark the message as read. When set to `$true`, the *MarkAsRead* parameter marks the message as read.
MarkImportance	Optional	Microsoft.Exchange.Data.Storage.Importance	The *MarkImportance* parameter specifies that the message importance be marked with one of the following values: • `Low` • `Normal` • `High`
MessageTypeMatches	Optional	Microsoft.Exchange.Data.Storage.InboxRuleM	The *MessageTypeMatches* parameter specifies a message type. You can

		essageType	use the following values: • `AutomaticReply` • `AutomaticForward` • `Encrypted` • `Calendaring` • `CalendaringResponse` • `PermissionControlled` • `Voicemail` • `Signed` • `ApprovalRequest` • `ReadReceipt` • `NonDeliveryReport`
MoveToFolder	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *MoveToFolder* parameter specifies an existing mailbox folder to which the message is copied.
MyNameInCcBox	Optional	System.Boolean	The *MyNameInCcBox* parameter specifies that the mailbox for which the rule is being created should appear in the Cc field of the message. To use this predicate, set the value to `$true`.
MyNameInToBox	Optional	System.Boolean	The *MyNameInToBox* parameter specifies that the mailbox for which the rule is being created should appear in the To field of the message. To use this predicate, set the value to `$true`.
MyNameInToOrCcBox	Optional	System.Boolean	The *MyNameInToOrCcBox* parameter specifies that the mailbox for which the

			rule is being created should appear in the To or Cc fields of the message. To add this condition, set the value to `$true`.
MyNameNotInToBox	Optional	System.Boolean	The *MyNameNotInToBox* parameter checks that the mailbox owner's name isn't in the To box. We recommend that you set the value to `$true` to use this predicate.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
Priority	Optional	System.Int32	The *Priority* parameter sets a priority for the Inbox rule.
ReceivedAfterDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *ReceivedAfterDate* parameter specifies a date. The rule is applied to messages received after the specified date. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/

			yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
ReceivedBeforeDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *ReceivedBeforeDate* parameter specifies a date. The rule is applied only to messages received before the specified date. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in

			quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
RecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RecipientAddressContainsWords* parameter specifies one or more words to check for in a recipient address.
RedirectTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *RedirectTo* parameter specifies a recipient to redirect the message to.
SendTextMessageNotificationTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SendTextMessageNotificationTo* parameter specifies one or more text message recipients to send a notification to.
SentOnlyToMe	Optional	System.Boolean	The *SentOnlyToMe* parameter specifies whether the mailbox owner is specified as the only recipient. You can use the following values: • `$true` • `$false`
SentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentTo* parameter specifies the identity of a recipient as a condition.
StopProcessingRules	Optional	System.Boolean	The *StopProcessingRules* parameter specifies that

			Exchange stop processing additional rules if the conditions of this Inbox rule are met.
SubjectContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SubjectContainsWords* parameter specifies one or more keywords to be matched in the subject field of a message.
SubjectOrBodyContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SubjectOrBodyContainsWords* parameter specifies one or more words to be matched in the message subject or body.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WithImportance	Optional	Microsoft.Exchange.Data.Storage.Importance	The *WithImportance* parameter checks messages with the

			specified importance level. You can use the following values: • `High` • `Normal` • `Low`
WithinSizeRangeMaximum	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *WithinSizeRangeMaximum* parameter specifies the maximum message size. When using this parameter, you must also specify a minimum message size value using the *WithinSizeRangeMinimum* parameter.
WithinSizeRangeMinimum	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *WithinSizeRangeMinimum* parameter specifies the minimum message size. When using this parameter, you must also specify a maximum message size value using the *WithinSizeRangeMaximum* parameter.
WithSensitivity	Optional	Microsoft.Exchange.Data.Storage.Sensitivity	The *WithSensitivity* parameter specifies a message sensitivity level. Valid values include one of the following: • `Normal`

			• Personal • Private • CompanyConfidential

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-InboxRule

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-28*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-InboxRule** cmdlet to modify an existing Inbox rule. Inbox rules are used to process messages in the Inbox based on conditions specified and take actions such as moving a message to a specified folder or deleting a message.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-InboxRule -Identity <InboxRuleIdParameter> [-
AlwaysDeleteOutlookRulesBlob <SwitchParameter>] [-ApplyCategory
<MultiValuedProperty>] [-BodyContainsWords <MultiValuedProperty>] [-
Confirm [<SwitchParameter>]] [-CopyToFolder <MailboxFolderIdParameter>] [-
DeleteMessage <$true | $false>] [-DomainController <Fqdn>] [-
ExceptIfBodyContainsWords <MultiValuedProperty>] [-
ExceptIfFlaggedForAction <String>] [-ExceptIfFrom <RecipientIdParameter[]
>] [-ExceptIfFromAddressContainsWords <MultiValuedProperty>] [-
ExceptIfFromSubscription <AggregationSubscriptionIdentity[]>] [-
ExceptIfHasAttachment <$true | $false>] [-ExceptIfHasClassification
<MessageClassificationIdParameter[]>] [-ExceptIfHeaderContainsWords
<MultiValuedProperty>] [-ExceptIfMessageTypeMatches <AutomaticReply |
AutomaticForward | Encrypted | Calendaring | CalendaringResponse |
PermissionControlled | Voicemail | Signed | ApprovalRequest | ReadReceipt
| NonDeliveryReport>] [-ExceptIfMyNameInCcBox <$true | $false>] [-
ExceptIfMyNameInToBox <$true | $false>] [-ExceptIfMyNameInToOrCcBox <$true
| $false>] [-ExceptIfMyNameNotInToBox <$true | $false>] [-
ExceptIfReceivedAfterDate <ExDateTime>] [-ExceptIfReceivedBeforeDate
<ExDateTime>] [-ExceptIfRecipientAddressContainsWords
<MultiValuedProperty>] [-ExceptIfSentOnlyToMe <$true | $false>] [-
ExceptIfSentTo <RecipientIdParameter[]>] [-ExceptIfSubjectContainsWords
<MultiValuedProperty>] [-ExceptIfSubjectOrBodyContainsWords
<MultiValuedProperty>] [-ExceptIfWithImportance <Low | Normal | High>] [-
```

```
ExceptIfWithinSizeRangeMaximum <ByteQuantifiedSize>] [-
ExceptIfWithinSizeRangeMinimum <ByteQuantifiedSize>] [-
ExceptIfWithSensitivity <Normal | Personal | Private |
CompanyConfidential>] [-FlaggedForAction <String>] [-Force
<SwitchParameter>] [-ForwardAsAttachmentTo <RecipientIdParameter[]>] [-
ForwardTo <RecipientIdParameter[]>] [-From <RecipientIdParameter[]>] [-
FromAddressContainsWords <MultiValuedProperty>] [-FromSubscription
<AggregationSubscriptionIdentity[]>] [-HasAttachment <$true | $false>] [-
HasClassification <MessageClassificationIdParameter[]>] [-
HeaderContainsWords <MultiValuedProperty>] [-Mailbox <MailboxIdParameter>]
[-MarkAsRead <$true | $false>] [-MarkImportance <Low | Normal | High>] [-
MessageTypeMatches <AutomaticReply | AutomaticForward | Encrypted |
Calendaring | CalendaringResponse | PermissionControlled | Voicemail |
Signed | ApprovalRequest | ReadReceipt | NonDeliveryReport>] [-
MoveToFolder <MailboxFolderIdParameter>] [-MyNameInCcBox <$true | $false>]
[-MyNameInToBox <$true | $false>] [-MyNameInToOrCcBox <$true | $false>] [-
MyNameNotInToBox <$true | $false>] [-Name <String>] [-Priority <Int32>] [-
ReceivedAfterDate <ExDateTime>] [-ReceivedBeforeDate <ExDateTime>] [-
RecipientAddressContainsWords <MultiValuedProperty>] [-RedirectTo
<RecipientIdParameter[]>] [-SendTextMessageNotificationTo
<MultiValuedProperty>] [-SentOnlyToMe <$true | $false>] [-SentTo
<RecipientIdParameter[]>] [-StopProcessingRules <$true | $false>] [-
SubjectContainsWords <MultiValuedProperty>] [-SubjectOrBodyContainsWords
<MultiValuedProperty>] [-WhatIf [<SwitchParameter>]] [-WithImportance <Low
| Normal | High>] [-WithinSizeRangeMaximum <ByteQuantifiedSize>] [-
WithinSizeRangeMinimum <ByteQuantifiedSize>] [-WithSensitivity <Normal |
Personal | Private | CompanyConfidential>]
```

Examples

EXAMPLE 1

This example modifies the action of the existing Inbox rule ProjectContoso. The *MarkImportance* parameter is used to mark the message with high importance.

```
Set-InboxRule ProjectContoso -MarkImportance "High"
```

Detailed Description

The **Set-InboxRule** cmdlet exposes rule predicates and actions, in addition to the parameters such as a rule name and the mailbox identity, required to create an Inbox rule.

Important:
When you create, modify, remove, enable, or disable an Inbox rule on Microsoft Exchange Server 2013, some client-side rules including disabled client-side rules and outbound rules that were created by Microsoft Outlook are removed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Inbox rules" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.InboxRuleIdParameter	The *Identity* parameter specifies the name of the Inbox rule to be modified.
AlwaysDeleteOutlookRulesBlob	Optional	System.Management.Automation.SwitchParameter	The *AlwaysDeleteOutlookRulesBlob* parameter suppresses a warning that end users or administrators get if they use Outlook Web App or Windows PowerShell to modify Inbox rules.
ApplyCategory	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ApplyCategory* parameter specifies one or more categories to apply to a message.
BodyContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BodyContainsWords* parameter specifies one or more words or phrases to check the message body for. If phrases contain a space, you must enclose it in quotation marks ("). Use a comma to separate phrases.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before

			processing continues. You don't have to specify a value with the *Confirm* switch.
CopyToFolder	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *CopyToFolder* parameter specifies the name of an existing mailbox folder to copy the message to.
DeleteMessage	Optional	System.Boolean	The *DeleteMessage* parameter specifies whether the message is sent to the Deleted Items folder. If set to `$true`, the *DeleteMessage* parameter specifies that the message is sent to the Deleted Items folder.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in the cloud-based service. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExceptIfBodyContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *ExceptIfBodyContainsWor*

		y	*ds* parameter specifies one or more words or phrases to check the message body for. If the message body contains the specified words or phrases, the Inbox rule action isn't applied.
ExceptIfFlaggedForAction	Optional	System.String	The *ExceptIfFlaggedForAction* parameter specifies one or more message flags to check the message for. Valid values include: • `Any` • `Call` • `DoNotForward` • `FollowUp` • `ForYourInformation` • `Forward` • `NoResponseNecessary` • `Read` • `Reply` • `ReplyToAll` • `Review` If the requested action in the message matches the requested action specified in this parameter, the Inbox rule action isn't applied.
ExceptIfFrom	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfFrom* parameter specifies one or more senders. If the message sender matches the senders specified in

			this parameter, the Inbox rule action isn't applied.
ExceptIfFromAddressContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfFromAddressContainsWords* parameter specifies one or more words to check the sender's address for. If the sender's address contains the specified words, the Inbox rule action isn't applied to the message.
ExceptIfFromSubscription	Optional	Microsoft.Exchange.Transport.Sync.Common.Subscription.AggregationSubscriptionIdentity[]	This parameter is available only in the cloud-based service. The *ExceptIfFromSubscription* parameter specifies an exception when a message is received from a POP, IMAP, or Hotmail subscription.
ExceptIfHasAttachment	Optional	System.Boolean	The *ExceptIfHasAttachment* parameter specifies whether the message has attachments. Valid values include `$true` and `$false`. We recommend you set the value to `$true`. If the message has an

			attachment, the Inbox rule action isn't applied.
ExceptIfHasClassification	Optional	Microsoft.Exchange.Configuration.Tasks.MessageClassificationIdParameter[]	The *ExceptIfHasClassification* parameter checks the message for the specified classification name. Use the **Get-MessageClassification** cmdlet to retrieve a list of classifications defined in the organization. If the message classification matches the classification specified in this parameter, the Inbox rule action isn't applied.
ExceptIfHeaderContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfHeaderContainsWords* parameter specifies one or more values for a header. If the message header contains the words specified in this parameter, the Inbox rule action isn't applied.
ExceptIfMessageTypeMatches	Optional	Microsoft.Exchange.Data.Storage.InboxRuleMessageType	The *ExceptIfMessageTypeMatches* parameter specifies a message type. Valid

			values include one of the following: • `AutomaticReply` • `AutomaticForward` • `Encrypted` • `Calendaring` • `CalendaringResponse` • `PermissionControlled` • `Voicemail` • `Signed` • `ApprovalRequest` • `ReadReceipt` • `NonDeliveryReport` If the message type equals one of the preceding values, the Inbox rule action isn't applied.
ExceptIfMyNameInCcBox	Optional	System.Boolean	The *ExceptIfMyNameInCcBox* parameter specifies that the Cc field of messages is checked for the mailbox owner's address. If the name of the recipient whose Inbox is being edited is in the Cc box, the Inbox rule action isn't applied. This parameter accepts `$true` or `$false`.
ExceptIfMyNameInToBox	Optional	System.Boolean	The *ExceptIfMyNameInToBox* parameter specifies that the To field of messages is checked for the mailbox owner's address.

			If the name of the recipient whose Inbox is being edited is in the To box, the Inbox rule action isn't applied. This parameter accepts `$true` or `$false`.
ExceptIfMyNameInToOrCcBox	Optional	System.Boolean	The *ExceptIfMyNameInToOrCcBox* parameter specifies that the To or Cc field of messages is checked for the mailbox owner's address. If the name of the recipient whose Inbox is being edited is in the To or Cc box, the Inbox rule action isn't applied. This parameter accepts `$true` or `$false`.
ExceptIfMyNameNotInToBox	Optional	System.Boolean	The *ExceptIfMyNameNotInToBox* parameter matches messages where the mailbox owner isn't addressed in the To field of the message. If the name of the recipient whose Inbox is being edited isn't in the To

			box, the Inbox rule action isn't applied. This parameter accepts `$true` or `$false`.
ExceptIfReceivedAfterDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *ExceptIfReceivedAfterDate* parameter specifies to check for messages received after the specified date. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**. If the message was received after the date specified, the Inbox rule

			action isn't applied.
ExceptIfReceivedBeforeDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *ExceptIfReceivedBeforeDate* parameter specifies to check for messages received before the specified date. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**. If the message was received before the time specified in this parameter, the Inbox rule action isn't applied.

ExceptIfRecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfRecipientAddressContainsWords* parameter specifies one or more words to check in the message recipient's address. If the recipient address contains one or more words specified in this parameter, the Inbox rule action isn't applied.
ExceptIfSentOnlyToMe	Optional	System.Boolean	The *ExceptIfSentOnlyToMe* parameter specifies to check for messages where the mailbox owner is the only recipient. This parameter accepts `$true` or `$false`. If the message is only sent to the recipient whose Inbox rule is being edited, the Inbox rule action isn't applied.
ExceptIfSentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfSentTo* parameter specifies to check message recipients for the recipient specified. If the message is sent to the recipient specified in this parameter, the Inbox

			rule action isn't applied.
ExceptIfSubjectContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfSubjectContainsWords* parameter specifies one or more words or phrases to check in the message subject. If the message subject contains one or more words or phrases specified in this parameter, the Inbox rule action isn't applied.
ExceptIfSubjectOrBodyContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfSubjectOrBodyContainsWords* parameter specifies one or more words or phrases to check in the message subject or body. If the message subject or body contains one or more of the words or phrases specified in this parameter, the Inbox rule action isn't applied.
ExceptIfWithImportance	Optional	Microsoft.Exchange.Data.Storage.Importance	The *ExceptIfWithImportance* parameter checks the message for the specified importance level. Valid

			values include: • High • Normal • Low If the message importance matches the importance specified in this parameter, the Inbox rule action isn't applied.
ExceptIfWithinSizeRangeMaximum	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ExceptIfWithinSizeRangeMaximum* parameter specifies a maximum message size. When using this parameter, you must also specify the minimum message size using the *ExceptIfWithinSizeRangeMinimum* parameter. If the message size exceeds the maximum value specified in this parameter, the Inbox rule action isn't applied.
ExceptIfWithinSizeRangeMinimum	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ExceptIfWithinSizeRangeMinimum* parameter specifies a minimum message size. When using this parameter, you must also specify the maximum message size using the *ExceptIfWithinSizeRangeM*

			aximum parameter. If the message size is smaller than the minimum value specified in this parameter, the Inbox rule action isn't applied.
ExceptIfWithSensitivity	Optional	Microsoft.Exchange.Data.Storage.Sensitivity	The *ExceptIfWithSensitivity* parameter checks messages for the specified sensitivity level. Valid values include: • `Normal` • `Personal` • `Private` • `CompanyConfidential` If the sensitivity matches the sensitivity value specified in this parameter, the Inbox rule action isn't applied.
FlaggedForAction	Optional	System.String	The *FlaggedForAction* parameter specifies one or more message flags to check the message for. Valid values include: • `Any` • `Call` • `DoNotForward` • `FollowUp` • `ForYourInformation` • `Forward` • `NoResponseNecessary` • `Read` • `Reply` • `ReplyToAll` • `Review`

Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress the confirmation prompt produced if rules created by Outlook exist on the mailbox. When taking an action using Inbox rules on Exchange 2013, any client-side rules are removed.
ForwardAsAttachmentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ForwardAsAttachmentTo* parameter specifies a recipient to forward the message to as an attachment.
ForwardTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ForwardTo* parameter specifies the identity of a recipient to forward the message to.
From	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *From* parameter specifies the identity of the sender as a rule condition.
FromAddressContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *FromAddressContainsWords* parameter specifies one or more words to check in the From address of the message.

FromSubscription	Optional	Microsoft.Exchange.Transport.Sync.Common.Subscription.AggregationSubscriptionIdentity[]	This parameter is available only in the cloud-based service. The *FromSubscription* parameter specifies the condition when a message is received from a POP, IMAP, or Hotmail, subscription.
HasAttachment	Optional	System.Boolean	The *HasAttachment* parameter specifies whether to check for messages with attachments.
HasClassification	Optional	Microsoft.Exchange.Configuration.Tasks.MessageClassificationIdParameter[]	The *HasClassification* parameter checks messages for the specified message classification.
HeaderContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *HeaderContainsWords* parameter specifies one or more words to match in the specified message header.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the mailbox for which the new rule is being created. You can use the following values: • GUID

			• Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
MarkAsRead	Optional	System.Boolean	The *MarkAsRead* parameter specifies whether to mark the message as read. When set to `$true`, the *MarkAsRead* parameter marks the message as read.
MarkImportance	Optional	Microsoft.Exchange.Data.Storage.Importance	The *MarkImportance* parameter specifies that the message importance be marked with one of the following values: • `Low` • `Normal` • `High`
MessageTypeMatches	Optional	Microsoft.Exchange.Data.Storage.InboxRuleMessageType	The *MessageTypeMatches* parameter specifies a message type to apply the rule to. Valid values include: • `AutomaticReply` • `AutomaticForward` • `Encrypted` • `Calendaring` • `CalendaringResponse` • `PermissionControlled` • `Voicemail` • `Signed` • `ApprovalRequest`

			• `ReadReceipt` • `NonDeliveryReport`
MoveToFolder	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *MoveToFolder* parameter specifies an existing mailbox folder to which the message is copied.
MyNameInCcBox	Optional	System.Boolean	The *MyNameInCcBox* parameter specifies that the mailbox for which the rule is being created should appear in the Cc field of the message. To use this predicate, set the value to `$true`.
MyNameInToBox	Optional	System.Boolean	The *MyNameInToBox* parameter specifies that the mailbox for which the rule is being created should appear in the To field of the message. To use this predicate, set the value to `$true`.
MyNameInToOrCcBox	Optional	System.Boolean	The *MyNameInToOrCcBox* parameter specifies that the mailbox for which the rule is being created should appear in the To or Cc fields of the message. To add this condition, set the value to `$true`.
MyNameNotInToBox	Optional	System.Boolean	The *MyNameNotInToBox*

			parameter specifies that the mailbox for which the rule is being created should not appear in the To field of the message. To use this predicate, set the value to `$true`.
Name	Optional	System.String	The *Name* parameter specifies a name for the Inbox rule.
Priority	Optional	System.Int32	The *Priority* parameter specifies a priority for the Inbox rule.
ReceivedAfterDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *ReceivedAfterDate* parameter specifies a date. The rule is applied to messages received after the specified date. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If

			you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
ReceivedBeforeDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *ReceivedBeforeDate* parameter specifies a date. The rule is applied only to messages received before the specified date. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
RecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *RecipientAddressContains*

		y	*Words* parameter specifies one or more words to check for in the recipient address.
RedirectTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *RedirectTo* parameter specifies a recipient to redirect the message to.
SendTextMessageNotificationTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SendTextMessageNotificationTo* parameter specifies one or more text message recipients to send a notification to.
SentOnlyToMe	Optional	System.Boolean	The *SentOnlyToMe* parameter specifies whether a message is sent only to the mailbox owner.
SentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentTo* parameter specifies the identity of a recipient as a condition.
StopProcessingRules	Optional	System.Boolean	The *StopProcessingRules* parameter specifies whether Exchange stops processing additional rules if the conditions of this Inbox rule are met. If set to `$true`, the *StopProcessingRules* parameter instructs

			Exchange to stop processing additional rules if the conditions of this Inbox rule are met.
SubjectContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SubjectContainsWords* parameter specifies one or more keywords to be matched in the subject field of a message.
SubjectOrBodyContainsWords	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SubjectOrBodyContainsWords* parameter specifies one or more words to be matched in the message subject or body.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WithImportance	Optional	Microsoft.Exchange.Data.Storage.Importance	The *WithImportance* parameter specifies that messages with the

			specified importance level are checked. Valid values include: • `High` • `Normal` • `Low`
WithinSizeRangeMaximum	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *WithinSizeRangeMaximum* parameter specifies the maximum message size. When using this parameter, you must also specify a minimum message size value using the *WithinSizeRangeMinimum* parameter.
WithinSizeRangeMinimum	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *WithinSizeRangeMinimum* parameter specifies the minimum message size. When using this parameter, you must also specify a maximum message size value using the *WithinSizeRangeMaximum* parameter.
WithSensitivity	Optional	Microsoft.Exchange.Data.Storage.Sensitivity	The *WithSensitivity* parameter specifies a message sensitivity level. Valid values include one of the following: • `Normal`

			• Personal • Private • CompanyConfidential

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Connect-Mailbox

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Connect-Mailbox** cmdlet to connect a disconnected mailbox to an Active Directory user object.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Connect-Mailbox [-ActiveSyncMailboxPolicy <MailboxPolicyIdParameter>] [-
AddressBookPolicy <AddressBookMailboxPolicyIdParameter>] [-Alias <String>]
[-AllowLegacyDNMismatch <SwitchParameter>] [-Archive <SwitchParameter>] [-
ManagedFolderMailboxPolicy <MailboxPolicyIdParameter>] [-
ManagedFolderMailboxPolicyAllowed <SwitchParameter>] [-RetentionPolicy
<MailboxPolicyIdParameter>] [-User <UserIdParameter>] <COMMON PARAMETERS>


Connect-Mailbox -ValidateOnly <SwitchParameter> <COMMON PARAMETERS>


Connect-Mailbox -Shared <SwitchParameter> [-ActiveSyncMailboxPolicy
<MailboxPolicyIdParameter>] [-Alias <String>] [-ManagedFolderMailboxPolicy
<MailboxPolicyIdParameter>] [-ManagedFolderMailboxPolicyAllowed
<SwitchParameter>] [-RetentionPolicy <MailboxPolicyIdParameter>] [-User
<UserIdParameter>] <COMMON PARAMETERS>


Connect-Mailbox -Room <SwitchParameter> [-ActiveSyncMailboxPolicy
<MailboxPolicyIdParameter>] [-Alias <String>] [-ManagedFolderMailboxPolicy
<MailboxPolicyIdParameter>] [-ManagedFolderMailboxPolicyAllowed
<SwitchParameter>] [-RetentionPolicy <MailboxPolicyIdParameter>] [-User
<UserIdParameter>] <COMMON PARAMETERS>


Connect-Mailbox -Equipment <SwitchParameter> [-ActiveSyncMailboxPolicy
<MailboxPolicyIdParameter>] [-Alias <String>] [-ManagedFolderMailboxPolicy
<MailboxPolicyIdParameter>] [-ManagedFolderMailboxPolicyAllowed
```

```
<SwitchParameter>] [-RetentionPolicy <MailboxPolicyIdParameter>] [-User
<UserIdParameter>] <COMMON PARAMETERS>


Connect-Mailbox -LinkedDomainController <Fqdn> -LinkedMasterAccount
<UserIdParameter> [-ActiveSyncMailboxPolicy <MailboxPolicyIdParameter>] [-
Alias <String>] [-LinkedCredential <PSCredential>] [-
ManagedFolderMailboxPolicy <MailboxPolicyIdParameter>] [-
ManagedFolderMailboxPolicyAllowed <SwitchParameter>] [-RetentionPolicy
<MailboxPolicyIdParameter>] [-User <UserIdParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Database <DatabaseIdParameter> -Identity
<StoreMailboxIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Force <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example connects John Evans' (john@contoso.com) disconnected mailbox. The example doesn't specify a user object to connect the mailbox to, so the command attempts to find a uniquely matched user object to connect the mailbox to.

```
Connect-Mailbox -Database "Mailbox Database" -Identity
"John Evans"
```

EXAMPLE 2

This example connects a linked mailbox.

```
Connect-Mailbox -Identity "John Evans" -Database "MBXDB02"
-LinkedDomainController FabrikamDC01 -LinkedMasterAccount
john@fabrikam.com
```

EXAMPLE 3

This example connects an equipment mailbox.

```
Connect-Mailbox -Identity "CAR001" -Database
"MBXResourceDB" -Equipment -User "CAR001"
```

EXAMPLE 4

This example connects a room mailbox.

```
Connect-Mailbox -Identity "ConfRm212" -Database
"MBXResourceDB" -Room -User "Conference Room 212"
```

Detailed Description

If you link a mailbox to an existing Active Directory user object, that Active Directory user object has full access to the mailbox and all mail in the mailbox. If you use the *User* parameter to specify the Active Directory user account, make sure you specify the correct account. If you don't use the *User* parameter, we recommend that you use the *ValidateOnly* parameter to verify which user account the mailbox is connected to.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Database	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Database* parameter specifies the Exchange database that contains the mailbox that you want to connect. You can use the following values: • GUID of the database • Database name
Equipment	Required	System.Management.Automation.SwitchParameter	The *Equipment* switch specifies that the type of resource is equipment, if this mailbox is a resource mailbox. This switch is required only if you're connecting a resource mailbox.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.StoreMailboxIdParameter	The *Identity* parameter specifies the mailbox object in the Exchange

			database to connect to an Active Directory user object. This parameter doesn't specify an Active Directory object. You can use the following values: • GUID of the mailbox • Display name • LegacyExchangeDN
LinkedDomainController	Required	Microsoft.Exchange.Data.Fqdn	The *LinkedDomainController* parameter specifies the domain controller in the forest where the user account resides, if this mailbox is a linked mailbox. The domain controller in the forest where the user account resides is used to get security information for the account specified by the *LinkedMasterAccount* parameter. This parameter is required only if you're connecting a linked mailbox.
LinkedMasterAccount	Required	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	The *LinkedMasterAccount* parameter specifies the master account in the forest where the user account resides, if this

			mailbox is a linked mailbox. The master account is the account to which the mailbox links. The master account grants access to the mailbox. You can use one of the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias This parameter is required only if you're connecting a linked mailbox.
Room	Required	System.Management.Automation.SwitchParameter	The *Room* switch specifies that the type of resource is a room, if this mailbox is a resource mailbox. This switch is required only if you're connecting a resource mailbox.
Shared	Required	System.Management.Automation.SwitchParameter	The *Shared* switch specifies that you're creating a shared mailbox. A shared mailbox is a mailbox to which multiple

			users can log on. This mailbox isn't associated with any of the users that can log on. It's associated with a disabled user account. This switch is required only if you're connecting a shared mailbox.
ValidateOnly	Required	System.Management.Automation.SwitchParameter	The *ValidateOnly* switch tells the cmdlet to evaluate the conditions and requirements necessary to perform the operation and then reports whether the operation will succeed or fail. No changes are made when the *ValidateOnly* switch is used.
ActiveSyncMailboxPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *ActiveSyncMailboxPolicy* parameter specifies the Microsoft ActiveSync mailbox policy to enable for the mailbox after it's connected.
AddressBookPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.AddressBookMailboxPolicyIdParameter	The *AddressBookPolicy* parameter specifies the Address book policy to apply to this mailbox

			when it's connected.
Alias	Optional	System.String	The *Alias* parameter specifies the alias (mail nickname) for the mailbox after it's connected. The alias can be a combination of characters separated by a period with no intervening spaces. Don't use special characters in the alias.
AllowLegacyDNMismatch	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* switch specifies whether to connect the archive mailbox. You don't have to specify a value with this switch.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, administrative input is prompted. If the *Force* switch is provided in the command, but the value is omitted, its default value is `$true`.
LinkedCredential	Optional	System.Management.Automation.PSCredential	The *LinkedCredential* parameter specifies credentials to use to access the domain controller specified by the *LinkedDomainController* parameter. This parameter

			is optional, even if you're enabling a linked mailbox.
ManagedFolderMailboxPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *ManagedFolderMailboxPolicy* parameter specifies a managed folder policy object for the mailbox.
ManagedFolderMailboxPolicyAllowed	Optional	System.Management.Automation.SwitchParameter	The *ManagedFolderMailboxPolicyAllowed* switch bypasses the warning that messaging records management (MRM) features aren't supported for email clients using versions of Microsoft Outlook earlier than Office Outlook 2007. When a managed folder mailbox policy is assigned to a mailbox using the *ManagedFolderMailboxPolicy* parameter, the warning appears by default unless the *ManagedFolderMailboxPolicyAllowed* switch is used. **Note:** Outlook 2003 Service Pack 3 clients are supported but are provided limited functionality for MRM.

RetentionPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *RetentionPolicy* parameter specifies the name of a retention policy that you want applied to this mailbox. Retention policies consist of tags that are applied to mailbox folders and mail items to determine the period of time that the items should be retained.
User	Optional	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	The *User* parameter specifies the user object in Active Directory to which you want to connect the Exchange mailbox object. If you don't specify this parameter, the command uses the **LegacyExchangeDN** and **DisplayName** attributes of the Exchange mailbox object to find a user account that matches the mailbox object. If it can't find a unique match, it doesn't connect the mailbox.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-Mailbox

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-Mailbox** cmdlet to disable the mailbox of an existing user, InetOrgPerson object, or public folder mailbox and remove that object's Exchange attributes from Active Directory. The user account associated with the mailbox remains in Active Directory, but it's no longer associated with a mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-Mailbox <COMMON PARAMETERS>

Disable-Mailbox [-Archive <SwitchParameter>] <COMMON PARAMETERS>
```

```
Disable-Mailbox [-RemoteArchive <SwitchParameter>] <COMMON PARAMETERS>


Disable-Mailbox [-Arbitration <SwitchParameter>] [-
DisableArbitrationMailboxWithOABsAllowed <SwitchParameter>] [-
DisableLastArbitrationMailboxAllowed <SwitchParameter>] <COMMON
PARAMETERS>


Disable-Mailbox [-PublicFolder <SwitchParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreDefaultScope
<SwitchParameter>] [-IgnoreLegalHold <SwitchParameter>] [-
IncludeSoftDeletedObjects <SwitchParameter>] [-PreserveEmailAddresses
<SwitchParameter>] [-PreventRecordingPreviousDatabase <SwitchParameter>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the mailbox of user John Woods, whose alias is john, and removes all the mailbox attributes from Active Directory.

```
Disable-Mailbox john@contoso.com
```

EXAMPLE 2

This example disables the remote archive for the mailbox of user John Woods, whose alias is john.

```
Disable-Mailbox -Identity john@contoso.com -RemoteArchive
```

Detailed Description

The **Disable-Mailbox** cmdlet removes the mailbox's Exchange attributes from Active Directory. The mailbox isn't deleted and can be reconnected to its user at a later date by using the **Connect-Mailbox** cmdlet.

The **Disable-Mailbox** cmdlet also performs the clean-up task on the individual mailbox, so the mailbox is disconnected immediately after this task completes.

Under normal circumstances, a mailbox is marked as disconnected immediately after the **Disable-Mailbox** or **Remove-Mailbox** command completes. However, if the mailbox was disabled or removed while the Microsoft Exchange Information Store service was stopped, or if it was disabled or removed by an external means other than Exchange management interfaces, the status of the mailbox object in the Exchange mailbox database won't be marked as disconnected.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient

Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name`

			Example: JPhillips@contoso.com
Arbitration	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Arbitration* parameter specifies that the mailbox for which you are executing the command is an arbitration mailbox. Arbitration mailboxes are used for managing approval workflow. For example, an arbitration mailbox is used for handling moderated recipients and distribution group membership approval.
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* switch specifies whether to disconnect the archive mailbox from the associated mailbox user.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You

			must include a colon (:) in the syntax.
DisableArbitrationMailboxWithOABsAllowed	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *DisableArbitrationMailboxWithOABsAllowed* parameter is used bypass the checks for offline address books (OABs) within the specified arbitration mailbox that is being disabled, and disable the arbitration mailbox even if OABs are present in the mailbox.
DisableLastArbitrationMailboxAllowed	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *DisableLastArbitrationMailboxAllowed* switch specifies whether to disable the specified mailbox if it's the last arbitration mailbox in the organization. You don't have to specify a value with this parameter. If you disable the last arbitration

			mailbox in the organization, you can't have user-created distribution groups or moderated recipient functionality.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default

			scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
IgnoreLegalHold	Optional	System.Management.Automation.SwitchParameter	The *IgnoreLegalHold* switch ignores the legal hold status of the mailbox and allows you to disable a mailbox that's on legal hold. **⚠Warning:** When you disable a mailbox, the mailbox is disconnected from the user account. After you disable a mailbox, you can't include it in a discovery search. Disconnected mailboxes are permanently deleted from the mailbox database after the deleted mailbox retention period expires. Check with your

			organization's legal or Human Resources department before disabling a mailbox that's on legal hold.
IncludeSoftDeletedObjects	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PreserveEmailAddresses	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PreventRecordingPreviousDatabase	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PublicFolder	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *PublicFolder* parameter specifies that the mailbox to disable is a public folder mailbox. Public folder mailboxes are specially designed mailboxes to store the hierarchy and content of public folders. You have to include this parameter to disable a public folder mailbox.
RemoteArchive	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013.

			The *RemoteArchive* parameter specifies whether to disconnect the remote archive for this mailbox. A remote archive exists in a cloud-based service. When you use this parameter, the **RemoteRecipientType** property for the mailbox is reset to specify that this mailbox doesn't have a remote archive. You don't need to specify a value with this parameter. When you use this parameter, you can't use the *Archive* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-Mailbox

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-22*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-Mailbox** cmdlet to mailbox-enable an existing user, public folder mailbox, or Active Directory **InetOrgPerson** object.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-Mailbox [-AccountDisabled <$true | $false>] [-AddOnSKUCapability
<MultiValuedProperty>] [-AddressBookPolicy
<AddressBookMailboxPolicyIdParameter>] [-BypassModerationCheck
<SwitchParameter>] [-Database <DatabaseIdParameter>] [-MailboxPlan
<MailboxPlanIdParameter>] [-SKUAssigned <$true | $false>] [-SKUCapability
<None | BPOS_S_Deskless | BPOS_S_Standard | BPOS_S_Enterprise |
BPOS_S_Archive | BPOS_L_Standard | BPOS_B_Standard | BPOS_B_CustomDomain |
BPOS_S_MidSize | BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn |
BPOS_S_EopPremiumAddOn | BPOS_Unmanaged | TOU_Signed | FederatedUser |
Partner_Managed | MasteredOnPremise | ResourceMailbox |
ExcludedFromBackSync | UMFeatureRestricted | RichCoexistence |
OrganizationCapabilityUMGrammar | OrganizationCapabilityUMDataStorage |
OrganizationCapabilityOABGen | OrganizationCapabilityGMGen |
OrganizationCapabilityClientExtensions | BEVDirLockdown |
OrganizationCapabilityUMGrammarReady | OrganizationCapabilityMailRouting |
OrganizationCapabilityManagement | OrganizationCapabilityTenantUpgrade |
OrganizationCapabilityScaleOut | OrganizationCapabilityMessageTracking |
OrganizationCapabilityPstProvider |
OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-TargetAllMDBs <SwitchParameter>] [-
UsageLocation <CountryInfo>] <COMMON PARAMETERS>


Enable-Mailbox -PublicFolder <SwitchParameter> [-Database
<DatabaseIdParameter>] [-HoldForMigration <SwitchParameter>] <COMMON
PARAMETERS>


Enable-Mailbox -Discovery <SwitchParameter> [-Database
```

<DatabaseIdParameter>] [-TargetAllMDBs <SwitchParameter>] <COMMON
PARAMETERS>

Enable-Mailbox -Equipment <SwitchParameter> [-AccountDisabled <$true |
$false>] [-BypassModerationCheck <SwitchParameter>] [-Database
<DatabaseIdParameter>] [-TargetAllMDBs <SwitchParameter>] <COMMON
PARAMETERS>

Enable-Mailbox -LinkedDomainController <String> -LinkedMasterAccount
<UserIdParameter> [-Database <DatabaseIdParameter>] [-LinkedCredential
<PSCredential>] [-TargetAllMDBs <SwitchParameter>] <COMMON PARAMETERS>

Enable-Mailbox -Room <SwitchParameter> [-AccountDisabled <$true | $false>]
[-BypassModerationCheck <SwitchParameter>] [-Database
<DatabaseIdParameter>] [-TargetAllMDBs <SwitchParameter>] <COMMON
PARAMETERS>

Enable-Mailbox -LinkedDomainController <String> -LinkedMasterAccount
<UserIdParameter> -LinkedRoom <SwitchParameter> [-Database
<DatabaseIdParameter>] [-LinkedCredential <PSCredential>] [-TargetAllMDBs
<SwitchParameter>] <COMMON PARAMETERS>

Enable-Mailbox -Shared <SwitchParameter> [-AccountDisabled <$true |
$false>] [-BypassModerationCheck <SwitchParameter>] [-Database
<DatabaseIdParameter>] [-TargetAllMDBs <SwitchParameter>] <COMMON
PARAMETERS>

Enable-Mailbox -Arbitration <SwitchParameter> [-Database
<DatabaseIdParameter>] [-TargetAllMDBs <SwitchParameter>] <COMMON
PARAMETERS>

Enable-Mailbox [-AccountDisabled <$true | $false>] [-AddOnSKUCapability
<MultiValuedProperty>] [-BypassModerationCheck <SwitchParameter>] [-
Database <DatabaseIdParameter>] [-MailboxPlan <MailboxPlanIdParameter>] [-
SKUAssigned <$true | $false>] [-SKUCapability <None | BPOS_S_Deskless |
BPOS_S_Standard | BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard |
BPOS_B_Standard | BPOS_B_CustomDomain | BPOS_S_MidSize |
BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn |
BPOS_Unmanaged | TOU_Signed | FederatedUser | Partner_Managed |
MasteredOnPremise | ResourceMailbox | ExcludedFromBackSync |
UMFeatureRestricted | RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-TargetAllMDBs <SwitchParameter>] [-
UsageLocation <CountryInfo>] <COMMON PARAMETERS>

Enable-Mailbox [-Archive <SwitchParameter>] [-ArchiveDatabase
<DatabaseIdParameter>] [-ArchiveGuid <Guid>] [-ArchiveName
<MultiValuedProperty>] [-BypassModerationCheck <SwitchParameter>] <COMMON
PARAMETERS>

Enable-Mailbox -ArchiveDomain <SmtpDomain> [-RemoteArchive
<SwitchParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: -Identity <UserIdParameter> [-ActiveSyncMailboxPolicy
<MailboxPolicyIdParameter>] [-Alias <String>] [-Confirm
[<SwitchParameter>]] [-DisplayName <String>] [-DomainController <Fqdn>] [-
Force <SwitchParameter>] [-IncludeSoftDeletedObjects <SwitchParameter>] [-
ManagedFolderMailboxPolicy <MailboxPolicyIdParameter>] [-
ManagedFolderMailboxPolicyAllowed <SwitchParameter>] [-
OverrideRecipientQuotas <SwitchParameter>] [-PrimarySmtpAddress

```
<SmtpAddress>] [-RetentionPolicy <MailboxPolicyIdParameter>] [-
RoleAssignmentPolicy <MailboxPolicyIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a mailbox for the existing user Ayla. The mailbox is created in Database01.

```
Enable-Mailbox -Identity Contoso\Ayla -Database Database01
```

EXAMPLE 2

This example creates a remote archive for the existing user Ayla. The archive is created with the following settings:

- The archive database has the GUID 82025f12-8000-4d5e-8059-c052f9355125.
- The archive domain is archive.contoso.com.

```
Enable-Mailbox -Identity ayla@contoso.com -RemoteArchive -
ArchiveDatabase "82025f12-8000-4d5e-8059-c052f9355125" -
ArchiveDomain "archive.contoso.com"
```

Detailed Description

Mailbox-enabling an existing user or **InetOrgPerson** object creates additional mailbox attributes on the user object in Active Directory. When the user logs on to the mailbox or receives email messages, a mailbox object in the Exchange database is created.

Use the *Identity* parameter to specify the user or **InetOrgPerson** object for whom the mailbox is enabled. Use the *Database* parameter to specify the Exchange database that contains the mailbox.

Important:
When mailbox-enabling an existing user, if an alias isn't specified, Exchange uses the name and converts all non-ASCII characters to question mark (?) characters. In some languages that use nonstandard character sets, the user account may have a non-ASCII value for the name. In this case, when you mailbox-enable the user, the alias is changed to all question mark characters. To avoid this, confirm that the user account has an ASCII name before you create the new mailbox, or make sure you specify a value for the alias.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Arbitration	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Arbitration* parameter specifies that the mailbox for which you are executing the command is an arbitration mailbox. Arbitration mailboxes are used for managing approval workflow. For example, an arbitration mailbox is used for handling moderated recipients and distribution group membership approval.
ArchiveDomain	Required	Microsoft.Exchange.Data.SmtpDomain	This parameter is available only in on-premises Exchange 2013. The *ArchiveDomain* parameter specifies the cloud-based domain on which the archive associated with this mailbox exists. For example, if the SMTP email address of the user is tony@contoso.com, the

			SMTP domain could be archive.contoso.com. **Note:** Only use this parameter if the archive is hosted in a cloud-based service.
Discovery	Required	System.Management.Automation.SwitchParameter	The *Discovery* parameter specifies that this mailbox is a Discovery mailbox. Discovery mailboxes are created as target mailboxes for Discovery searches. You don't have to include a value with this parameter. After being created or enabled, a Discovery mailbox can't be converted to another type of mailbox. For more information, see In-Place eDiscovery.
Equipment	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Equipment* parameter specifies that the type of resource is equipment, if this mailbox is a resource mailbox. This parameter is required only if you're enabling a resource

			mailbox.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	The *Identity* parameter specifies the user or **InetOrgPerson** object that you want to mailbox-enable. You can use one of the following values: • GUID • Distinguished name (DN) • Display name • *Domain\Account* • User principal name (UPN)
LinkedDomainController	Required	System.String	This parameter is available only in on-premises Exchange 2013. The *LinkedDomainController* parameter specifies the domain controller in the forest where the user account resides. The domain controller in this forest is used to get security information for the account specified by the *LinkedMasterAccount* parameter. This parameter is required only if you're creating a linked mailbox. Use the fully qualified

			domain name (FQDN) of the domain controller you want to use as the value for this parameter.
LinkedMasterAccount	Required	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	This parameter is available only in on-premises Exchange 2013. The *LinkedMasterAccount* parameter specifies the master account in the forest where the user account resides. The master account is the account to link the mailbox to. The master account grants access to the mailbox. This parameter is required only if you're creating a linked mailbox. You can use one of the following values: • GUID • DN • *Domain\Account* • UPN • LegacyExchangeDN • SmtpAddress • Alias
LinkedRoom	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013.

			The *LinkedRoom* parameter is used to specify that the mailbox being enabled is a linked resource mailbox. A linked resource mailbox is useful in a scenario where you have an account in an authentication forest and you want it to be directly linked to a resource mailbox in resource forest.
PublicFolder	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *PublicFolder* parameter specifies that the mailbox to enable is a public folder mailbox. Public folder mailboxes are specially designed mailboxes to store the hierarchy and content of public folders. This parameter is required if you're enabling a public folder mailbox.
Room	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Room* parameter specifies that the type of

			resource is a room, if this mailbox is a resource mailbox. This parameter is required only if you're enabling a resource mailbox. You don't have to specify a value with this parameter.
Shared	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Shared* parameter specifies that you're creating a shared mailbox. A shared mailbox is a mailbox to which multiple users can log on. This mailbox isn't associated with any of the users that can log on. It's associated with a disabled user account. This parameter is required only if you're creating a shared mailbox.
AccountDisabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ActiveSyncMailboxPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	This parameter is available only in on-premises Exchange 2013.

			The *ActiveSyncMailboxPolicy* parameter specifies the mailbox policy to enable for the mailbox that you create. If you don't specify this parameter, the default mailbox policy is used.
AddOnSKUCapability	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
AddressBookPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.AddressBookMailboxPolicyIdParameter	This parameter is available only in on-premises Exchange 2013. The *AddressBookPolicy* parameter specifies the address book policy to apply to this mailbox. For more information about address book policies, see Address book policies.
Alias	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *Alias* parameter specifies the email alias of the mailbox that you're enabling. The alias can be a combination of characters separated by a period

			with no intervening spaces. Don't use special characters in the alias.
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* parameter specifies that when this mailbox is enabled, an archive will be created.
ArchiveDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArchiveDatabase* parameter specifies the Exchange database that contains the archive associated with this mailbox. You can use the following values: • GUID of the database • Database name
ArchiveGuid	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
ArchiveName	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ArchiveName* parameter specifies the name of the archive mailbox. This is the name displayed to users in Microsoft Office Outlook Web App and Microsoft Outlook. The default name is Online Archive - *<Mailbox User's Display*

			Name>.
BypassModerationCheck	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Database	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *Database* parameter specifies which Exchange database contains the new mailbox. You can use one of the following values: • GUID of the database • Database name
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the display name for the mailbox. The *DisplayName* is the name that appears in the Exchange Administration

			Center.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* parameter specifies whether to suppress warning or confirmation messages. This parameter can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* parameter isn't provided in the command, administrative input is prompted. If the *Force* parameter is provided in the command, but the value is omitted, its default value is `$true`.
HoldForMigration	Optional	System.Management.	This parameter is

		Automation.SwitchParameter	available only in on-premises Exchange 2013. The *HoldForMigration* parameter specifies whether to prevent any client or user, except the Microsoft Exchange Mailbox Replication service (MRS) process, from logging into a public folder mailbox. You must use this parameter when you create the first public folder, which is called the *hierarchy mailbox*, in your organization.
IncludeSoftDeletedObjects	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
LinkedCredential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *LinkedCredential* parameter specifies credentials to use to access the domain controller specified by the *LinkedDomainController* parameter. This parameter is optional, even if you're enabling a linked mailbox.

MailboxPlan	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPlanIdParameter	This parameter is reserved for internal Microsoft use.
ManagedFolderMailboxPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	This parameter is available only in on-premises Exchange 2013. The *ManagedFolderMailboxPolicy* parameter specifies the managed folder mailbox policy to enable for the mailbox that you create. If you don't specify this parameter, the default managed folder mailbox policy is used.
ManagedFolderMailboxPolicyAllowed	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ManagedFolderMailboxPolicyAllowed* parameter specifies whether to bypass the warning that messaging records management (MRM) features aren't supported for email clients using versions of Outlook earlier than Office Outlook 2007. When a managed folder mailbox

			policy is assigned to a mailbox using the *ManagedFolderMailboxPolicy* parameter, the warning appears by default unless the *ManagedFolderMailboxPolicyAllowed* parameter is used. **Note:** Outlook 2003 Service Pack 3 clients are supported but are provided limited functionality for MRM.
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in on-premises Exchange 2013. The *PrimarySmtpAddress* parameter specifies the primary SMTP address for the mailbox. By default, the primary SMTP address is generated based on the default email address policy. If you specify a primary SMTP address by using this parameter, the command sets the **EmailAddressPolicyEna**

			bled attribute of the mailbox to `$false`, and the email addresses of this mailbox aren't automatically updated based on email address policies.
RemoteArchive	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemoteArchive* parameter specifies that when the mailbox is enabled, a remote archive for this mailbox will be created. A remote archive exists in a cloud-based service. You don't have to specify a value with this parameter. You can't use the *Archive* parameter in conjunction with this parameter.
RetentionPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	This parameter is available only in on-premises Exchange 2013. The *RetentionPolicy* parameter specifies the name of a retention policy that you want applied to this mailbox. Retention

			policies consist of tags that are applied to mailbox folders and mail items to determine the period of time that the items should be retained.
RoleAssignmentPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *RoleAssignmentPolicy* parameter specifies the management role assignment policy to assign to the mailbox when it's created or enabled. If you don't include this parameter when you create or enable a mailbox, the default assignment policy is used. If the assignment policy name contains spaces, enclose the name in quotation marks ("). If you don't want to assign an assignment policy when a mailbox is created or enabled, specify a value of `$null`. For more information about assignment policies, see Understanding management role assignment policies.
SKUAssigned	Optional	System.Boolean	This parameter is reserved

			for internal Microsoft use.
SKUCapability	Optional	Microsoft.Exchange.Data.Directory.Capability	This parameter is reserved for internal Microsoft use.
TargetAllMDBs	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
UsageLocation	Optional	Microsoft.Exchange.Data.Directory.CountryInfo	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-Mailbox

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-Mailbox** cmdlet to view mailbox objects and attributes, populate property pages, or supply mailbox information to other tasks.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-Mailbox [-Identity <MailboxIdParameter>] <COMMON PARAMETERS>

Get-Mailbox [-MailboxPlan <MailboxPlanIdParameter>] <COMMON PARAMETERS>

Get-Mailbox [-Server <ServerIdParameter>] <COMMON PARAMETERS>

Get-Mailbox [-Database <DatabaseIdParameter>] <COMMON PARAMETERS>

Get-Mailbox [-Anr <String>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
Arbitration <SwitchParameter>] [-Archive <SwitchParameter>] [-AuxMailbox
<SwitchParameter>] [-Credential <PSCredential>] [-DomainController <Fqdn>]
[-Filter <String>] [-IgnoreDefaultScope <SwitchParameter>] [-
InactiveMailboxOnly <SwitchParameter>] [-IncludeInactiveMailbox
<SwitchParameter>] [-IncludeSoftDeletedMailbox <SwitchParameter>] [-
Monitoring <SwitchParameter>] [-Organization <OrganizationIdParameter>] [-
OrganizationalUnit <OrganizationalUnitIdParameter>] [-PublicFolder
<SwitchParameter>] [-ReadFromDomainController <SwitchParameter>] [-
RecipientTypeDetails <RecipientTypeDetails[]>] [-RemoteArchive
<SwitchParameter>] [-ResultSize <Unlimited>] [-SoftDeletedMailbox
<SwitchParameter>] [-SortBy <String>] [-UsnForReconciliationSearch
<Int64>]
```

Examples

EXAMPLE 1

This example returns a list of all the mailboxes in your organization.

```
Get-Mailbox -ResultSize unlimited
```

EXAMPLE 2

This example returns a list of all the mailboxes in your organization in the Users OU.

```
Get-Mailbox -OrganizationalUnit Users
```

EXAMPLE 3

This example returns all the mailboxes that resolve from the ambiguous name resolution search on the string "Chr" that are in the domain named DC01. This example returns mailboxes for users such as Chris Ashton, Christian Hess, and Christa Geller.

```
Get-Mailbox -Anr Chr -DomainController DC01
```

EXAMPLE 4

This example returns information about the mailbox named Chris, including archive mailbox information.

```
Get-Mailbox -Identity Chris -Archive
```

EXAMPLE 5

This example returns information about the mailbox ed@contoso.com, including information about his remote archive mailbox.

```
Get-Mailbox -Identity ed@contoso.com -RemoteArchive
```

Detailed Description

The **Get-Mailbox** cmdlet retrieves the attributes and objects for a mailbox. No parameters are required. If the cmdlet is used without a parameter, all mailboxes in the organization are listed.

Note:

To accurately evaluate the current storage quota status using the **Get-Mailbox** cmdlet, you must look at the **UseDatabaseQuotaDefaults** property in addition to the **ProhibitSendQuota**, **ProhibitSendReceiveQuota**, and **IssueWarningQuota** properties. A value of True for the **UseDatabaseQuotaDefaults** property means that the per-mailbox settings are ignored and the mailbox database limits are used. If this property is set to True and the **ProhibitSendQuota**, **ProhibitSendReceiveQuota**, and **IssueWarningQuota** properties are set to unlimited, the mailbox doesn't have unlimited size. Instead, you must reference the mailbox database storage limits to see what the limits for the mailbox are. A value of False for the **UseDatabaseQuotaDefaults** property means that the per-mailbox settings are used.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Arbitration	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Arbitration* parameter specifies that the mailbox for which you are executing the command is an arbitration mailbox. Arbitration mailboxes are used for managing

			approval workflow. For example, an arbitration mailbox is used for handling moderated recipients and distribution group membership approval.
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* parameter specifies whether to return information about the recipient's archive mailbox.
AuxMailbox	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.

Database	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *Database* parameter specifies the database from which to get the mailbox. You can use the following values: • GUID of the database • Database name This parameter can't be used with the *Filter* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.

Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identity of the mailbox. You can use one of the following values: • GUID • Distinguished name (DN) • Display name • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the

			following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
InactiveMailboxOnly	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *InactiveMailboxOnly* parameter specifies the command to return only inactive mailboxes. An inactive mailbox is a mailbox that has been removed or soft-deleted. An inactive mailbox can be recovered for up to 30 days after it's removed.

			This parameter is available only in the cloud-based service.
IncludeInactiveMailbox	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *IncludeInactiveMailbox* parameter specifies the command to return both active and inactive mailboxes. An inactive mailbox is a mailbox that has been removed or soft-deleted. An inactive mailbox can be recovered for up to 30 days after it's removed. This parameter is available only in the cloud-based service.
IncludeSoftDeletedMailbox	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
MailboxPlan	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPlanIdParameter	This parameter is available only in the cloud-based service. The *MailboxPlan* parameter specifies the command to return mailboxes associated with

			this mailbox plan. A mailbox plan specifies the permissions and features available to a mailbox user. The mailbox plan name you provide must be included in the service plan for the organization in which this mailbox belongs.
Monitoring	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Monitoring* parameter specifies a list of mailboxes that have a value of `MonitoringMailbox` for the *RecipientTypeDetails* property. Two monitoring mailboxes are automatically created for each mailbox database in your organization: one for monitoring the health of public folders and one for monitoring the health of site mailboxes.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies an organizational unit (OU) and is used to limit the results. If you use this parameter, you only get mailboxes in the container that you specify. You can use either the OU or the domain name. If you use the OU, you must specify the canonical name of the OU.
PublicFolder	Optional	System.Management.Automation.SwitchParameter	The *PublicFolder* parameter specifies that the mailbox is a public folder mailbox. Public folder mailboxes are specially designed mailboxes to store the hierarchy and content of public folders. This parameter is required to display information about a public folder mailbox.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information

			is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest and don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Exchange.
RecipientTypeDetails	Optional	Microsoft.Exchange.Data.Directory.Recipient.RecipientTypeDetails[]	The *RecipientTypeDetails* parameter specifies the type of recipients returned. Recipient types are divided into recipient types and subtypes. Each recipient type contains all common properties for all subtypes. For example, the type `UserMailbox` represents a user account in Active Directory that has an associated mailbox. Because there are several mailbox types, each

			mailbox type is identified by the *RecipientTypeDetails* parameter. For example, a conference room mailbox has *RecipientTypeDetails* set to `RoomMailbox`, whereas a user mailbox has *RecipientTypeDetails* set to `UserMailbox`. This parameter accepts the following values: • `RoomMailbox` • `EquipmentMailbox` • `LegacyMailbox` • `LinkedMailbox` • `UserMailbox` • `DiscoveryMailbox` • `SharedMailbox`
RemoteArchive	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemoteArchive* parameter specifies whether to disconnect the remote archive for this mailbox. A remote archive exists in a cloud-based service. When you use this parameter, you can't use the *Archive* parameter.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum

			number of results to return. If you want to return all mailboxes that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	This parameter is available only in on-premises Exchange 2013. The *Server* parameter specifies an individual server and is used to limit the results. If you use this parameter, you only get mailboxes that reside on the server that you specify. Use the common name of the server that you want to specify.
SoftDeletedMailbox	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *SoftDeletedMailbox* parameter specifies a list of deleted mailboxes that were deleted within the last 30 days.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results.

			You can sort by only one attribute at a time. You can sort by the following attributes: • Alias • Display name • Name The results are sorted in ascending order.
UsnForReconciliationSearch	Optional	System.Int64	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-Mailbox

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-22*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-Mailbox** cmdlet to create a user in Active Directory and mailbox-enable this new user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-Mailbox -Password <SecureString> -UserPrincipalName <String> [-
```

```
AddOnSKUCapability <MultiValuedProperty>] [-ArbitrationMailbox
<MailboxIdParameter>] [-MailboxPlan <MailboxPlanIdParameter>] [-
ModeratedBy <MultiValuedProperty>] [-ModerationEnabled <$true | $false>]
[-RemovedMailbox <RemovedMailboxIdParameter>] [-
SendModerationNotifications <Never | Internal | Always>] [-SKUAssigned
<$true | $false>] [-SKUCapability <None | BPOS_S_Deskless |
BPOS_S_Standard | BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard |
BPOS_B_Standard | BPOS_B_CustomDomain | BPOS_S_MidSize |
BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn |
BPOS_Unmanaged | TOU_Signed | FederatedUser | Partner_Managed |
MasteredOnPremise | ResourceMailbox | ExcludedFromBackSync |
UMFeatureRestricted | RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-UsageLocation <CountryInfo>] <COMMON
PARAMETERS>


New-Mailbox -Room <SwitchParameter> [-ArbitrationMailbox
<MailboxIdParameter>] [-ModeratedBy <MultiValuedProperty>] [-
ModerationEnabled <$true | $false>] [-Office <String>] [-Password
<SecureString>] [-Phone <String>] [-ResourceCapacity <Int32>] [-
SendModerationNotifications <Never | Internal | Always>] [-
UserPrincipalName <String>] <COMMON PARAMETERS>


New-Mailbox -LinkedDomainController <String> -LinkedMasterAccount
<UserIdParameter> -LinkedRoom <SwitchParameter> [-ArbitrationMailbox
<MailboxIdParameter>] [-LinkedCredential <PSCredential>] [-ModeratedBy
<MultiValuedProperty>] [-ModerationEnabled <$true | $false>] [-Office
<String>] [-Password <SecureString>] [-Phone <String>] [-ResourceCapacity
<Int32>] [-SendModerationNotifications <Never | Internal | Always>] [-
UserPrincipalName <String>] <COMMON PARAMETERS>


New-Mailbox -ImportLiveId <SwitchParameter> -WindowsLiveID <WindowsLiveId>
[-AddOnSKUCapability <MultiValuedProperty>] [-ArbitrationMailbox
<MailboxIdParameter>] [-MailboxPlan <MailboxPlanIdParameter>] [-
ModeratedBy <MultiValuedProperty>] [-ModerationEnabled <$true | $false>]
[-RemovedMailbox <RemovedMailboxIdParameter>] [-
SendModerationNotifications <Never | Internal | Always>] [-SKUAssigned
<$true | $false>] [-SKUCapability <None | BPOS_S_Deskless |
BPOS_S_Standard | BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard |
BPOS_B_Standard | BPOS_B_CustomDomain | BPOS_S_MidSize |
BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn |
BPOS_Unmanaged | TOU_Signed | FederatedUser | Partner_Managed |
MasteredOnPremise | ResourceMailbox | ExcludedFromBackSync |
UMFeatureRestricted | RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-UsageLocation <CountryInfo>] <COMMON
PARAMETERS>


New-Mailbox -AccountDisabled <SwitchParameter> [-AddOnSKUCapability
<MultiValuedProperty>] [-ArbitrationMailbox <MailboxIdParameter>] [-
MailboxPlan <MailboxPlanIdParameter>] [-ModeratedBy <MultiValuedProperty>]
[-ModerationEnabled <$true | $false>] [-Password <SecureString>] [-
SendModerationNotifications <Never | Internal | Always>] [-SKUAssigned
<$true | $false>] [-SKUCapability <None | BPOS_S_Deskless |
BPOS_S_Standard | BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard |
BPOS_B_Standard | BPOS_B_CustomDomain | BPOS_S_MidSize |
BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn |
BPOS_Unmanaged | TOU_Signed | FederatedUser | Partner_Managed |
MasteredOnPremise | ResourceMailbox | ExcludedFromBackSync |
UMFeatureRestricted | RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
```

```
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-UsageLocation <CountryInfo>] [-
UserPrincipalName <String>] <COMMON PARAMETERS>


New-Mailbox -MicrosoftOnlineServicesID <WindowsLiveId> -Password
<SecureString> [-AddOnSKUCapability <MultiValuedProperty>] [-
ArbitrationMailbox <MailboxIdParameter>] [-MailboxPlan
<MailboxPlanIdParameter>] [-ModeratedBy <MultiValuedProperty>] [-
ModerationEnabled <$true | $false>] [-RemovedMailbox
<RemovedMailboxIdParameter>] [-SendModerationNotifications <Never |
Internal | Always>] [-SKUAssigned <$true | $false>] [-SKUCapability <None
| BPOS_S_Deskless | BPOS_S_Standard | BPOS_S_Enterprise | BPOS_S_Archive |
BPOS_L_Standard | BPOS_B_Standard | BPOS_B_CustomDomain | BPOS_S_MidSize |
BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn |
BPOS_Unmanaged | TOU_Signed | FederatedUser | Partner_Managed |
MasteredOnPremise | ResourceMailbox | ExcludedFromBackSync |
UMFeatureRestricted | RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-UsageLocation <CountryInfo>] <COMMON
PARAMETERS>


New-Mailbox -UserPrincipalName <String> [-AddOnSKUCapability
<MultiValuedProperty>] [-ArbitrationMailbox <MailboxIdParameter>] [-
MailboxPlan <MailboxPlanIdParameter>] [-ModeratedBy <MultiValuedProperty>]
[-ModerationEnabled <$true | $false>] [-SendModerationNotifications <Never
| Internal | Always>] [-SKUAssigned <$true | $false>] [-SKUCapability
<None | BPOS_S_Deskless | BPOS_S_Standard | BPOS_S_Enterprise |
BPOS_S_Archive | BPOS_L_Standard | BPOS_B_Standard | BPOS_B_CustomDomain |
BPOS_S_MidSize | BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn |
BPOS_S_EopPremiumAddOn | BPOS_Unmanaged | TOU_Signed | FederatedUser |
Partner_Managed | MasteredOnPremise | ResourceMailbox |
ExcludedFromBackSync | UMFeatureRestricted | RichCoexistence |
OrganizationCapabilityUMGrammar | OrganizationCapabilityUMDataStorage |
OrganizationCapabilityOABGen | OrganizationCapabilityGMGen |
OrganizationCapabilityClientExtensions | BEVDirLockdown |
OrganizationCapabilityUMGrammarReady | OrganizationCapabilityMailRouting |
OrganizationCapabilityManagement | OrganizationCapabilityTenantUpgrade |
OrganizationCapabilityScaleOut | OrganizationCapabilityMessageTracking |
OrganizationCapabilityPstProvider |
OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] <COMMON PARAMETERS>


New-Mailbox -FederatedIdentity <String> -WindowsLiveID <WindowsLiveId> [-
AddOnSKUCapability <MultiValuedProperty>] [-EvictLiveId <SwitchParameter>]
[-MailboxPlan <MailboxPlanIdParameter>] [-NetID <NetID>] [-RemovedMailbox
<RemovedMailboxIdParameter>] [-SKUAssigned <$true | $false>] [-
SKUCapability <None | BPOS_S_Deskless | BPOS_S_Standard |
BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard | BPOS_B_Standard |
BPOS_B_CustomDomain | BPOS_S_MidSize | BPOS_S_ArchiveAddOn |
BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn | BPOS_Unmanaged |
TOU_Signed | FederatedUser | Partner_Managed | MasteredOnPremise |
ResourceMailbox | ExcludedFromBackSync | UMFeatureRestricted |
RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
```

```
OrganizationCapabilityMigration>] [-UsageLocation <CountryInfo>] <COMMON
PARAMETERS>


New-Mailbox -UseExistingLiveId <SwitchParameter> -WindowsLiveID
<WindowsLiveId> [-AddOnSKUCapability <MultiValuedProperty>] [-
ArbitrationMailbox <MailboxIdParameter>] [-BypassLiveId <SwitchParameter>]
[-MailboxPlan <MailboxPlanIdParameter>] [-ModeratedBy
<MultiValuedProperty>] [-ModerationEnabled <$true | $false>] [-NetID
<NetID>] [-RemovedMailbox <RemovedMailboxIdParameter>] [-
SendModerationNotifications <Never | Internal | Always>] [-SKUAssigned
<$true | $false>] [-SKUCapability <None | BPOS_S_Deskless |
BPOS_S_Standard | BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard |
BPOS_B_Standard | BPOS_B_CustomDomain | BPOS_S_MidSize |
BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn |
BPOS_Unmanaged | TOU_Signed | FederatedUser | Partner_Managed |
MasteredOnPremise | ResourceMailbox | ExcludedFromBackSync |
UMFeatureRestricted | RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-UsageLocation <CountryInfo>] <COMMON
PARAMETERS>


New-Mailbox -ArchiveDomain <SmtpDomain> -Password <SecureString> -
UserPrincipalName <String> [-AddOnSKUCapability <MultiValuedProperty>] [-
ArbitrationMailbox <MailboxIdParameter>] [-MailboxPlan
<MailboxPlanIdParameter>] [-ModeratedBy <MultiValuedProperty>] [-
ModerationEnabled <$true | $false>] [-RemoteArchive <SwitchParameter>] [-
RemovedMailbox <RemovedMailboxIdParameter>] [-SendModerationNotifications
<Never | Internal | Always>] [-SKUAssigned <$true | $false>] [-
SKUCapability <None | BPOS_S_Deskless | BPOS_S_Standard |
BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard | BPOS_B_Standard |
BPOS_B_CustomDomain | BPOS_S_MidSize | BPOS_S_ArchiveAddOn |
BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn | BPOS_Unmanaged |
TOU_Signed | FederatedUser | Partner_Managed | MasteredOnPremise |
ResourceMailbox | ExcludedFromBackSync | UMFeatureRestricted |
RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] <COMMON PARAMETERS>


New-Mailbox -RemovedMailbox <RemovedMailboxIdParameter> [-
AddOnSKUCapability <MultiValuedProperty>] [-ArbitrationMailbox
<MailboxIdParameter>] [-MailboxPlan <MailboxPlanIdParameter>] [-
ModeratedBy <MultiValuedProperty>] [-ModerationEnabled <$true | $false>]
[-Password <SecureString>] [-SendModerationNotifications <Never | Internal
| Always>] [-SKUAssigned <$true | $false>] [-SKUCapability <None |
BPOS_S_Deskless | BPOS_S_Standard | BPOS_S_Enterprise | BPOS_S_Archive |
BPOS_L_Standard | BPOS_B_Standard | BPOS_B_CustomDomain | BPOS_S_MidSize |
BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn |
BPOS_Unmanaged | TOU_Signed | FederatedUser | Partner_Managed |
MasteredOnPremise | ResourceMailbox | ExcludedFromBackSync |
UMFeatureRestricted | RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] <COMMON PARAMETERS>
```

```
New-Mailbox -FederatedIdentity <String> -MicrosoftOnlineServicesID
<WindowsLiveId> [-AddOnSKUCapability <MultiValuedProperty>] [-MailboxPlan
<MailboxPlanIdParameter>] [-NetID <NetID>] [-RemovedMailbox
<RemovedMailboxIdParameter>] [-SKUAssigned <$true | $false>] [-
SKUCapability <None | BPOS_S_Deskless | BPOS_S_Standard |
BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard | BPOS_B_Standard |
BPOS_B_CustomDomain | BPOS_S_MidSize | BPOS_S_ArchiveAddOn |
BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn | BPOS_Unmanaged |
TOU_Signed | FederatedUser | Partner_Managed | MasteredOnPremise |
ResourceMailbox | ExcludedFromBackSync | UMFeatureRestricted |
RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-UsageLocation <CountryInfo>] <COMMON
PARAMETERS>


New-Mailbox -Password <SecureString> -WindowsLiveID <WindowsLiveId> [-
AddOnSKUCapability <MultiValuedProperty>] [-ArbitrationMailbox
<MailboxIdParameter>] [-EvictLiveId <SwitchParameter>] [-MailboxPlan
<MailboxPlanIdParameter>] [-ModeratedBy <MultiValuedProperty>] [-
ModerationEnabled <$true | $false>] [-RemovedMailbox
<RemovedMailboxIdParameter>] [-SendModerationNotifications <Never |
Internal | Always>] [-SKUAssigned <$true | $false>] [-SKUCapability <None
| BPOS_S_Deskless | BPOS_S_Standard | BPOS_S_Enterprise | BPOS_S_Archive |
BPOS_L_Standard | BPOS_B_Standard | BPOS_B_CustomDomain | BPOS_S_MidSize |
BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn |
BPOS_Unmanaged | TOU_Signed | FederatedUser | Partner_Managed |
MasteredOnPremise | ResourceMailbox | ExcludedFromBackSync |
UMFeatureRestricted | RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-UsageLocation <CountryInfo>] <COMMON
PARAMETERS>


New-Mailbox -Discovery <SwitchParameter> [-Password <SecureString>] [-
UserPrincipalName <String>] <COMMON PARAMETERS>


New-Mailbox -Equipment <SwitchParameter> [-ArbitrationMailbox
<MailboxIdParameter>] [-ModeratedBy <MultiValuedProperty>] [-
ModerationEnabled <$true | $false>] [-Password <SecureString>] [-
SendModerationNotifications <Never | Internal | Always>] [-
UserPrincipalName <String>] <COMMON PARAMETERS>


New-Mailbox -Shared <SwitchParameter> [-ArbitrationMailbox
<MailboxIdParameter>] [-ModeratedBy <MultiValuedProperty>] [-
ModerationEnabled <$true | $false>] [-Password <SecureString>] [-
SendModerationNotifications <Never | Internal | Always>] [-
UserPrincipalName <String>] <COMMON PARAMETERS>


New-Mailbox -Arbitration <SwitchParameter> -UserPrincipalName <String> [-
Password <SecureString>] <COMMON PARAMETERS>


New-Mailbox [-ArbitrationMailbox <MailboxIdParameter>] [-ModeratedBy
<MultiValuedProperty>] [-ModerationEnabled <$true | $false>] [-Password
<SecureString>] [-SendModerationNotifications <Never | Internal | Always>]
[-UserPrincipalName <String>] <COMMON PARAMETERS>


New-Mailbox [-AuxMailbox <SwitchParameter>] [-Password <SecureString>] [-
UserPrincipalName <String>] <COMMON PARAMETERS>
```

```
New-Mailbox [-ArbitrationMailbox <MailboxIdParameter>] [-ModeratedBy
<MultiValuedProperty>] [-ModerationEnabled <$true | $false>] [-Password
<SecureString>] [-SendModerationNotifications <Never | Internal | Always>]
[-UserPrincipalName <String>] <COMMON PARAMETERS>


New-Mailbox -LinkedDomainController <String> -LinkedMasterAccount
<UserIdParameter> [-ArbitrationMailbox <MailboxIdParameter>] [-
LinkedCredential <PSCredential>] [-ModeratedBy <MultiValuedProperty>] [-
ModerationEnabled <$true | $false>] [-Password <SecureString>] [-
SendModerationNotifications <Never | Internal | Always>] [-
UserPrincipalName <String>] <COMMON PARAMETERS>


New-Mailbox [-ArbitrationMailbox <MailboxIdParameter>] [-ModeratedBy
<MultiValuedProperty>] [-ModerationEnabled <$true | $false>] [-Password
<SecureString>] [-SendModerationNotifications <Never | Internal | Always>]
[-UserPrincipalName <String>] <COMMON PARAMETERS>


New-Mailbox -EnableRoomMailboxAccount <$true | $false> -Room
<SwitchParameter> [-MicrosoftOnlineServicesID <WindowsLiveId>] [-
RoomMailboxPassword <SecureString>] [-UserPrincipalName <String>] <COMMON
PARAMETERS>


New-Mailbox [-ArbitrationMailbox <MailboxIdParameter>] [-ModeratedBy
<MultiValuedProperty>] [-SendModerationNotifications <Never | Internal |
Always>] <COMMON PARAMETERS>


New-Mailbox -PublicFolder <SwitchParameter> [-HoldForMigration
<SwitchParameter>] [-IsExcludedFromServingHierarchy <$true | $false>]
<COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-ActiveSyncMailboxPolicy
<MailboxPolicyIdParameter>] [-AddressBookPolicy
<AddressBookMailboxPolicyIdParameter>] [-Alias <String>] [-Archive
<SwitchParameter>] [-ArchiveDatabase <DatabaseIdParameter>] [-Confirm
[<SwitchParameter>]] [-Database <DatabaseIdParameter>] [-DisplayName
<String>] [-DomainController <Fqdn>] [-ExternalDirectoryObjectId <String>]
[-FirstName <String>] [-Force <SwitchParameter>] [-
ForestWideDomainControllerAffinityByExecutingUser <SwitchParameter>] [-
ImmutableId <String>] [-Initials <String>] [-Languages
<MultiValuedProperty>] [-LastName <String>] [-MailboxContainerGuid <Guid>]
[-MailboxProvisioningConstraint <MailboxProvisioningConstraint>] [-
MailboxProvisioningPreferences <MultiValuedProperty>] [-
ManagedFolderMailboxPolicy <MailboxPolicyIdParameter>] [-
ManagedFolderMailboxPolicyAllowed <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-OriginalNetID <NetID>] [-
OverrideRecipientQuotas <SwitchParameter>] [-PrimarySmtpAddress
<SmtpAddress>] [-QueryBaseDNRestrictionEnabled <$true | $false>] [-
RemoteAccountPolicy <RemoteAccountPolicyIdParameter>] [-
RemotePowerShellEnabled <$true | $false>] [-ResetPasswordOnNextLogon
<$true | $false>] [-RetentionPolicy <MailboxPolicyIdParameter>] [-
RoleAssignmentPolicy <MailboxPolicyIdParameter>] [-SamAccountName
<String>] [-SharingPolicy <SharingPolicyIdParameter>] [-TargetAllMDBs
<SwitchParameter>] [-ThrottlingPolicy <ThrottlingPolicyIdParameter>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a user Chris Ashton in Active Directory and creates a mailbox for the user. The mailbox is located on Mailbox Database 1. The password must be reset at the next logon. To set the initial value of the password, this example creates a variable (*$password*), prompts you to enter

a password, and assigns that password to the variable as a **SecureString** object.

```
$password = Read-Host "Enter password" -AsSecureString
New-Mailbox -UserPrincipalName chris@contoso.com -Alias
chris -Database "Mailbox Database 1" -Name ChrisAshton -
OrganizationalUnit Users -Password $password -FirstName
Chris -LastName Ashton -DisplayName "Chris Ashton" -
ResetPasswordOnNextLogon $true
```

EXAMPLE 2

This example creates a user in Active Directory and a resource mailbox for a conference room. The resource mailbox is located in Mailbox Database 1. The password must be reset at the next logon. The Exchange Management Shell prompts for the value of the initial password because it's not specified.

```
New-Mailbox -UserPrincipalName confmbx@contoso.com -Alias
confmbx -Name ConfRoomMailbox -Database "Mailbox Database
1" -OrganizationalUnit Users -Room -
ResetPasswordOnNextLogon $true
```

EXAMPLE 3

This example creates an enabled user account in Active Directory and a room mailbox for a conference room in an on-premises Exchange organization. The *RoomMailboxPassword* parameter specifies the password for the user account.

```
New-Mailbox -UserPrincipalName confroom1010@contoso.com -
Alias confroom1010 -Name "Conference Room 1010" -Room -
EnableRoomMailboxAccount $true -RoomMailboxPassword
(ConvertTo-SecureString -String P@ssw0rd -AsPlainText -
Force)
```

EXAMPLE 4

This example creates the shared mailbox "Sales Department" and grants *Full Access* and *Send on Behalf* permissions for the security group "MarketingSG". Users who are members of the security group will be granted the permissions to the mailbox.

Note:

This example assumes that you've already created a mail-enabled security group named "MarketingSG" by using the **New-DistributionGroup** cmdlet.

```
New-Mailbox -Shared -Name "Sales Department" -DisplayName
```

```
"Sales Department" -Alias Sales
Set-Mailbox -Identity Sales -GrantSendOnBehalfTo
MarketingSG
Add-MailboxPermission -Identity Sales -User MarketingSG -
AccessRights FullAccess -InheritanceType All
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountDisabled	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *AccountDisabled* parameter specifies whether to create the mailbox in a disabled state. You don't have to specify a value with this parameter.
Arbitration	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Arbitration* parameter specifies that the mailbox for which you are executing the command is an arbitration mailbox. Arbitration mailboxes are

			used for managing approval workflow. For example, an arbitration mailbox is used for handling moderated recipients and distribution group membership approval.
ArchiveDomain	Required	Microsoft.Exchange.Data.SmtpDomain	This parameter is available only in on-premises Exchange 2013. The *ArchiveDomain* parameter specifies the cloud-based services domain on which the archive associated with this mailbox exists. For example, if the SMTP email address of the user is tony@contoso.com, the SMTP domain could be archive.contoso.com.
Discovery	Required	System.Management.Automation.SwitchParameter	The *Discovery* parameter specifies that this mailbox is a Discovery mailbox. Discovery mailboxes are created as target mailboxes for Discovery searches. After being created or enabled, a Discovery mailbox can't be repurposed or

			converted to another type of mailbox. You don't have to include a value with this parameter. For more information, see In-Place eDiscovery.
EnableRoomMailboxAccount	Required	System.Boolean	Use the *EnableRoomMailboxAccount* parameter and the `$true` value to specify that the corresponding account in Active Directory for the new room mailbox is enabled when the mailbox is created. Both the *Room* and *RoomMailboxPassword* parameters are also required to create a logon-enabled room mailbox. When you create a room mailbox using only the *Room* parameter, the account in Active Directory is logon-disabled, which prevents users from signing in to the mailbox. When you include the *EnableRoomMailboxAcco*

			unt and *RoomMailboxPassword* parameters, the account in Active Directory is logon-enabled, which is required for implementing some scenarios, such as the Lync Room System. In Exchange Online, a logon-enabled room mailbox doesn't require a license.
Equipment	Required	System.Management.Automation.SwitchParameter	The *Equipment* parameter specifies that the type of resource is equipment, if this mailbox is a resource mailbox. This parameter is required only if you're creating a resource mailbox.
FederatedIdentity	Required	System.String	This parameter is available only in the cloud-based service. The *FederatedIdentity* parameter associates an on-premises Active Directory user with a Microsoft Office user.
ImportLiveId	Required	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *ImportLiveId*

			parameter imports an unmanaged Microsoft account (formerly known as a Windows Live ID) into the cloud-based domain. An unmanaged Microsoft account was created in the domain before the domain was enrolled in the cloud-based service. Importing a Microsoft account into the domain lets you save any settings associated with the Microsoft account, like instant messaging contacts. However, the Microsoft account is now subject to the security and privacy policies of the organization.
LinkedDomainController	Required	System.String	This parameter is available only in on-premises Exchange 2013. The *LinkedDomainController* parameter specifies the domain controller in the forest where the user account resides. The domain controller in this forest is used to get

			security information for the account specified by the *LinkedMasterAccount* parameter. This parameter is required only if you're creating a linked mailbox.
LinkedMasterAccount	Required	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	The *LinkedMasterAccount* parameter specifies the master account in the forest where the user account resides. The master account is the account to link the mailbox to. The master account grants access to the mailbox. This parameter is required only if you're creating a linked mailbox. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account`

			Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
LinkedRoom	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *LinkedRoom* parameter is used to specify that the mailbox being created is a linked resource mailbox. A linked resource mailbox is useful in a scenario where you have an account in an authentication forest and you want it to be directly linked to a resource mailbox in resource forest.

MicrosoftOnlineServicesID	Required	Microsoft.Exchange.Data.WindowsLiveId	This parameter is available only in the cloud-based service. The *MicrosoftOnlineServicesID* parameter specifies the user ID for the object. This parameter only applies to objects in the cloud-based service. It isn't available for on-premises deployments.
Name	Required	System.String	The *Name* parameter specifies the user's name. This is the name that appears in Active Directory Users and Computers.
Password	Required	System.Security.SecureString	The *Password* parameter specifies the initial password for the newly created user. This parameter isn't required if you're creating a linked mailbox, resource mailbox, or shared mailbox, because the user account for these types of mailboxes is disabled.
PublicFolder	Required	System.Management.Automation.SwitchPar	The *PublicFolder* parameter specifies that

		ameter	the new mailbox will be a public folder mailbox. Public folder mailboxes are specially designed mailboxes to store the hierarchy and content of public folders. This parameter is required to create a public folder mailbox. The first public folder mailbox created in your Exchange organization is called the *primary hierarchy mailbox*. It contains the writeable copy of the hierarchy of public folders for the organization and public folder content. There can be only one writeable copy of the public folder hierarchy in your organization. All other public folder mailboxes are called *secondary public folder mailboxes* and contain a read-only copy of the hierarchy and the content for public folders.
Room	Required	System.Management.	The *Room* parameter

		Automation.SwitchParameter	specifies that the type of resource is a room, if this mailbox is a resource mailbox. This parameter is required only if you're creating a resource mailbox.
Shared	Required	System.Management.Automation.SwitchParameter	The *Shared* parameter specifies that you're creating a shared mailbox. A shared mailbox is a mailbox to which multiple users can log on. This mailbox isn't associated with any of the users that can log on. It's associated with a disabled user account. This parameter is required only if you're creating a shared mailbox.
UseExistingLiveId	Required	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *UseExistingLiveId* parameter uses the specified Microsoft account (formerly known as a Windows Live ID) that already exists in the cloud-based domain. The

			specified Microsoft account can't have a mailbox associated with it.
UserPrincipalName	Required	System.String	This parameter is available only in on-premises Exchange 2013. The *UserPrincipalName* parameter specifies the UPN for this mailbox. This is the logon name for the user. The UPN consists of a user name and a suffix. Typically, the suffix is the domain name where the user account resides.
WindowsLiveID	Required	Microsoft.Exchange.Data.WindowsLiveId	This parameter is available only in the cloud-based service. The *WindowsLiveID* parameter specifies the Microsoft account (formerly known as a Windows Live ID) of the mailbox.
ActiveSyncMailboxPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *ActiveSyncMailboxPolicy* parameter specifies the mailbox policy to enable for the mailbox that you create. If you don't specify this parameter, the default

			mailbox policy is used.
AddOnSKUCapability	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
AddressBookPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.AddressBookMailboxPolicyIdParameter	This parameter is available only in on-premises Exchange 2013. The *AddressBookPolicy* parameter specifies the address book policy to apply to this mailbox. For more information about address book policies, see Address book policies.
Alias	Optional	System.String	The *Alias* parameter specifies the email alias of the user that you're creating. The alias can be a combination of characters separated by a period with no intervening spaces. Don't use special characters in the alias.
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArbitrationMailbox* parameter specifies the mailbox used to manage

			the moderation process.
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* parameter specifies whether to create an archive mailbox for the specified user. You don't have to specify a value with this parameter.
ArchiveDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArchiveDatabase* parameter specifies the Exchange database that contains the archive associated with this mailbox. You can use the following values: • GUID of the database • Database name
AuxMailbox	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
BypassLiveId	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Database	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *Database* parameter specifies which Exchange database contains the new user's mailbox. You can use the following values: • GUID of the database • Database name
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the display name for the new user created with this mailbox. The value of the *DisplayName* parameter is the name that appears in the Exchange Administration Center. The value of the *DisplayName* parameter also appears in Active Directory Users and Computers on the user **Properties General** tab.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EvictLiveId	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *EvictLiveId* parameter specifies whether to remove an unmanaged Microsoft account (formerly known as a Windows Live ID) from the cloud-based domain. An unmanaged Microsoft account was created in the domain before the domain was enrolled in the cloud-based service. Evicting a Microsoft account from the domain lets you save any settings associated with the Microsoft account, like instant messaging

			contacts.
ExternalDirectoryObjectId	Optional	System.String	This parameter is reserved for internal Microsoft use.
FirstName	Optional	System.String	The *FirstName* parameter specifies the first name of the user that you create.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* parameter specifies whether to suppress warning or confirmation messages. This parameter can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* parameter isn't provided in the command, administrative input is prompted. If the *Force* parameter is provided in the command, but the value is omitted, its default value is `$true`.
ForestWideDomainControllerAffinityByExecutingUser	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
HoldForMigration	Optional	System.Management.Automation.SwitchParameter	The *HoldForMigration* parameter prevents any client or user, except the Microsoft Exchange

			Mailbox Replication service (MRS) process, from logging into a public folder mailbox. Use this parameter when creating the first public folder mailbox in Exchange 2013 if you plan to migrate legacy public folders from Exchange 2010 or Exchange 2007 to Exchange 2013. **⚠Warning:** Use this parameter only if you plan to migrate legacy public folders to Exchange 2013. If you use this parameter but don't have legacy public folders to migrate, you won't be able to create any public folders.
ImmutableId	Optional	System.String	The *ImmutableId* parameter is used by Outlook Live Directory Sync (OLSync) and specifies a unique and immutable identifier in the form of an SMTP address for an Exchange mailbox that's used for federated delegation when requesting Security Assertion Markup Language (SAML) tokens.

			If federation is configured for this mailbox and you don't set this parameter when you create the mailbox, Exchange will create the value for the immutable ID based upon the mailbox's **ExchangeGUID** and the federated account name space, for example, 7a78e7c8-620e-4d85-99d3-c90d90f29699@mail.contoso.com. You must set the *ImmutableId* parameter if Active Directory Federation Services (AD FS) is deployed to allow single sign-on into off-premises mailboxes and AD FS is configured to use a different attribute than **ExchangeGUID** for sign-on token requests. Both, Exchange and AD FS must request the same token for the same user to ensure proper functionality for a cross-premise Exchange deployment scenario.

Initials	Optional	System.String	The *Initials* parameter specifies the initials of the user that you create.
IsExcludedFromServingHierarchy	Optional	System.Boolean	The *IsExcludedFromServingHierarchy* parameter prevents users from accessing the public folder hierarchy on the specified public folder mailbox. For load-balancing purposes, users are equally distributed across public folder mailboxes by default. When this parameter is set on a public folder mailbox, that mailbox isn't included in this automatic load-balancing and won't be accessed by users to retrieve the public folder hierarchy. However, if an administrator has set the *DefaultPublicFolderMailbox* property on a user mailbox to a specific public folder mailbox, the user will still access the specified public folder mailbox even if the *IsExcludedFromServingHie*

			rarchy parameter is set for that public folder mailbox.
Languages	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Languages* parameter specifies the language preferences for this mailbox, in order of preference. Several Exchange components display information to a mailbox user in the preferred language, if that language is supported. Some of those components include quota messages, non-delivery reports (NDRs), the Outlook Web App user interface, and Unified Messaging (UM) voice prompts. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...`

			To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
LastName	Optional	System.String	The *LastName* parameter specifies the last name of the user that you create.
LinkedCredential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *LinkedCredential* parameter specifies credentials to use to access the domain controller specified by the *LinkedDomainController* parameter. This parameter is optional, even if you're enabling a linked mailbox.
MailboxContainerGuid	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
MailboxPlan	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPlanIdParameter	This parameter is available only in the cloud-based service. The *MailboxPlan* parameter specifies the mailbox plan to associate

			with this mailbox. A mailbox plan specifies the permissions and features available to a mailbox user. The mailbox plan name you provide must be included in the service plan of the organization in which this mailbox resides.
MailboxProvisioningConstraint	Optional	Microsoft.Exchange.Data.Directory.MailboxProvisioningConstraint	This parameter is reserved for internal Microsoft use.
MailboxProvisioningPreferences	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
ManagedFolderMailboxPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	This parameter is available only in on-premises Exchange 2013. The *ManagedFolderMailboxPolicy* parameter specifies the managed folder mailbox policy to enable for the mailbox that you create.
ManagedFolderMailboxPolicyAllowed	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The

			ManagedFolderMailboxPolicyAllowed parameter specifies whether to bypass the warning that messaging records management (MRM) features aren't supported for email clients using versions of Microsoft Outlook earlier than Office Outlook 2007. When a managed folder mailbox policy is assigned to a mailbox using the *ManagedFolderMailboxPolicy* parameter, the warning appears by default unless the *ManagedFolderMailboxPolicyAllowed* parameter is used.
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users who are responsible for moderating the messages sent to this mailbox. To designate more than one user, separate the users with commas. This parameter is required if you set the

			ModerationEnabled parameter to `$true`. If you leave this parameter blank and there's a user who's already specified as the manager of this mailbox, the *ModeratedBy* parameter is automatically set to the *ManagedBy* parameter of the mailbox. Otherwise, an error is returned.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable moderation for the mailbox. The two possible values for this parameter are `$true` or `$false`. To enable moderation, set this parameter to `$true`. To disable moderation, set this parameter to `$false`. The default value is `$false`.
NetID	Optional	Microsoft.Exchange.Data.NetID	This parameter is reserved for internal Microsoft use.
Office	Optional	System.String	The *Office* parameter specifies the Microsoft Office attribute for this mailbox.

Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies the container where the user is created.
OriginalNetID	Optional	Microsoft.Exchange.Data.NetID	This parameter is reserved for internal Microsoft use.
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Phone	Optional	System.String	The *Phone* parameter specifies the user's telephone number for this mailbox.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the primary SMTP address of the mailbox.
QueryBaseDNRestrictionEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RemoteAccountPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.RemoteAccountPolicyIdParameter	This parameter is reserved for internal Microsoft use.
RemoteArchive	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013.

			The *RemoteArchive* parameter specifies whether to disconnect the remote archive for this mailbox. A remote archive exists in a cloud-based service. When you use this parameter, you can't use the *Archive* parameter.
RemotePowerShellEnabled	Optional	System.Boolean	The *RemotePowerShellEnabled* parameter specifies whether the user can use remote PowerShell. Remote PowerShell is required to open the Exchange Management Shell or the Exchange Administration Center. Access to remote PowerShell is required even if you're trying to open the Shell or the EAC on the local server.
RemovedMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.RemovedMailboxIdParameter	This parameter is reserved for internal Microsoft use.
ResetPasswordOnNextLogon	Optional	System.Boolean	The *ResetPasswordOnNextLog*

			on parameter specifies whether the password in the *Password* parameter must be reset the next time the user logs on. If set to `$true`, the *ResetPasswordOnNextLogon* parameter specifies that the password in the *Password* parameter must be reset the next time the user logs on.
ResourceCapacity	Optional	System.Int32	The *ResourceCapacity* parameter specifies capacity, if this mailbox is a resource mailbox. You must specify a non-negative integer.
RetentionPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	This parameter is available only in on-premises Exchange 2013. The *RetentionPolicy* parameter specifies the name of a retention policy that you want applied to this mailbox. Retention policies consist of tags that are applied to mailbox folders and mail items to determine the period of time that the

			items should be retained.
RoleAssignmentPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *RoleAssignmentPolicy* parameter specifies the management role assignment policy to assign to the mailbox when it's created or enabled. If you don't include this parameter when you create or enable a mailbox, the default assignment policy is used. If the assignment policy name contains spaces, enclose the name in quotation marks ("). If you don't want to assign an assignment policy when a mailbox is created or enabled, specify a value of `$null`. For more information about assignment policies, see Understanding management role assignment policies.
RoomMailboxPassword	Optional	System.Security.SecureString	Use the *RoomMailboxPassword* parameter to specify a password when using the *EnableRoomMailboxAccount* parameter to create a

			logon-enabled room mailbox. Use the following syntax to specify the password: `-RoomMailboxPassword (ConvertTo-SecureString -String <password> -AsPlainText -Force)`.
SamAccountName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *SamAccountName* parameter specifies the logon name used to support clients and servers running older versions of the operating system, such as Microsoft Windows NT 4.0, Windows 95, Windows 98, and LAN Manager. This attribute must be less than 20 characters to support older clients. If you don't specify the *SamAccountName* parameter, Active Directory creates a **SAMAccountName** attribute automatically, based on the UPN.
SendModerationNotifi	Optional	Microsoft.Exchange.Da	The

cations		ta.Directory.Recipient.TransportModerationNotificationFlags	*SendModerationNotifications* parameter specifies whether status notifications are sent to users when they send a message to the moderated mailbox. You can specify one of the following values: • `Always` • `Internal` • `Never` If you want notifications to be sent to all senders, set this value to `Always`. If you want notifications to be sent only to the senders who are internal to your organization, set this value to `Internal`. To disable all status notifications, set this value to `Never`. **Note:** The sender is always notified if the message is rejected by the moderators, regardless of the value of this parameter. The default value is `Never`.
SharingPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.SharingPolicyIdParameter	This parameter is available only in on-premises Exchange 2013. The *SharingPolicy*

			parameter specifies the identity of the sharing policy associated with this mailbox.
SKUAssigned	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
SKUCapability	Optional	Microsoft.Exchange.Data.Directory.Capability	This parameter is reserved for internal Microsoft use.
TargetAllMDBs	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ThrottlingPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.ThrottlingPolicyIdParameter	This parameter is available only in on-premises Exchange 2013. The *ThrottlingPolicy* parameter specifies the identity of the throttling policy that you want to specify for this mailbox.
UsageLocation	Optional	Microsoft.Exchange.Data.Directory.CountryInfo	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply

			any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-Mailbox

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-22*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-Mailbox** cmdlet to delete the user account associated with a specific mailbox from Active Directory and to process the associated, disconnected mailbox as directed by the specified parameters.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-Mailbox -Identity <MailboxIdParameter> [-KeepWindowsLiveID
<SwitchParameter>] [-Permanent <$true | $false>] <COMMON PARAMETERS>


Remove-Mailbox -Database <DatabaseIdParameter> -StoreMailboxIdentity
<StoreMailboxIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Arbitration <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-Disconnect <SwitchParameter>] [-DomainController
<Fqdn>] [-Force <SwitchParameter>] [-ForReconciliation <SwitchParameter>]
[-IgnoreDefaultScope <SwitchParameter>] [-IgnoreLegalHold
<SwitchParameter>] [-PublicFolder <SwitchParameter>] [-
RemoveArbitrationMailboxWithOABsAllowed <SwitchParameter>] [-
RemoveLastArbitrationMailboxAllowed <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disconnects the user John Rodman's (john) mailbox from the user account and removes the mailbox object from Active Directory. The mailbox remains in the Exchange database for the deleted mailbox retention period configured for the mailbox database.

```
Remove-Mailbox -Identity contoso\john
```

EXAMPLE 2

This example disconnects the user John Rodman's (john) mailbox from the user account, removes the mailbox object from Active Directory, and removes the mailbox from the Exchange database.

```
Remove-Mailbox -Identity contoso\john -Permanent $true
```

EXAMPLE 3

This example removes John Rodman's mailbox from the Exchange database, assuming the mailbox has already been disconnected from the user. The example uses the **Get-Mailbox** cmdlet to retrieve the mailbox GUID value using the display name of the disconnected mailbox. This value is needed for the *StoreMailboxIdentity* parameter of the **Remove-Mailbox** cmdlet.

```
$Temp = Get-Mailbox | Where {$_.DisplayName -eq 'John
Rodman'}
Remove-Mailbox -Database Server01\Database01 -
StoreMailboxIdentity $Temp.MailboxGuid
```

Detailed Description

Use the *Identity* parameter alone to disconnect the mailbox from the user and remove the user object from Active Directory. The mailbox object still exists. By default, this mailbox remains in the Exchange database for 30 days, and then is deleted.

Use the *Identity* and *Permanent* parameters to disconnect the mailbox from the user, remove the user object from Active Directory, and remove the mailbox object from the Exchange database. The mailbox object doesn't remain in the Exchange database as a disconnected mailbox.

Use the *Database* and *StoreMailboxIdentity* parameters to remove a mailbox object from the Exchange database. In this case, the mailbox object has already been disconnected from the user. For example, if you run the **Disable-Mailbox** cmdlet, the Exchange mailbox object still exists, but is disconnected from the Active Directory user object. You can use the *Database* and *StoreMailboxIdentity* parameters to remove this disconnected mailbox object.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Database	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *Database* parameter specifies the database that contains the mailbox object. You can use one of the following values: • GUID • Database name This parameter must be used in conjunction with the *StoreMailboxIdentity* parameter. The *Database* parameter can't be used with the *Identity* parameter. If you've disconnected a mailbox from its associated user and want to remove the mailbox object from the Exchange store, use the *Database* and *StoreMailboxIdentity* parameters.

Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter identifies the mailbox object that you want to remove. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name`

			Example: JPhillips@contoso.com The *Identity* parameter can't be used with the *Database* parameter.
StoreMailboxIdentity	Required	Microsoft.Exchange.Configuration.Tasks.StoreMailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *StoreMailboxIdentity* parameter specifies the mailbox object to remove. The *StoreMailboxIdentity* parameter is used in conjunction with the *Database* parameter to remove the mailbox object from the Exchange database. If you've disconnected a mailbox from its associated user and want to remove the mailbox object from the Exchange store, use the *Database* and *StoreMailboxIdentity* parameters.
Arbitration	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Arbitration* parameter specifies that the mailbox for which you are

			executing the command is an arbitration mailbox. Arbitration mailboxes are used for managing approval workflow. For example, an arbitration mailbox is used for handling moderated recipients and distribution group membership approval.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Disconnect	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
ForReconciliation	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell

			session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
IgnoreLegalHold	Optional	System.Management.Automation.SwitchParameter	The *IgnoreLegalHold* switch ignores the legal hold status of the mail user and allows you to remove the mailbox that's on legal hold. **Caution:** After you remove a mailbox, you can't include it in a discovery search. Depending on the

			command parameters you use, removed mailboxes are either purged immediately or when the deleted mailbox retention period expires. Check with your organization's legal or Human Resources department before disabling a mailbox that's on legal hold.
KeepWindowsLiveID	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *KeepWindowsLiveID* parameter preserves the Microsoft account (formerly known as a Windows Live ID) that's associated with the deleted mailbox.
Permanent	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *Permanent* parameter, when used in conjunction with the *Identity* parameter, specifies whether to disconnect the mailbox from the user, remove the associated user object from Active Directory, and remove the mailbox

			object from the Exchange database. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`.
PublicFolder	Optional	System.Management.Automation.SwitchParameter	The *PublicFolder* parameter specifies that the mailbox to remove is a public folder mailbox. Public folder mailboxes are specially designed mailboxes to store the hierarchy and content of public folders. You have to include this parameter to remove a public folder mailbox.
RemoveArbitrationMailboxWithOABsAllowed	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemoveArbitrationMailboxWithOABsAllowed* parameter is used bypass the checks for offline address books (OABs) within the specified arbitration mailbox that is being removed, and remove the arbitration mailbox even if OABs are

			present in the mailbox.
RemoveLastArbitrationMailboxAllowed	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemoveLastArbitrationMailboxAllowed* switch specifies whether the mailbox that you're trying to use is the last arbitration mailbox in the organization.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Search-Mailbox

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-28*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Search-Mailbox** cmdlet to search a mailbox and copy the results to a specified target mailbox, delete messages from the source mailbox, or both.

```
Search-Mailbox -TargetFolder <String> -TargetMailbox <MailboxIdParameter>
[-DeleteContent <SwitchParameter>] [-LogLevel <Suppress | Basic | Full>]
[-LogOnly <SwitchParameter>] <COMMON PARAMETERS>


Search-Mailbox [-DeleteContent <SwitchParameter>] <COMMON PARAMETERS>


Search-Mailbox -EstimateResultOnly <SwitchParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: -Identity <MailboxOrMailUserIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-DoNotIncludeArchive
<SwitchParameter>] [-Force <SwitchParameter>] [-IncludeUnsearchableItems
<SwitchParameter>] [-SearchDumpster <SwitchParameter>] [-
SearchDumpsterOnly <SwitchParameter>] [-SearchQuery <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example searches the mailbox of Joe Healy and copies the search results to the DiscoveryMailbox in the folder JoeHealy-ProjectHamilton.

```
Search-Mailbox -Identity "Joe Healy" -SearchQuery
"Subject:Project Hamilton" -TargetMailbox
"DiscoveryMailbox" -TargetFolder "JoeHealy-ProjectHamilton"
-LogLevel Full
```

EXAMPLE 2

This example searches April Stewart's mailbox for messages that contain the phrase "Your bank statement" in the subject and logs the result in the SearchAndDeleteLog folder in the administrator's mailbox. Messages aren't copied to the target mailbox.

```
Search-Mailbox -Identity "April Stewart" -SearchQuery
'Subject:"Your bank statement"' -TargetMailbox
"administrator" -TargetFolder "SearchAndDeleteLog" -LogOnly
-LogLevel Full
```

EXAMPLE 3

This example searches April Stewart's mailbox for messages that contain the phrase "Your bank statement" in the subject and deletes the messages from the source mailbox.

```
Search-Mailbox -Identity "April Stewart" -SearchQuery
'Subject:"Your bank statement"' -DeleteContent
```

EXAMPLE 4

This example searches all mailboxes in your organization for messages that contain the words "election", "candidate", or "vote". The search results are copied to the Discovery Search Mailbox in the folder AllMailboxes-Election.

```
Get-Mailbox | Search-Mailbox -SearchQuery 'election OR
candidate OR vote' -TargetMailbox "Discovery Search
Mailbox" -TargetFolder "AllMailboxes-Election" -LogLevel
Full
```

Detailed Description

You can use the **Search-Mailbox** cmdlet to search messages in a specified mailbox and perform any of the following tasks:

- Copy messages to a specified target mailbox.
- Delete messages from the source mailbox.
- Copy messages from the source mailbox and delete them from the target mailbox.
- Perform single item recovery to recover items from a user's Recoverable Items folder.
- Clean up the Recoverable Items folder for a mailbox when it has reached the Recoverable Items hard quota.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "In-Place eDiscovery" and "Delete mailbox content" entries in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
EstimateResultOnly	Required	System.Management.Automation.SwitchParameter	The *EstimateResultOnly* switch specifies that only an estimate of the total number and size of messages returned by the search be provided. Messages aren't copied to the target mailbox. You can't use this switch with the *TargetMailbox* parameter.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxOrMailUserIdParameter	The *Identity* parameter specifies the identity of the mailbox to search. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2

			• `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
TargetFolder	Required	System.String	The *TargetFolder* parameter specifies a folder name in which search results are saved in the target mailbox. The folder is created in the target mailbox upon execution.
TargetMailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *TargetMailbox* parameter specifies the identity of the destination mailbox where search results are copied. You can use the following values: • Alias • Display name • *Domain\Account* • SMTP address • DN

			• Object GUID • UPN • LegacyExchangeDN When you specify a value for the *TargetMailbox* parameter, you must also specify the *TargetFolder* parameter. You can't use this parameter with the *EstimateResultOnly* switch.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DeleteContent	Optional	System.Management.Automation.SwitchParameter	The *DeleteContent* switch specifies that the messages returned by the search be permanently deleted from the source mailbox. When used with the *TargetMailbox* parameter, messages are copied to the target mailbox and removed from the source mailbox.

			If you set the logging level for the search to `Basic` or `Full`, you must specify a target mailbox and a target folder to place the log in. To delete messages from the source mailbox without copying them to the target mailbox, don't specify the *TargetMailbox*, *TargetFolder*, and *LogLevel* parameters. **Important:** You need to be assigned the Mailbox Import Export management role to use this switch. By default, this role isn't assigned to any role group. Typically, you assign a role to a built-in or custom role group. Or you can assign a role to a user, or a universal security group. Before you use the *DeleteContent* switch to delete content, we recommend that you test search parameters by using the *LogOnly* parameter, as shown in Example 2.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory.
DoNotIncludeArchive	Optional	System.Management.Automation.SwitchParameter	The *DoNotIncludeArchive* switch specifies that the user's archive mailbox shouldn't be included in the search. You don't need to specify a value for this switch.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch overrides the confirmation prompt displayed when your use the *DeleteContent* switch to permanently delete messages.
IncludeUnsearchableItems	Optional	System.Management.Automation.SwitchParameter	The *IncludeUnsearchableItems* switch specifies whether to include items that couldn't be indexed by Exchange Search. When set to `$true`, the *IncludeUnsearchableItems* switch specifies that items that couldn't be indexed by Exchange Search should be included in the

			search results.
LogLevel	Optional	Microsoft.Exchange.Data.Storage.Infoworker.MailboxSearch.LoggingLevel	The *LogLevel* parameter specifies the logging level for the search. It can have one of the following values: • `Suppress` No logs are kept. • `Basic` Basic information about the query and who ran it is kept. • `Full` In addition to the information kept by the `Basic` log level, the `Full` log level adds a complete list of search results. The default log level is `Basic`.
LogOnly	Optional	System.Management.Automation.SwitchParameter	The *LogOnly* switch specifies that a search be performed and only a log be generated. Messages returned by the search aren't copied to the target mailbox. The logging level is specified by using the *LogLevel* parameter.
SearchDumpster	Optional	System.Management.Automation.SwitchParameter	The *SearchDumpster* parameter specifies whether to search the Recoverable Items folder, which is the storage

			location in which items deleted from the Deleted Items folder or hard-deleted items are stored until they're purged from the mailbox database. By default, the Recoverable Items folder is always searched. To exclude the folder from the search, set the *SearchDumpster* switch to `$false`, for example,- `SearchDumpster:$false`
SearchDumpsterOnly	Optional	System.Management.Automation.SwitchParameter	The *SearchDumpsterOnly* switch specifies that only the Recoverable Items folder of the specified mailbox be searched. You can also use this switch with the *DeleteContent* switch to delete messages from the Recoverable Items folder and reduce the size of the folder.
SearchQuery	Optional	System.String	The *SearchQuery* parameter specifies a search string or a query formatted using Keyword Query Language (KQL). For more details about KQL, see Keyword Query Language syntax

			reference. If this parameter is empty, all messages are returned.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-Mailbox

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-21*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-Mailbox** cmdlet to modify the settings of an existing mailbox. You can use this cmdlet for one mailbox at a time. To perform bulk management, you can pipeline the output of various **Get-** cmdlets (for example, the **Get-Mailbox** or **Get-User** cmdlets) and configure several mailboxes in a single-line command. You can also use the **Set-Mailbox** cmdlet in scripts.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-Mailbox -Identity <MailboxIdParameter> <COMMON PARAMETERS>


Set-Mailbox -AddAggregatedAccount <SwitchParameter> -AggregatedMailboxGuid
<Guid> -Identity <MailboxIdParameter> <COMMON PARAMETERS>


Set-Mailbox -AggregatedMailboxGuid <Guid> -Identity <MailboxIdParameter> -
RemoveAggregatedAccount <SwitchParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-AcceptMessagesOnlyFrom <MultiValuedProperty>] [-
AcceptMessagesOnlyFromDLMembers <MultiValuedProperty>] [-
AcceptMessagesOnlyFromSendersOrMembers <MultiValuedProperty>] [-
AddOnSKUCapability <MultiValuedProperty>] [-AddressBookPolicy
<AddressBookMailboxPolicyIdParameter>] [-Alias <String>] [-
AntispamBypassEnabled <$true | $false>] [-ApplyMandatoryProperties
<SwitchParameter>] [-Arbitration <SwitchParameter>] [-ArbitrationMailbox
<MailboxIdParameter>] [-ArchiveDatabase <DatabaseIdParameter>] [-
ArchiveDomain <SmtpDomain>] [-ArchiveName <MultiValuedProperty>] [-
ArchiveQuota <Unlimited>] [-ArchiveStatus <None | Active>] [-
ArchiveWarningQuota <Unlimited>] [-AuditAdmin <MultiValuedProperty>] [-
AuditDelegate <MultiValuedProperty>] [-AuditEnabled <$true | $false>] [-
AuditLogAgeLimit <EnhancedTimeSpan>] [-AuditOwner <MultiValuedProperty>]
[-BypassLiveId <SwitchParameter>] [-BypassModerationFromSendersOrMembers
<MultiValuedProperty>] [-CalendarLoggingQuota <Unlimited>] [-
CalendarRepairDisabled <$true | $false>] [-CalendarVersionStoreDisabled
<$true | $false>] [-ClientExtensions <$true | $false>] [-Confirm
[<SwitchParameter>]] [-CreateDTMFMap <$true | $false>] [-CustomAttribute1
<String>] [-CustomAttribute10 <String>] [-CustomAttribute11 <String>] [-
CustomAttribute12 <String>] [-CustomAttribute13 <String>] [-
CustomAttribute14 <String>] [-CustomAttribute15 <String>] [-
CustomAttribute2 <String>] [-CustomAttribute3 <String>] [-CustomAttribute4
<String>] [-CustomAttribute5 <String>] [-CustomAttribute6 <String>] [-
CustomAttribute7 <String>] [-CustomAttribute8 <String>] [-CustomAttribute9
<String>] [-Database <DatabaseIdParameter>] [-DefaultPublicFolderMailbox
<RecipientIdParameter>] [-DeliverToMailboxAndForward <$true | $false>] [-
DisplayName <String>] [-DomainController <Fqdn>] [-
DowngradeHighPriorityMessagesEnabled <$true | $false>] [-
DumpsterMessagesPerFolderCountReceiveQuota <Int32>] [-
DumpsterMessagesPerFolderCountWarningQuota <Int32>] [-EmailAddresses
<ProxyAddressCollection>] [-EmailAddressPolicyEnabled <$true | $false>] [-
EnableRoomMailboxAccount <$true | $false>] [-EndDateForRetentionHold
<DateTime>] [-EvictLiveId <SwitchParameter>] [-
ExtendedPropertiesCountQuota <Int32>] [-ExtensionCustomAttribute1
<MultiValuedProperty>] [-ExtensionCustomAttribute2 <MultiValuedProperty>]
[-ExtensionCustomAttribute3 <MultiValuedProperty>] [-
ExtensionCustomAttribute4 <MultiValuedProperty>] [-
ExtensionCustomAttribute5 <MultiValuedProperty>] [-ExternalOofOptions
<InternalOnly | External>] [-FederatedIdentity <String>] [-
FolderHierarchyChildrenCountReceiveQuota <Int32>] [-
FolderHierarchyChildrenCountWarningQuota <Int32>] [-
FolderHierarchyDepthReceiveQuota <Int32>] [-
FolderHierarchyDepthWarningQuota <Int32>] [-FoldersCountReceiveQuota
<Int32>] [-FoldersCountWarningQuota <Int32>] [-Force <SwitchParameter>] [-
ForestWideDomainControllerAffinityByExecutingUser <SwitchParameter>] [-
ForwardingAddress <RecipientIdParameter>] [-ForwardingSmtpAddress
<ProxyAddress>] [-GMGen <$true | $false>] [-GrantSendOnBehalfTo
<MultiValuedProperty>] [-HiddenFromAddressListsEnabled <$true | $false>]
[-IgnoreDefaultScope <SwitchParameter>] [-ImListMigrationCompleted <$true
| $false>] [-ImmutableId <String>] [-IsExcludedFromServingHierarchy <$true
| $false>] [-IsHierarchyReady <$true | $false>] [-IssueWarningQuota
<Unlimited>] [-JournalArchiveAddress <SmtpAddress>] [-Languages
<MultiValuedProperty>] [-LinkedCredential <PSCredential>] [-
LinkedDomainController <String>] [-LinkedMasterAccount <UserIdParameter>]
[-LitigationHoldDate <DateTime>] [-LitigationHoldDuration <Unlimited>] [-
```

```
LitigationHoldEnabled <$true | $false>] [-LitigationHoldOwner <String>] [-
MailboxContainerGuid <Guid>] [-MailboxMessagesPerFolderCountReceiveQuota
<Int32>] [-MailboxMessagesPerFolderCountWarningQuota <Int32>] [-
MailboxPlan <MailboxPlanIdParameter>] [-MailboxProvisioningConstraint
<MailboxProvisioningConstraint>] [-MailboxProvisioningPreferences
<MultiValuedProperty>] [-MailRouting <$true | $false>] [-MailTip <String>]
[-MailTipTranslations <MultiValuedProperty>] [-ManagedFolderMailboxPolicy
<MailboxPolicyIdParameter>] [-ManagedFolderMailboxPolicyAllowed
<SwitchParameter>] [-Management <$true | $false>] [-MaxBlockedSenders
<Int32>] [-MaxReceiveSize <Unlimited>] [-MaxSafeSenders <Int32>] [-
MaxSendSize <Unlimited>] [-MessageTracking <$true | $false>] [-
MessageTrackingReadStatusEnabled <$true | $false>] [-
MicrosoftOnlineServicesID <SmtpAddress>] [-Migration <$true | $false>] [-
ModeratedBy <MultiValuedProperty>] [-ModerationEnabled <$true | $false>]
[-Name <String>] [-NetID <NetID>] [-NewPassword <SecureString>] [-OABGen
<$true | $false>] [-Office <String>] [-OfflineAddressBook
<OfflineAddressBookIdParameter>] [-OldPassword <SecureString>] [-
OMEncryption <$true | $false>] [-OriginalNetID <NetID>] [-Password
<SecureString>] [-PrimarySmtpAddress <SmtpAddress>] [-ProhibitSendQuota
<Unlimited>] [-ProhibitSendReceiveQuota <Unlimited>] [-PstProvider <$true
| $false>] [-PublicFolder <SwitchParameter>] [-QueryBaseDN
<OrganizationalUnitIdParameter>] [-QueryBaseDNRestrictionEnabled <$true |
$false>] [-RecipientLimits <Unlimited>] [-RecoverableItemsQuota
<Unlimited>] [-RecoverableItemsWarningQuota <Unlimited>] [-
RejectMessagesFrom <MultiValuedProperty>] [-RejectMessagesFromDLMembers
<MultiValuedProperty>] [-RejectMessagesFromSendersOrMembers
<MultiValuedProperty>] [-RemoteAccountPolicy
<RemoteAccountPolicyIdParameter>] [-RemoteRecipientType <None |
ProvisionMailbox | ProvisionArchive | Migrated | DeprovisionMailbox |
DeprovisionArchive | RoomMailbox | EquipmentMailbox | SharedMailbox |
TeamMailbox>] [-RemoveManagedFolderAndPolicy <SwitchParameter>] [-
RemovePicture <SwitchParameter>] [-RemoveSpokenName <SwitchParameter>] [-
RequireSecretQA <$true | $false>] [-RequireSenderAuthenticationEnabled
<$true | $false>] [-ResetPasswordOnNextLogon <$true | $false>] [-
ResourceCapacity <Int32>] [-ResourceCustom <MultiValuedProperty>] [-
RetainDeletedItemsFor <EnhancedTimeSpan>] [-RetainDeletedItemsUntilBackup
<$true | $false>] [-RetentionComment <String>] [-RetentionHoldEnabled
<$true | $false>] [-RetentionPolicy <MailboxPolicyIdParameter>] [-
RetentionUrl <String>] [-RoleAssignmentPolicy <MailboxPolicyIdParameter>]
[-RoomMailboxPassword <SecureString>] [-RulesQuota <ByteQuantifiedSize>]
[-SamAccountName <String>] [-SCLDeleteEnabled <$true | $false>] [-
SCLDeleteThreshold <Int32>] [-SCLJunkEnabled <$true | $false>] [-
SCLJunkThreshold <Int32>] [-SCLQuarantineEnabled <$true | $false>] [-
SCLQuarantineThreshold <Int32>] [-SCLRejectEnabled <$true | $false>] [-
SCLRejectThreshold <Int32>] [-SecondaryAddress <String>] [-
SecondaryDialPlan <UMDialPlanIdParameter>] [-SendModerationNotifications
<Never | Internal | Always>] [-SharingPolicy <SharingPolicyIdParameter>]
[-SimpleDisplayName <String>] [-SingleItemRecoveryEnabled <$true |
$false>] [-SkipMailboxProvisioningConstraintValidation <SwitchParameter>]
[-SKUAssigned <$true | $false>] [-SKUCapability <None | BPOS_S_Deskless |
BPOS_S_Standard | BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard |
BPOS_B_Standard | BPOS_B_CustomDomain | BPOS_S_MidSize |
BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn |
BPOS_Unmanaged | TOU_Signed | FederatedUser | Partner_Managed |
MasteredOnPremise | ResourceMailbox | ExcludedFromBackSync |
UMFeatureRestricted | RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-StartDateForRetentionHold <DateTime>]
[-SuiteServiceStorage <$true | $false>] [-TenantUpgrade <$true | $false>]
[-ThrottlingPolicy <ThrottlingPolicyIdParameter>] [-Type <Regular | Room |
Equipment | Shared>] [-UMDataStorage <$true | $false>] [-UMDtmfMap
<MultiValuedProperty>] [-UMGrammar <$true | $false>] [-UnifiedMailbox
<CrossTenantObjectId>] [-UsageLocation <CountryInfo>] [-
UseDatabaseQuotaDefaults <$true | $false>] [-UseDatabaseRetentionDefaults
<$true | $false>] [-UserCertificate <MultiValuedProperty>] [-
UserPrincipalName <String>] [-UserSMimeCertificate <MultiValuedProperty>]
[-WhatIf [<SwitchParameter>]] [-WindowsEmailAddress <SmtpAddress>] [-
WindowsLiveID <SmtpAddress>]
```

Examples

EXAMPLE 1

This example delivers John Woods's email messages to John's mailbox and also forwards them to Manuel Oliveira's (manuel@contoso.com) mailbox.

```
Set-Mailbox -Identity John -DeliverToMailboxAndForward
$true -ForwardingSMTPAddress manuel@contoso.com
```

EXAMPLE 2

This example uses the **Get-Mailbox** cmdlet to find all the mailboxes in the Marketing organizational unit, and then uses the **Set-Mailbox** cmdlet to configure these mailboxes. The custom warning, prohibit send, and prohibit send and receive limits are set to 200 megabytes (MB), 250 MB, and 280 MB respectively, and the mailbox database's default limits are ignored. This command can be used to configure a specific set of mailboxes to have larger or smaller limits than other mailboxes in the organization.

```
Get-Mailbox -OrganizationalUnit "Marketing" | Set-Mailbox -
IssueWarningQuota 209715200 -ProhibitSendQuota 262144000 -
ProhibitSendReceiveQuota 293601280 -
UseDatabaseQuotaDefaults $false
```

EXAMPLE 3

This example uses the **Get-User** command to find all users in the Customer Service department, and then uses the **Set-Mailbox** command to change the maximum message size for sending messages to 2 MB.

```
Get-User -Filter "Department -eq 'Customer Service'" | Set-
Mailbox -MaxSendSize 2097152
```

EXAMPLE 4

This example sets the MailTip translation in French and Chinese.

```
Set-Mailbox John@contoso.com -MailTipTranslations ("FR:
C'est la langue française", "CHT: 這是漢語語言")
```

EXAMPLE 5

This example resets the password for Florence Flipo's mailbox. The next time she signs in to her mailbox, she'll have to reset the new password.

```
Set-Mailbox florencef -Password (ConvertTo-SecureString -
String 'P@$$w0rd1' -AsPlainText -Force) -OldPassword
(ConvertTo-SecureString -String 'Pa$$word1' -AsPlainText -
Force) -ResetPasswordOnNextLogon $true
```

EXAMPLE 6

This example removes the message tracking organization capability from the arbitration mailbox named SystemMailbox{bb558c35-97f1-4cb9-8ff7-d53741dc928c} and assigns it to an arbitration mailbox named SystemMailbox{1f05a927-b864-48a7-984d-95b1adfbfe2d}.

```
Set-Mailbox -Arbitration -Identity "SystemMailbox{bb558c35-
97f1-4cb9-8ff7-d53741dc928c}" -MessageTracking $false

Set-Mailbox -Arbitration -Identity "SystemMailbox{1f05a927-
b864-48a7-984d-95b1adfbfe2d}" -MessageTracking $true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AddAggregatedAccount	Required	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
AggregatedMailboxGuid	Required	System.Guid	This parameter is reserved for internal Microsoft use.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox. This parameter accepts the following values: • `Alias`

			Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
RemoveAggregatedAccount	Required	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
AcceptMessagesOnlyF	Optional	Microsoft.Exchange.Da	The

rom		ta.MultiValuedProperty	*AcceptMessagesOnlyFrom* parameter specifies the mailbox users, mail users, and mail contacts that can send email messages to this mailbox. You can also specify Exchange as a valid recipient for this parameter. If you configure a mailbox to accept messages only from the Exchange recipient, the mailbox receives only system-generated messages. You can use any of the following values for the valid senders: • DN • Canonical name • GUID • Name • Display name • Alias • Exchange DN • Primary SMTP email address By default, this parameter is blank, which enables the mailbox to accept messages from all senders.

			To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
AcceptMessagesOnlyFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromDLMembers* parameter specifies the distribution groups whose members are allowed to send email messages to this mailbox. You can use any of the following values for the allowed distribution groups: • DN • Canonical name

			• GUID • Name • Display name • LegacyExchangeDN • Primary SMTP email address. By default, this parameter is blank, which enables the mailbox to accept messages from all senders. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
AcceptMessagesOnlyFromSendersOrMembe	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *AcceptMessagesOnlyFrom*

rs		y	*SendersOrMembers* parameter specifies the recipients who can send email messages to this mailbox. You can specify users, contacts, or distribution groups. If you specify a distribution group, messages are accepted from all recipients that are members of that distribution group. You can also specify Exchange as a valid recipient for this parameter. If you configure a distribution group to accept messages only from the Exchange recipient, the distribution group only receives system-generated messages. You can use any of the following values for the valid senders: • DN • Canonical name • GUID • Name • Display name • Alias • Exchange DN

			• Primary SMTP email address By default, the value of this parameter is blank, which enables the distribution group to accept messages from all senders. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
AddOnSKUCapability	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
AddressBookPolicy	Optional	Microsoft.Exchange.Co	The *AddressBookPolicy*

		nfiguration.Tasks.AddressBookMailboxPolicyIdParameter	parameter specifies the address book policy that applies to this mailbox. For more information about address book policies, see Address book policies.
Alias	Optional	System.String	The *Alias* parameter specifies the alias (mail nickname) of the user. The alias can be a combination of characters separated by a period with no intervening spaces. Don't use special characters in the alias.
AntispamBypassEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *AntispamBypassEnabled* parameter specifies whether to skip anti-spam processing on this mailbox. The two possible values for this parameter are `$true` or `$false`.
ApplyMandatoryProperties	Optional	System.Management.Automation.SwitchParameter	The *ApplyMandatoryProperties* parameter specifies whether to modify the

			mandatory properties of a mailbox. Creating a mailbox through the Microsoft Exchange extensions to the Active Directory Users and Computers console isn't supported. If a mailbox is created with this tool, it's identified as a legacy mailbox, even though it resides on a server running Microsoft Exchange. This parameter modifies the mandatory properties of a mailbox in this state to correct the problem and remove the legacyMailbox tag from the mailbox.
Arbitration	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Arbitration* parameter specifies that the mailbox for which you are executing the command is an arbitration mailbox. Arbitration mailboxes are used for managing approval workflow. For example, an arbitration

			mailbox is used for handling moderated recipients and distribution group membership approval.
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArbitrationMailbox* parameter specifies the mailbox used to manage the moderation process.
ArchiveDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArchiveDatabase* parameter specifies the Exchange database that contains the archive associated with this mailbox. You can use the following values: • GUID of the database • Database name
ArchiveDomain	Optional	Microsoft.Exchange.Data.SmtpDomain	This parameter is available only in on-premises Exchange 2013. The *ArchiveDomain* parameter specifies the cloud-based service domain on which the

			archive associated with this mailbox exists. For example, if the SMTP email address of the user is tony@mail.contoso.com, the SMTP domain could be hosted.contoso.com.
ArchiveName	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ArchiveName* parameter specifies the name of the archive mailbox. This is the name displayed to users in Microsoft Office Outlook Web App and Microsoft Outlook. The default name is "Online Archive - *<Mailbox User's Display Name>*".
ArchiveQuota	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *ArchiveQuota* parameter specifies the archive mailbox size at which messages will no longer be accepted. The value must be greater than the value of the *ArchiveWarningQuota* parameter. The valid input range for either

			parameter is from 1 through 9223372036854775807 bytes. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes.
ArchiveStatus	Optional	Microsoft.Exchange.Data.Directory.Recipient.ArchiveStatusFlags	This parameter is reserved for internal Microsoft use.
ArchiveWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *ArchiveWarningQuota* parameter specifies the archive mailbox size at which a warning message is sent to the user. The value must be less than the value of the *ArchiveQuota* parameter. The valid input range for either parameter is from 1 through

			9223372036854775807 bytes. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes.
AuditAdmin	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AuditAdmin* parameter specifies the operations to log for administrators. Valid values include: • `None` • `Update` • `Copy` • `Move` • `MoveToDeletedItems` • `SoftDelete` • `HardDelete` • `FolderBind` • `SendAs` • `SendOnBehalf` • `MessageBind` By default, the `Update`, `Move`, `MoveToDeletedItems`, `SoftDelete`, `HardDelete`, `FolderBind`, `SendAs`, and `SendOnBehalf` actions performed by administrators are logged. To enter multiple values and overwrite any existing

			entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. **Note:** The *AuditEnabled* parameter must be set to `$true` to enable logging.
AuditDelegate	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AuditDelegate* parameter specifies the operations to log for delegate users. Valid values include: • `None` • `Update` • `Move` • `MoveToDeletedItems` • `SoftDelete` • `HardDelete` • `FolderBind` • `SendAs` • `SendOnBehalf` By default, the `Update`, `SoftDelete`, `HardDelete`,

			and `SendAs` actions performed by delegates are logged. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. **Note:** The *AuditEnabled* parameter must be set to `$true` to enable logging.
AuditEnabled	Optional	System.Boolean	The *AuditEnabled* parameter specifies whether to enable or disable mailbox audit logging. If auditing is enabled, actions specified in the *AuditAdmin*,

			AuditDelegate, and *AuditOwner* parameters are logged. Valid values include: • `$true` Mailbox audit logging is enabled. • `$false` Mailbox audit logging is disabled. The default value is `$false`.
AuditLogAgeLimit	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *AuditLogAgeLimit* parameter specifies the period for which audit logs for a mailbox are retained. Logs older than the specified period are removed. The default value is 90 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify 10 days for this parameter, use 10.00:00:00. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of this parameter to 00:00:00 prevents the automatic

			removal of server statistics log files.
AuditOwner	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *AuditOwner* parameter specifies the operations to log for mailbox owners. Valid values include: • `None` • `Update` • `Move` • `MoveToDeletedItems` • `SoftDelete` • `HardDelete` By default, mailbox access by the owner isn't logged. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax:

			`@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. **Note:** The *AuditEnabled* parameter must be set to `$true` to enable logging.
BypassLiveId	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
BypassModerationFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassModerationFromSendersOrMembers* parameter specifies the recipients whose messages bypass moderation when sending to this mailbox. You can use any of the following values: • DN • Canonical name • GUID • Name • Display name • LegacyExchangeDN • Primary SMTP email address. By default, this parameter is blank, which ensures that all messages are moderated when this mailbox is configured for

			moderation. Senders designated as moderators for this mailbox are never moderated. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
CalendarLoggingQuota	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *CalendarLoggingQuota* parameter specifies the quota in the Recoverable

			Items folder that's used to store logs generated when changes are made to a calendar item. When the mailbox exceeds this quota, calendar logging is disabled until the Messaging records management feature removes older calendar logs to free more space. The default quota size is 6 GB.
CalendarRepairDisabled	Optional	System.Boolean	The *CalendarRepairDisabled* parameter specifies that this mailbox won't have its calendar items repaired by the Calendar Repair Assistant. The two possible values for this parameter are `$true` or `$false`. The default is `$false`.
CalendarVersionStoreDisabled	Optional	System.Boolean	The *CalendarVersionStoreDisabled* parameter specifies that calendar changes in a user's mailbox aren't logged. The two possible values for this parameter are `$true` or `$false`. The

			default value is `$false`. By default, all changes to a calendar item are recorded in the mailbox of a user to keep older versions of meeting items.
ClientExtensions	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *ClientExtensions* parameter specifies whether the organization-wide client extensions (also called *Apps for Outlook*) will be installed in the specified arbitration mailbox. Only one arbitration mailbox (also called the *organization mailbox*) in the organization can be configured to store client extensions. The two possible values for this parameter are `$true` or `$false`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CreateDTMFMap	Optional	System.Boolean	The *CreateDTMFMap* parameter specifies that a dual tone multi-frequency (DTMF) map be created for the user.
CustomAttribute1	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute10	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute11	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.

CustomAttribute12	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute13	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute14	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute15	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute2	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify

			custom attributes. You can use these attributes to store additional information.
CustomAttribute3	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute4	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute5	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute6	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional

			information.
CustomAttribute7	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute8	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute9	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
Database	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *Database* parameter specifies the database that contains the mailbox object. You can use one of the following values: • GUID

			• Database name
DefaultPublicFolderMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *DefaultPublicFolderMailbox* parameter assigns a specific public folder mailbox to the user. By default, the public folder mailbox used by a user is automatically selected by an algorithm that load-balances users across all public folder mailboxes.
DeliverToMailboxAndForward	Optional	System.Boolean	The *DeliverToMailboxAndForward* parameter specifies whether messages sent to this mailbox are forwarded to another address. If the *DeliverToMailboxAndForward* parameter is set to `$true`, messages are delivered to the mailbox and to the forwarding address. If set to `$false`, messages are delivered only to the forwarding address.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the

			display name for the user account associated with this mailbox. The display name is used by Microsoft Outlook.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
DowngradeHighPriorityMessagesEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *DowngradeHighPriorityMessagesEnabled* parameter specifies whether to prevent the mailbox from sending high priority messages to an X.400 mail system. If this parameter is set to `$true` and the mailbox sends a high priority message destined to an

			X.400 mail system, the message priority is changed to normal priority.
DumpsterMessagesPerFolderCountReceiveQuota	Optional	System.Int32	The *DumpsterMessagesPerFolderCountReceiveQuota* parameter specifies the maximum number of messages that can be contained in each folder in the Recoverable Items folder (called *the dumpster* in previous versions of Exchange). When a folder exceeds this limit, it can't store new messages. For example, if the Deletions folder in the Recoverable Items folder has exceeded the message count limit and the mailbox owner attempts to permanently delete items from their mailbox, the deletion will fail.
DumpsterMessagesPerFolderCountWarningQuota	Optional	System.Int32	The *DumpsterMessagesPerFolderCountWarningQuota* parameters specifies the number of messages that

			each folder in the Recoverable Items folder (called *the dumpster* in previous versions of Exchange) can hold before Exchange sends a warning message to the mailbox owner and logs an event to the application event log. When this quota is reached, warning messages and logged events occur once a day.
EmailAddresses	Optional	Microsoft.Exchange.Data.ProxyAddressCollection	The *EmailAddresses* parameter specifies all the proxy addresses of the mailbox. It includes the primary SMTP address as one of the proxy addresses. If you use this parameter, you can't use the *PrimarySmtpAddress* parameter. **Important:** Exchange doesn't validate custom addresses for proper formatting. You must ensure that the custom address you specify complies with the format requirements for that address type. Because X.400 addresses are considered custom

			addresses in Exchange, they also aren't validated, and you must provide the correct syntax when specifying an X.400 address.
EmailAddressPolicyEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *EmailAddressPolicyEnabled* parameter specifies whether the email address policy for this mailbox is enabled. The two possible values for this parameter are `$true` or `$false`.
EnableRoomMailboxAccount	Optional	System.Boolean	Use the *EnableRoomMailboxAccount* parameter to enable or disable the Active Directory user account that corresponds to a room mailbox. Use the `$true` value to enable the user account or use `$false` to disable a logon-enabled user account. To logon-enable a user account, you also have to use the *RoomMailboxPassword* parameter to specify a password for the user

			account. Logon-enabled Active Directory user accounts for room mailboxes are required for solutions such as the Lync Room System. In Exchange Online, a logon-enabled room mailbox doesn't require a license. **Important:** After configuring a logon-enabled room mailbox using the *EnableRoomMailboxAccount* and *RoomMailboxPassword* parameters in an on-premises Exchange organization, you have to enable the corresponding user account in Active Directory Users and Computers or by running the **Enable-ADAccount** cmdlet in Windows PowerShell.
EndDateForRetentionHold	Optional	System.DateTime	The *EndDateForRetentionHold* parameter specifies the end date for retention hold for messaging records management (MRM). To use this parameter, the *RetentionHoldEnabled* parameter must be set to

			`$true`.
EvictLiveId	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ExtendedPropertiesCountQuota	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *ExtendedPropertiesCountQuota* property is used to configure the Named Properties and NonMAPI Named Properties quotas for a mailbox. This should typically only be done if you are experiencing QuotaExceededException or MapiExceptionNamedPropsQuotaExceeded errors.
ExtensionCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute*

			parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *ExtensionCustomAttribute*

		y	*1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute 1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information

			about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExternalOofOptions	Optional	Microsoft.Exchange.Data.Directory.Recipient.ExternalOofOptions	The *ExternalOofOptions* parameter specifies the option for sending an Out of Office message to external senders. You can

			use the following values: • `External` • `InternalOnly`
FederatedIdentity	Optional	System.String	This parameter is available only in the cloud-based service. The *FederatedIdentity* parameter associates an on-premises Active Directory user with a Microsoft Office user.
FolderHierarchyChildrenCountReceiveQuota	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *FolderHierarchyChildrenCountReceiveQuota* parameter specifies the maximum number of subfolders that can be created in a mailbox folder. The mailbox owner won't be able to create a new subfolder when this limit is reached.
FolderHierarchyChildrenCountWarningQuota	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *FolderHierarchyChildrenCountWarningQuota* parameter specifies the

			number of subfolders that can be created in a mailbox folder before Exchange sends a warning message to the mailbox owner and logs an event to the application event log. When this quota is reached, warning messages and logged events occur once a day.
FolderHierarchyDepthReceiveQuota	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *FolderHierarchyDepthReceiveQuota* parameter specifies the maximum number of levels in the folder hierarchy of a mailbox folder. The mailbox owner won't be able to create another level in the folder hierarchy of the mailbox folder when this limit is reached.
FolderHierarchyDepthWarningQuota	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The

			FolderHierarchyDepthWarningQuota parameter specifies the number of levels in the folder hierarchy of a mailbox folder that can be created before Exchange sends a warning message to the mailbox owner and logs an event to the application event log. When this quota is reached, warning messages and logged events occur once a day.
FoldersCountReceiveQuota	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *FoldersCountReceiveQuota* parameter is used to specify a maximum number of folders within a mailbox, typically a public folder mailbox. If this value is configured and the limit is reached, no new folders will be able to be created.
FoldersCountWarningQuota	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013.

			The *FoldersCountWarningQuota* parameter is used to display a warning message that the folder hierarchy is full when the value specified for this parameter is reached. This parameter is typically used for public folder mailboxes.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress the warning or confirmation messages that appear during specific configuration changes.
ForestWideDomainControllerAffinityByExecutingUser	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ForwardingAddress	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *ForwardingAddress* parameter specifies a forwarding address.
ForwardingSmtpAddress	Optional	Microsoft.Exchange.Data.ProxyAddress	The *ForwardingSmtpAddress* parameter specifies a forwarding SMTP address.
GMGen	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013.

			The *GMGen* parameter specifies the arbitration mailbox (also called the *organization mailbox*) for group metrics generation for the organization. In MailTips, group metrics information is used to indicate the number of recipients a message will be sent to or whether recipients are outside your organization. The two possible values for this parameter are `$true` or `$false`.
GrantSendOnBehalfTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *GrantSendOnBehalfTo* parameter specifies the DN of other mailboxes that can send messages on behalf of this mailbox.
HiddenFromAddressListsEnabled	Optional	System.Boolean	The *HiddenFromAddressListsEnabled* parameter specifies whether this mailbox is hidden from other address lists. The two possible values for this parameter are `$true` or `$false`.

IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.

ImListMigrationCompleted	Optional	System.Boolean	The *ImListMigrationCompleted* parameter specifies whether a UM-enabled user's Microsoft Lync 2013 contacts have been successfully moved from their Exchange 2013 mailbox to a Lync 2013 server. This indicates the mailbox can be safely migrated from Exchange 2013 back to Exchange 2010 or Exchange 2007. The value `$false` indicates that the user's Lync contacts have not been moved. The default value is `$false`. Lync 2013 supports storing the user's Lync contacts in their Exchange 2013 mailbox. This feature is known as the *unified contact store*. Exchange 2010 and Exchange 2007 don't support the unified contact store. Before you migrate a user's Exchange 2013 mailbox back to Exchange 2010 or Exchange 2007, the Lync administrator needs to

			move the user's Lync contacts from the unified contact store back to a Lync 2013 server. **Note:** You shouldn't migrate a mailbox back to Exchange 2010 or Exchange 2007 while the user's Lync contacts are stored in the unified contact store. If you do, the user could permanently lose access to their Lync contacts. After you contact the Lync administrator and verify that the user's Lync contacts have been moved back to the Lync server, you should be able to complete the migration. If you need to migrate the mailbox despite the potential data loss, you can set the *ImListMigrationCompleted* parameter to `$false`.
ImmutableId	Optional	System.String	The *ImmutableId* parameter is used by Outlook Live Directory Sync (OLSync) and specifies a unique and immutable identifier in the form of an SMTP address for an Exchange mailbox that's used for federated delegation when requesting Security Assertion Markup

			Language (SAML) tokens. If federation is configured for this mailbox and you don't set this parameter when you create the mailbox, Exchange will create the value for the immutable ID based upon the mailbox's **ExchangeGUID** and the federated account namespace, for example, 7a78e7c8-620e-4d85-99d3-c90d90f29699@mail.contoso.com. You must set the *ImmutableId* parameter if Active Directory Federation Services (AD FS) is deployed to allow single sign-on into off-premises mailboxes and AD FS is configured to use a different attribute than **ExchangeGUID** for sign-on token requests. Both, Exchange and AD FS must request the same token for the same user to ensure proper functionality for a cross-premise Exchange

			deployment scenario.
IsExcludedFromServingHierarchy	Optional	System.Boolean	The *IsExcludedFromServingHierarchy* parameter prevents users from accessing the public folder hierarchy on the specified public folder mailbox. For load-balancing purposes, users are equally distributed across public folder mailboxes by default. When this parameter is set on a public folder mailbox, that mailbox isn't included in this automatic load-balancing and won't be accessed by users to retrieve the public folder hierarchy. However, if an administrator has set the *DefaultPublicFolderMailbox* property on a user mailbox to a specific public folder mailbox, the user will still access the specified public folder mailbox even if the *IsExcludedFromServingHierarchy* parameter is set for that public folder mailbox.

			Note: This parameter should only be used during public folder migrations. Do not use this parameter once the initial migration validation is complete.
IsHierarchyReady	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
IssueWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *IssueWarningQuota* parameter specifies the mailbox size at which a warning message is sent to the user. You must specify either an integer or unlimited. If you set this attribute on a mailbox, that mailbox setting overrides the value set for this attribute on the mailbox database.
JournalArchiveAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is reserved for internal Microsoft use.
Languages	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Languages* parameter specifies the language preferences for this mailbox, in order of preference. Several Exchange components display information to a mailbox user in the preferred language, if that language is supported.

			Some of those components include quota messages, non-delivery reports (NDRs), the Outlook Web App user interface, and Unified Messaging (UM) voice prompts. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
LinkedCredential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *LinkedCredential* parameter specifies

			credentials to use to access the domain controller specified by the *LinkedDomainController* parameter. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
LinkedDomainController	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *LinkedDomainController* parameter specifies the domain controller in the forest where the user account resides, if this mailbox is a linked mailbox. The domain controller in the forest where the user account resides is used to get security information for the account specified by the *LinkedMasterAccount* parameter.

LinkedMasterAccount	Optional	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	This parameter is available only in on-premises Exchange 2013. The *LinkedMasterAccount* parameter specifies the master account in the forest where the user account resides, if this mailbox is a linked mailbox. The master account is the account to which the mailbox links. The master account grants access to the mailbox. You can use one of the following values: • GUID • DN • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
LitigationHoldDate	Optional	System.DateTime	The *LitigationHoldDate* parameter specifies the date when the mailbox is placed on litigation hold. The parameter is populated automatically when placing a mailbox on litigation hold. This can be used for informational

			or reporting purposes. **Note:** When using the Exchange Management Shell to place the mailbox on litigation hold, you can optionally specify any date as the *LitigationHoldDate*, but the mailbox is placed on litigation hold when the cmdlet is run.
LitigationHoldDuration	Optional	Microsoft.Exchange.Data.Unlimited	The *LitigationHoldDuration* parameter specifies how long the mailbox will be on litigation hold. Use days to specify the duration. **Tip:** You have to use the *IncludeLitigationHoldDuration* parameter with the **Get-Mailbox** cmdlet to view the value of the litigation-hold duration.
LitigationHoldEnabled	Optional	System.Boolean	The *LitigationHoldEnabled* parameter specifies that the mailbox is under litigation hold and that messages can't be deleted from the user's account. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`. After a mailbox is placed

			on litigation hold, deleted items and all versions of changed items are retained in the Recoverable Items folder. Items that are purged from the dumpster are also retained and the items are held indefinitely. If you enable litigation hold, single-item recovery quotas aren't applied.
LitigationHoldOwner	Optional	System.String	The *LitigationHoldOwner* parameter specifies the user who placed the mailbox on litigation hold. The parameter is populated automatically when placing a mailbox on litigation hold. This can be used for informational and reporting purposes. **Note:** When using the Shell to place a mailbox on litigation hold, you can optionally specify a string value for this parameter.
MailboxContainerGuid	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
MailboxMessagesPerFolderCountReceiveQuota	Optional	System.Int32	The *MailboxMessagesPerFolderCountReceiveQuota*

			parameter specifies the maximum number of messages for a mailbox folder. When this limit is reached, the folder can't receive new messages.
MailboxMessagesPerFolderCountWarningQuota	Optional	System.Int32	The *MailboxMessagesPerFolderCountWarningQuota* parameter specifies the number of messages that a mailbox folder can hold before Exchange sends a warning message to the mailbox owner and logs an event to the application event log. When this quota is reached, warning messages and logged events occur once a day.
MailboxPlan	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPlanIdParameter	This parameter is reserved for internal Microsoft use.
MailboxProvisioningConstraint	Optional	Microsoft.Exchange.Data.Directory.MailboxProvisioningConstraint	This parameter is reserved for internal Microsoft use.
MailboxProvisioningPreferences	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
MailRouting	Optional	System.Boolean	This parameter is reserved

			for internal Microsoft use.
MailTip	Optional	System.String	The *MailTip* parameter specifies the message displayed to senders when they start drafting an email message to this recipient. The *MailTip* parameter message must be less than or equal to 250 characters.
MailTipTranslations	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *MailTipTranslations* parameter specifies additional languages when you want to provide the *MailTip* parameter information for this recipient in multiple languages. For each language, you must provide the locale, followed by a colon and the specific *MailTip* parameter message in that language. Each *MailTip* parameter message must be less than or equal to 250 characters. Multiple languages can be separated by commas. To enter multiple values and overwrite any existing

			entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
ManagedFolderMailboxPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	This parameter is available only in on-premises Exchange 2013. The *ManagedFolderMailboxPolicy* parameter specifies a managed folder mailbox policy that controls MRM for the mailbox. If the parameter is set to `$null`, Exchange removes the managed folder mailbox policy from the mailbox but any managed folders in the mailbox remain.

ManagedFolderMailboxPolicyAllowed	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ManagedFolderMailboxPolicyAllowed* parameter bypasses the warning that MRM features aren't supported for email clients running versions of Outlook earlier than Office Outlook 2007. When a managed folder mailbox policy is assigned to a mailbox using the *ManagedFolderMailboxPolicy* parameter, the warning appears by default unless the *ManagedFolderMailboxPolicyAllowed* parameter is used. **Note:** Although Outlook 2003 Service Pack 3 clients are supported, they have limited MRM functionality.
Management	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *Management* parameter specifies the arbitration mailbox (also

			call an *organization mailbox*) used to manage mailbox moves and mailbox migrations. The two possible values for this parameter are `$true` or `$false`.
MaxBlockedSenders	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *MaxBlockedSenders* parameter specifies the maximum number of senders that can be included in the blocked senders list. Blocked senders are senders that are considered junk senders by the mailbox user and are used in junk email rules. This parameter is validated only when the junk email rules are updated using Outlook Web App or web services.
MaxReceiveSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxReceiveSize* parameter specifies the

			maximum size of messages that this mailbox can receive. You must specify either an integer or `unlimited`.
MaxSafeSenders	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *MaxSafeSenders* parameter specifies the maximum number of senders that can be included in the safe senders list. Safe senders are senders that are trusted by the mailbox user and are used in junk email rules. This parameter is only validated when the junk email rules are updated using cloud-based organizations or services.
MaxSendSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxSendSize* parameter specifies the maximum size of messages that this mailbox can send. You

			must specify either an integer or `unlimited`.
MessageTracking	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *MessageTracking* parameter designates the specified arbitration mailbox (also called an *organization mailbox*) as the anchor mailbox for cross-organizational message tracking scenarios. By default, the message tracking organizational capability is assigned to the arbitration mailbox named SystemMailbox{bb558c35-97f1-4cb9-8ff7-d53741dc928c}. Values for this parameter are either `$true` or `$false`.
MessageTrackingReadStatusEnabled	Optional	System.Boolean	The *MessageTrackingReadStatusEnabled* parameter specifies that this mailbox can view the read status of sent messages. The two

			possible values for this parameter are `$true` or `$false`. If you set this parameter to `$false`, the read status won't be displayed to senders who view delivery reports for messages they send to this user and only the time that the message was delivered to the mailbox is available. The default value is `$true`.
MicrosoftOnlineServicesID	Optional	Microsoft.Exchange.Data.SmtpAddress	The *MicrosoftOnlineServicesID* parameter specifies the user ID for the object. This parameter only applies to objects in the cloud-based service. It isn't available for on-premises deployments.
Migration	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users who are responsible for moderating the messages sent to this mailbox. To designate more than one user, separate the users with

			commas. This parameter is required if you set the *ModerationEnabled* parameter to `$true`. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable moderation for the mailbox. To enable moderation, set this parameter to `$true`. To disable moderation, set

			this parameter to `$false`. The default value is `$false`.
Name	Optional	System.String	The *Name* parameter specifies the **Name** attribute for this mailbox. The **Name** attribute is used for the common name in Active Directory.
NetID	Optional	Microsoft.Exchange.Data.NetID	This parameter is reserved for internal Microsoft use.
NewPassword	Optional	System.Security.SecureString	This parameter is available only in on-premises Exchange 2013. The *NewPassword* parameter is used when an end-user changes their password in Outlook Web App. Administrators use the *Password* parameter to reset an end-user's password.
OABGen	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *OABGen* parameter specifies the arbitration mailbox (also called an *organization mailbox*) used for offline address

			book (OAB) file generation and storage for the organization. OAB requests are sent to the server where this arbitration mailbox is located. The two possible values for this parameter are `$true` or `$false`.
Office	Optional	System.String	The *Office* parameter specifies the Microsoft Office attribute for this mailbox.
OfflineAddressBook	Optional	Microsoft.Exchange.Configuration.Tasks.OfflineAddressBookIdParameter	This parameter is available only in on-premises Exchange 2013. The *OfflineAddressBook* parameter specifies the associated OAB.
OldPassword	Optional	System.Security.SecureString	This parameter is available only in on-premises Exchange 2013. The *OldPassword* parameter specifies the existing password for a mailbox. You have to include this parameter when using the **Set-Mailbox** cmdlet to directly change the

			password for a mailbox. To reset a password without having to specify the old password, you must be assigned the Reset Password role.
OMEncryption	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
OriginalNetID	Optional	Microsoft.Exchange.Data.NetID	This parameter is reserved for internal Microsoft use.
Password	Optional	System.Security.SecureString	The *Password* parameter resets the password of the user account associated with the mailbox. To reset a password without having to include the *OldPassword* parameter, you must be assigned the Reset Password role.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in on-premises Exchange 2013. The *PrimarySmtpAddress* parameter specifies the address that external users see when they receive a message from this mailbox. If you use this parameter, you can't use the

			EmailAddresses parameter because the *EmailAddresses* parameter includes the primary SMTP address.
ProhibitSendQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *ProhibitSendQuota* parameter specifies the mailbox size at which the user associated with this mailbox can no longer send messages. You must specify either an integer or `unlimited`. If you set this attribute on a mailbox, that mailbox setting overrides the value set for this attribute on the mailbox database.
ProhibitSendReceiveQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *ProhibitSendReceiveQuota* parameter specifies the mailbox size at which the user associated with this mailbox can no longer send or receive messages. You must specify either an integer or `unlimited`. If you set this attribute on a mailbox, that mailbox setting overrides the value set for this attribute on

			the mailbox database.
PstProvider	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
PublicFolder	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *PublicFolder* parameter specifies that the target mailbox is a public folder mailbox. This parameter is required when you use the **Set-Mailbox** cmldet to modify public folder mailboxes. Public folder mailboxes are specially designed mailboxes to store the hierarchy and content of public folders.
QueryBaseDN	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	This parameter is reserved for internal Microsoft use.
QueryBaseDNRestrictionEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RecipientLimits	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *RecipientLimits* parameter specifies the

			maximum number of recipients per message to which this mailbox can send. You must specify either an integer or `unlimited`. If you set this attribute on a mailbox, that mailbox setting overrides the value set for this attribute in the Transport service.
RecoverableItemsQuota	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *RecoverableItemsQuota* parameter specifies the limit for the Recovery Items folder. When you reach the quota limit, you can't put any more items in the Recovery Items folder.
RecoverableItemsWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *RecoverableItemsWarningQuota* parameter specifies the quota for when a warning event is entered

			in Event Viewer.
RejectMessagesFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFrom* parameter specifies the recipients from whom messages are rejected. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
RejectMessagesFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromDLMembers* parameter specifies distribution lists. Messages from any member of these distribution lists are

			rejected. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
RejectMessagesFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromSendersOrMembers* parameter specifies the identity of recipients from whom messages are rejected. You can use any of the following values for the valid senders: • DN • Canonical name • GUID

			• Name • Display name • Alias • Exchange DN • Primary SMTP email address To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
RemoteAccountPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.RemoteAccountPolicyIdParameter	This parameter is reserved for internal Microsoft use.
RemoteRecipientType	Optional	Microsoft.Exchange.Data.Directory.Recipient.	This parameter is reserved for internal Microsoft use.

		RemoteRecipientType	
RemoveManagedFolderAndPolicy	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemoveManagedFolderAndPolicy* parameter specifies whether to remove all MRM 1.0 policies and attributes from a mailbox. If you use this parameter, MRM 1.0 policies and MRM 1.0 properties from any managed folders that were created as part of any MRM 1.0 policies are removed. Managed folders that are empty are also removed from the mailbox, and managed folders that contain items are converted to standard folders.
RemovePicture	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemovePicture* switch specifies whether to remove the picture that a user has added to a

			mailbox. A picture file can be added to the mailbox by using the **Import-RecipientDataProperty** cmdlet.
RemoveSpokenName	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemoveSpokenName* switch specifies whether to remove the spoken name that a user has added to a mailbox. A sound file can be added to the mailbox by using the **Import-RecipientDataProperty** cmdlet.
RequireSecretQA	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RequireSenderAuthenticationEnabled	Optional	System.Boolean	The *RequireSenderAuthenticationEnabled* parameter specifies whether senders must be authenticated. The two possible values for this parameter are `$true` or `$false`.
ResetPasswordOnNextLogon	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013.

			The *ResetPasswordOnNextLogon* parameter specifies whether to require users to change their password the next time they sign in to their mailbox. If the *ResetPasswordOnNextLogon* parameter is set to `$true`, it requires users to change their password the next time they sign in to their mailbox. Valid input for this parameter is `$true` or `$false`.
ResourceCapacity	Optional	System.Int32	The *ResourceCapacity* parameter specifies capacity, if this mailbox is a resource mailbox. You must specify a non-negative integer.
ResourceCustom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ResourceCustom* parameter specifies additional information about the resource. You can define custom properties for resource mailboxes using the **Set-ResourceConfig** command and use this parameter to set those custom properties.

			To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
RetainDeletedItemsFor	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *RetainDeletedItemsFor* parameter specifies the length of time to keep deleted items. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify a 15-hour interval, enter

			15:00:00.
RetainDeletedItemsUntilBackup	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *RetainDeletedItemsUntilBackup* parameter specifies whether to retain deleted items until the next backup. The two possible values for this parameter are `$true` or `$false`.
RetentionComment	Optional	System.String	The *RetentionComment* parameter specifies a comment displayed in Outlook regarding the user's retention hold status. This comment can only be set if the *RetentionHoldEnabled* parameter is set to `$true`. This comment should be localized to the user's preferred language.
RetentionHoldEnabled	Optional	System.Boolean	The *RetentionHoldEnabled* parameter specifies whether retention hold is enabled for messaging retention policies. The two

			possible values for this parameter are `$true` or `$false`. To set the start date for retention hold, use the *StartDateForRetentionHold* parameter.
RetentionPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *RetentionPolicy* parameter specifies the name of a retention policy that you want applied to this mailbox. Retention policies consist of tags that are applied to mailbox folders and mail items to determine the period of time that the items should be retained.
RetentionUrl	Optional	System.String	The *RetentionUrl* parameter specifies the URL or an external web page with additional details about the organization's messaging retention policies. This URL can be used to expose details regarding retention policies in general, which is usually a customized legal or IT website for the company.

RoleAssignmentPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *RoleAssignmentPolicy* parameter specifies the management role assignment policy to assign to the mailbox when it's created or enabled. If you don't include this parameter when you create or enable a mailbox, the default assignment policy is used. If the assignment policy name contains spaces, enclose the name in quotation marks ("). If you don't want to assign an assignment policy when a mailbox is created or enabled, specify a value of `$null`. For more information, see Understanding management role assignment policies.
RoomMailboxPassword	Optional	System.Security.SecureString	Use the *RoomMailboxPassword* parameter to specify a password when you're using the *EnableRoomMailboxAccount* parameter to enable the Active Directory user

			account for a room mailbox. Use the following format: -`RoomMailboxPassword (ConvertTo-SecureString -String` *`password`* `-AsPlainText -Force)`. To set the password when enabling the Active Directory user account for a room mailbox, you must be assigned the Reset Password role.
RulesQuota	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *RulesQuota* parameter specifies the limit for the size of rules for this mailbox. When you enter a value, qualify the value with one of the following: • B (bytes) • KB (kilobytes) Unqualified values are treated as bytes. The default value for this parameter is 64 KB. The maximum value is 256 KB. **Note:** The quota for mailbox rules applies only to enabled rules. There is no restriction on the number of disabled rules a mailbox can have. However, the total size of

			rules that are enabled or active can't exceed the value specified for this parameter.
SamAccountName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *SamAccountName* parameter specifies the user name for earlier operating systems such as Microsoft Windows NT 4.0, Windows 98, Windows 95, and LAN Manager. This parameter is used to support clients and servers running older versions of the operating system. This attribute must be less than 20 characters in length.
SCLDeleteEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *SCLDeleteEnabled* parameter specifies whether messages that meet the spam confidence level (SCL) threshold specified by the *SCLDeleteThreshold* parameter are deleted. You can use the following

			values: • `$true` • `$false`
SCLDeleteThreshold	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *SCLDeleteThreshold* parameter specifies the SCL at which a message is deleted, if the *SCLDeleteEnabled* parameter is set to `$true`. You must specify an integer from 0 through 9 inclusive.
SCLJunkEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *SCLJunkEnabled* parameter specifies whether messages that meet the SCL threshold specified by the *SCLJunkThreshold* parameter are moved to the Junk Email folder. You can use the following values: • `$true` • `$false`
SCLJunkThreshold	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013.

			The *SCLJunkThreshold* parameter specifies the SCL threshold. Messages with an SCL greater than the value that you specify for the *SCLJunkThreshold* parameter are moved to the Junk Email folder, if the *SCLJunkEnabled* parameter is set to `$true`. You must specify an integer from 0 through 9 inclusive.
SCLQuarantineEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *SCLQuarantineEnabled* parameter specifies whether messages that meet the SCL threshold specified by the *SCLQuarantineThreshold* parameter are quarantined. If a message is quarantined, it's sent to the quarantine mailbox where the messaging administrator can review it. You can use the following values: • `$true` • `$false`

SCLQuarantineThreshold	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *SCLQuarantineThreshold* parameter specifies the SCL at which a message is quarantined, if the *SCLQuarantineEnabled* parameter is set to `$true`. You must specify an integer from 0 through 9 inclusive.
SCLRejectEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *SCLRejectEnabled* parameter specifies whether messages that meet the SCL threshold specified by the *SCLRejectThreshold* parameter are rejected. If a message is rejected, it's deleted and a rejection response is sent to the sender. You can use the following values: • `$true` • `$false`
SCLRejectThreshold	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013.

			The *SCLRejectThreshold* parameter specifies the SCL at which a message is rejected, if the *SCLRejectEnabled* parameter is set to `$true`. You must specify an integer from 0 through 9 inclusive.
SecondaryAddress	Optional	System.String	The *SecondaryAddress* parameter specifies the secondary address used by the UM-enabled user.
SecondaryDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *SecondaryDialPlan* parameter specifies a secondary UM dial plan to use. This parameter is provided to create a secondary proxy address.
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when a message they sent to the moderated distribution group is rejected by one of the moderators. You can specify one of the following values: • `Always`

			• `Internal` • `Never` Set this parameter to `Always` if you want notifications to be sent to all senders. Set this parameter to `Internal` if you want notifications to be sent to only the senders that are internal to your organization. Set this parameter to `Never` to disable all status notifications. The default value is `Never`.
SharingPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.SharingPolicyIdParameter	The *SharingPolicy* parameter specifies the sharing policy associated with this mailbox.
SimpleDisplayName	Optional	System.String	The *SimpleDisplayName* parameter is used to display an alternative description of the object when only a limited set of characters is permitted. This limited set of characters consists of ASCII characters 26 through 126, inclusively.
SingleItemRecoveryEn	Optional	System.Boolean	The

abled			*SingleItemRecoveryEnabled* parameter specifies whether to prevent the Recovery Items folder from being purged. When this parameter is set to `$true`, it prevents the Recovery Items folder from being purged. It prevents any items from being removed that have been deleted or edited. The possible values for this parameter are `$true` or `$false`. The default value is `$false`.
SkipMailboxProvisioningConstraintValidation	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
SKUAssigned	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
SKUCapability	Optional	Microsoft.Exchange.Data.Directory.Capability	This parameter is reserved for internal Microsoft use.
StartDateForRetentionHold	Optional	System.DateTime	The *StartDateForRetentionHold* parameter specifies the start date for retention hold for MRM. To use this parameter, the *RetentionHoldEnabled* parameter must be set to

			`$true`.
SuiteServiceStorage	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TenantUpgrade	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ThrottlingPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.ThrottlingPolicyIdParameter	This parameter is available only in on-premises Exchange 2013. The *ThrottlingPolicy* parameter specifies the identity of the throttling policy for this mailbox.
Type	Optional	Microsoft.Exchange.Management.RecipientTasks.ConvertibleMailboxSubType	The *Type* parameter specifies the type for the mailbox. You can use the following values: • `Regular` • `Room` • `Equipment` • `Shared`
UMDataStorage	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *UMDataStorage* parameter specifies the arbitration mailbox (also called the *organization mailbox*) used to store UM call data records and UM custom prompts. This capability can be assigned to only one arbitration

			mailbox for the organization. The two possible values for this parameter are `$true` or `$false`.
UMDtmfMap	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UMDtmfMap* parameter specifies whether you want to create a user-defined DTMF map for the UM-enabled user.
UMGrammar	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *UMGrammar* parameter specifies the arbitration mailbox (also called the *organization mailbox*) for UM directory speech grammar generation for the organization. UM directory speech grammars will be generated and used on the Mailbox server of this arbitration mailbox. UM directory speech grammars are used in speech-enabled directory search features, such as UM auto attendants.

			The two possible values for this parameter are `$true` or `$false`.
UnifiedMailbox	Optional	Microsoft.Exchange.Data.Directory.CrossTenantObjectId	This parameter is reserved for internal Microsoft use.
UsageLocation	Optional	Microsoft.Exchange.Data.Directory.CountryInfo	This parameter is reserved for internal Microsoft use.
UseDatabaseQuotaDefaults	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *UseDatabaseQuotaDefaults* parameter specifies that this mailbox uses the quota attributes specified for the mailbox database where this mailbox resides. The quota attributes are: • **ProhibitSendQuota** • **ProhibitSendReceiveQuota** • **IssueWarningQuota** • **RulesQuota** The two possible values for this parameter are `$true` or `$false`.
UseDatabaseRetentio	Optional	System.Boolean	This parameter is available only in on-

nDefaults			premises Exchange 2013. The *UseDatabaseRetentionDefaults* parameter specifies that this mailbox uses the **MailboxRetention** attribute specified for the mailbox database where this mailbox resides. The two possible values for this parameter are `$true` or `$false`.
UserCertificate	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UserCertificate* parameter specifies the digital certificate used to sign a user's email messages.
UserPrincipalName	Optional	System.String	The *UserPrincipalName* parameter specifies the UPN for this mailbox. This is the logon name for the user. The UPN consists of a user name and a suffix. Typically, the suffix is the domain name where the user account resides.
UserSMimeCertificate	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *UserSMimeCertificate* parameter specifies the

			SMIME certificate used to sign a user's email messages.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsEmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *WindowsEmailAddress* parameter specifies the Windows email address for this mailbox. This address isn't used by Exchange.
WindowsLiveID	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in the cloud-based service. The *WindowsLiveID* parameter renames the Microsoft account (formerly known as a Windows Live ID) associated with the

			mailbox.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxAutoReplyConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxAutoReplyConfiguration** cmdlet to retrieve Automatic Replies settings for a specific mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxAutoReplyConfiguration -Identity <MailboxIdParameter> [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-
ReadFromDomainController <SwitchParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns Automatic Replies settings for all mailboxes in the Exchange organization.

```
Get-Mailbox | Get-MailboxAutoReplyConfiguration
```

EXAMPLE 2

This example retrieves Automatic Replies settings for Tony's mailbox at contoso.com.

```
Get-MailboxAutoReplyConfiguration -Identity 'contoso.com/
Users/Tony Smith'
```

EXAMPLE 3

This example retrieves all Automatic Replies settings for all mailboxes in the Exchange organization.

```
Get-Mailbox | Get-MailboxAutoReplyConfiguration -ResultSize
unlimited
```

Detailed Description

You can use the **Get-MailboxAutoReplyConfiguration** cmdlet to retrieve all the mailboxes enabled for Automatic Replies. When run, the cmdlet returns Automatic Replies settings for the specified mailbox that include the following:

- Mailbox identity value
- Whether Automatic Replies is enabled, scheduled, or disabled for the mailbox
- Start and end date, time during which Automatic Replies will be sent
- Whether external senders receive Automatic Replies (none, known senders, or all)
- Automatic Replies message to be sent to internal and external senders

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Automatic replies" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies a unique identifier associated with a mailbox. Accepted values for the mailbox are as follows: • GUID • ADObjectID • Distinguished name (DN)

			• *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
Credential	Optional	System.Management.Automation.PSCredential	This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include

			all recipients in the forest, and if you don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Microsoft Exchange.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of settings to return. If you want to return all settings that match the command, use `unlimited` for the value of this parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxAutoReplyConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailboxAutoReplyConfiguration** cmdlet to configure Automatic Replies settings for a specific mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxAutoReplyConfiguration -Identity <MailboxIdParameter> [-
AutoReplyState <Disabled | Enabled | Scheduled>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-EndTime <DateTime>] [-
ExternalAudience <None | Known | All>] [-ExternalMessage <String>] [-
IgnoreDefaultScope <SwitchParameter>] [-InternalMessage <String>] [-
StartTime <DateTime>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures Automatic Replies for Tony's mailbox to be sent between the specified start and end dates and includes an internal message.

```
Set-MailboxAutoReplyConfiguration -Identity tony -
AutoReplyState Scheduled -StartTime "7/10/2012 08:00:00" -
EndTime "7/15/2012 17:00:00" -InternalMessage "Internal
auto-reply message"
```

EXAMPLE 2

This example configures Automatic Replies for Tony's mailbox to be sent and includes an internal and an external message.

```
Set-MailboxAutoReplyConfiguration -Identity tony -
AutoReplyState Enabled -InternalMessage "Internal auto-
reply message." -ExternalMessage "External auto-reply
message."
```

Detailed Description

You can disable Automatic Replies for a specified mailbox or organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Automatic replies" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies a unique identifier associated with a mailbox. You can use any value that uniquely identifies a mailbox. Accepted values for the mailbox are as follows: • GUID • ADObjectID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
AutoReplyState	Optional	Microsoft.Exchange.InfoWorker.Common.OOF.OofState	The *AutoReplyState* parameter specifies whether the mailbox is enabled for Automatic Replies. Accepted values are as follows: • `Enabled` If you use this value, auto-replies are

			sent until the value is changed to `Disabled`. • `Disabled` This is the default value. • `Scheduled` If you use this value, you must also specify the *StartTime* and *EndTime* parameters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EndTime	Optional	System.DateTime	The *EndTime* parameter specifies the end date and time that Automatic Replies are sent for the

			specified mailbox. To use this parameter, the *AutoReplyState* parameter must be set to `Scheduled`.
ExternalAudience	Optional	Microsoft.Exchange.InfoWorker.Common.OOF.ExternalAudience	The *ExternalAudience* parameter specifies whether Automatic Replies are sent to external senders. Accepted values are as follows: • `None` This is the default value. • `Known` • `All`
ExternalMessage	Optional	System.String	The *ExternalMessage* parameter specifies the Automatic Replies message that's sent to external senders or senders outside the organization.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default

			scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
InternalMessage	Optional	System.String	The *InternalMessage* parameter specifies the Automatic Replies message that's sent to internal senders or senders within the organization.
StartTime	Optional	System.DateTime	The *StartTime* parameter specifies the start date and time that Automatic Replies are sent for the specified mailbox. To use this parameter, the *AutoReplyState* parameter must be set to `Scheduled`.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxCalendarFolder

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxCalendarFolder** cmdlet to retrieve the publishing or sharing settings for a specified mailbox calendar folder.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxCalendarFolder -Identity <MailboxFolderIdParameter> [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example returns all provided publishing information for the specified calendar folder in Kai's mailbox. In this example, the *Identity* parameter specifies the mailbox with the alias format.

```
Get-MailboxCalendarFolder -Identity kai:\Calendar
```

EXAMPLE 2

This example returns all provided publishing information for the specified calendar folder in Kai's mailbox. This example also specifies DC1 as the domain controller to retrieve this information from Active Directory.

```
Get-MailboxCalendarFolder -Identity kai:\Calendar -
DomainController DC1
```

EXAMPLE 3

This example returns all provided publishing information for the specified calendar folder in Kai's mailbox. In this example, the *Identity* parameter specifies the mailbox with the *domain\account* format.

```
Get-MailboxCalendarFolder -Identity contoso\kai:\Calendar
```

Detailed Description

The **Get-MailboxCalendarFolder** cmdlet retrieves information for the specified calendar folder. This information includes the calendar folder name, whether the folder is currently published or shared, the start and end range of calendar days published, the level of details published for the calendar, whether the published URL of the calendar can be searched on the web, and the published URL for the calendar.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Calendar configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *Identity* parameter specifies the mailbox and folder path or folder name to the calendar folder that has the publishing settings configured. You can use the following values: • GUID • ADObjectID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxCalendarFolder

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailboxCalendarFolder** cmdlet to configure publishing or sharing settings on a calendar folder of a specified mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxCalendarFolder -Identity <MailboxFolderIdParameter> [-Confirm
[<SwitchParameter>]] [-DetailLevel <AvailabilityOnly | LimitedDetails |
FullDetails | Editor>] [-DomainController <Fqdn>] [-PublishDateRangeFrom
<OneDay | ThreeDays | OneWeek | OneMonth | ThreeMonths | SixMonths |
OneYear>] [-PublishDateRangeTo <OneDay | ThreeDays | OneWeek | OneMonth |
ThreeMonths | SixMonths | OneYear>] [-PublishEnabled <$true | $false>] [-
ResetUrl <SwitchParameter>] [-SearchableUrlEnabled <$true | $false>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the level of details to publish for Kai's shared calendar to `LimitedDetails`, which means limited details are displayed.

```
Set-MailboxCalendarFolder -Identity kai:\Calendar -
DetailLevel LimitedDetails
```

EXAMPLE 2

This example enables the calendar in Kai's mailbox to be searchable on the web.

```
Set-MailboxCalendarFolder -Identity kai:\Calendar -
SearchableUrlEnabled $true
```

Detailed Description

The **Set-MailboxCalendarFolder** cmdlet configures publishing information. The calendar folder can be configured as follows:

- Whether the calendar folder is enabled for publishing
- Range of start and end calendar days to publish
- Level of detail to publish for the calendar
- Whether the published URL of the calendar is enabled for search on the web

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Calendar configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *Identity* parameter specifies the mailbox and folder path or folder name to the calendar folder that has the publishing settings configured. You can use the following values: • GUID • ADObjectID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
Confirm	Optional	System.Management.	The *Confirm* switch

		Automation.SwitchParameter	causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DetailLevel	Optional	Microsoft.Exchange.Data.Storage.DetailLevelEnumType	The *DetailLevel* parameter specifies the level of calendar detail that's published and available to anonymous users. You can use the following values: • `AvailabilityOnly` • `LimitedDetails` • `FullDetails` • `Editor` The default value is `AvailabilityOnly`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

PublishDateRangeFrom	Optional	Microsoft.Exchange.Data.Storage.Management.DateRangeEnumType	The *PublishDateRangeFrom* parameter specifies the number of days of calendar information to publish before the current date. You can use the following values: • `OneDay` • `ThreeDays` • `OneWeek` • `OneMonth` • `ThreeMonths` • `SixMonths` • `OneYear` The default value is `ThreeMonths`.
PublishDateRangeTo	Optional	Microsoft.Exchange.Data.Storage.Management.DateRangeEnumType	The *PublishDateRangeTo* parameter specifies the number of days of calendar information to publish after the current date. You can use the following values: • `OneDay` • `ThreeDays` • `OneWeek` • `OneMonth` • `ThreeMonthS` • `SixMonths` • `OneYear` The default value is `ThreeMonths`.
PublishEnabled	Optional	System.Boolean	The *PublishEnabled* parameter specifies whether the specified calendar should be

			enabled for publishing. The default value is `$true`.
ResetUrl	Optional	System.Management.Automation.SwitchParameter	The *ResetUrl* parameter replaces the existing non-public URL with a new URL for a calendar that has been published without being publicly searchable.
SearchableUrlEnabled	Optional	System.Boolean	The *SearchableUrlEnabled* parameter specifies whether the published calendar URL can be searched on the web. The default value is `$false`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-MailboxDiagnosticLogs

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-18*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Export-MailboxDiagnosticLogs** cmdlet to export diagnostic data from user and system mailboxes in Microsoft Exchange Server 2013 Enterprise.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Export-MailboxDiagnosticLogs -ComponentName <String> -Identity
<MailUserOrGeneralMailboxIdParameter> [-Archive <SwitchParameter>] <COMMON
PARAMETERS>


Export-MailboxDiagnosticLogs -ExtendedProperties <SwitchParameter> -
Identity <MailUserOrGeneralMailboxIdParameter> [-Archive
<SwitchParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-Credential
<PSCredential>] [-DomainController <Fqdn>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example retrieves the out-of-office diagnostic log for the user John Smith.

```
Export-MailboxDiagnosticLogs -ComponentName OOF -Identity
JohnSmith
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
ComponentName	Required	System.String	The *ComponentName* parameter specifies the component for which to retrieve the logs. Any log that was created with the common logging code and is stored in the root of the mailbox works. The cmdlet accepts the following names: • OOF • Sharing • MeetingMessageProcessingAgent • SharingSyncAssistant • MRM • InternetCalendar • Calendar
ExtendedProperties	Required	System.Management.Automation.SwitchParameter	The *ExtendedProperties* parameter specifies whether to retrieve all of the well-known properties from the mailbox table that are useful for troubleshooting.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.Mail	The *Identity* parameter specifies what mailbox the

		UserOrGeneralMailboxIdParameter	diagnostic logs are being retrieved from. The mailboxes can be piped from the **Get-Mailbox** cmdlet.
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* parameter retrieves the diagnostics logs of the archive mailbox instead of the primary mailbox.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is

			created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest and don't use this parameter, it's possible that the user information is read from a global catalog with outdated

			information. If you use this parameter, multiple reads might be necessary to get the information.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxExportRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxExportRequest** cmdlet to view the detailed status of an ongoing export request that was initiated by using the New-MailboxExportRequest cmdlet.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxExportRequest [-AccountPartition <AccountPartitionIdParameter>]
[-Identity <MailboxExportRequestIdParameter>] [-Organization
<OrganizationIdParameter>] <COMMON PARAMETERS>


Get-MailboxExportRequest [-AccountPartition <AccountPartitionIdParameter>]
[-BatchName <String>] [-HighPriority <$true | $false>] [-Mailbox
<MailboxOrMailUserIdParameter>] [-Name <String>] [-Organization
<OrganizationIdParameter>] [-RequestQueue <DatabaseIdParameter>] [-Status
<None | Queued | InProgress | AutoSuspended | CompletionInProgress |
Synced | Completed | CompletedWithWarning | Suspended | Failed>] [-Suspend
<$true | $false>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns the status of the ongoing export request with the identity tony\DB01toPST.

```
Get-MailboxExportRequest -Identity "tony\DB01toPST"
```

EXAMPLE 2

This example returns the status of export requests in the Attachment_CompanyReport batch that completed.

```
Get-MailboxExportRequest -BatchName
"Attachment_CompanyReport" -Status Completed
```

EXAMPLE 3

This example returns all export requests that have the name DB01toPST where the export has been suspended.

```
Get-MailboxExportRequest -Name "DB01toPST" -Suspend $true
```

Detailed Description

The search criteria for the **Get-MailboxExportRequest** cmdlet is a Boolean **And** statement. If you use multiple parameters, you narrow your search and reduce your search results.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
BatchName	Optional	System.String	The *BatchName* parameter specifies the name given to a batch export request. You can't use this parameter in conjunction with the *Identity* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
HighPriority	Optional	System.Boolean	The *HighPriority* parameter specifies that the cmdlet returns requests that were created with the *HighPriority* flag. The *HighPriority* flag indicates that the request should be processed before other lower priority requests in the queue. You can't use this parameter in conjunction with the *Identity* parameter.
Identity	Optional	Microsoft.Exchange.MailboxReplicationService.MailboxExportRequestIdParameter	The *Identity* parameter specifies the identity of the export request. By default, export requests are named <*alias*>\MailboxExport*X* (where *X* = 0–9). If you specified a name for the export request when the request was created using the **New-MailboxExportRequest**

			cmdlet, use the following syntax: <*alias*>\<*name*>. Microsoft Exchange Server 2013 automatically precedes the request with the mailbox's alias. This parameter can't be used in conjunction with the following parameters: • *BatchName* • *Mailbox* • *Name* • *Status* • *Suspend* • *HighPriority*
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxOrMailUserIdParameter	The *Mailbox* parameter specifies the identity of the mailbox or mail user from which contents are being exported. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias You can't use this parameter in conjunction

			with the *Identity* parameter.
Name	Optional	System.String	The *Name* parameter specifies that export requests that have the specified name are returned. Use this parameter to search on the name that you provided when you created the export request. If you didn't specify a name when the request was created, the default name is MailboxExport*X* (where *X* = 0–9). You can't use this parameter in conjunction with the *Identity* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter specifies the organization you want configuration data from.
RequestQueue	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies that the cmdlet retrieves requests that are based in a queue on the specified database. By default, the

			request is created in a queue on the same database as where the mailbox is hosted.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Status	Optional	Microsoft.Exchange.Data.Directory.Recipient.RequestStatus	The *Status* parameter specifies that export requests with the specified status are returned. You can use the following values: • `AutoSuspended` • `Completed` • `CompletionInProgress` • `CompletedWithWarning` • `Failed` • `InProgress` • `None` • `Queued` • `Suspended` **Note:** `CompletionInProgress` and `AutoSuspended` don't apply to export requests and won't return any information. You can't use this parameter in conjunction with the *Identity*

			parameter.
Suspend	Optional	System.Boolean	The *Suspend* parameter specifies whether to return mailboxes with export requests that have been suspended. This parameter accepts `$true` or `$false`. You can't use this parameter in conjunction with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailboxExportRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-29*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-MailboxExportRequest** cmdlet to begin the process of exporting contents of a primary mailbox or archive to a .pst file.

Note:
This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't

assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MailboxExportRequest -FilePath <LongPath> -Mailbox
<MailboxOrMailUserIdParameter> [-AssociatedMessagesCopyOption <DoNotCopy |
MapByMessageClass | Copy>] [-ConflictResolutionOption <KeepSourceItem |
KeepLatestItem | KeepAll>] [-ContentFilter <String>] [-
ContentFilterLanguage <CultureInfo>] [-ExcludeDumpster <SwitchParameter>]
[-ExcludeFolders <String[]>] [-IncludeFolders <String[]>] [-IsArchive
<SwitchParameter>] [-SourceRootFolder <String>] [-TargetRootFolder
<String>] [-AcceptLargeDataLoss <SwitchParameter>] [-BadItemLimit
<Unlimited>] [-BatchName <String>] [-CompletedRequestAgeLimit <Unlimited>]
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-InternalFlags
<InternalMrsFlag[]>] [-LargeItemLimit <Unlimited>] [-Name <String>] [-
Priority <Lowest | Lower | Low | Normal | High | Higher | Highest |
Emergency>] [-SkipMerging <SkippableMergeComponent[]>] [-Suspend
<SwitchParameter>] [-SuspendComment <String>] [-WhatIf
[<SwitchParameter>]] [-WorkloadType <None | Local | Onboarding |
Offboarding | TenantUpgrade | LoadBalancing | Emergency |
RemotePstIngestion | SyncAggregation>]
```

Examples

EXAMPLE 1

This example exports the user Ayla Kol's primary mailbox to a .pst file on the network shared folder PSTFileShare on SERVER01.

```
New-MailboxExportRequest -Mailbox AylaKol -FilePath "\
\SERVER01\PSTFileShare\Ayla_Recovered.pst"
```

EXAMPLE 2

This example exports the user Kweku's archive to a .pst file on the network shared folder PSTFileShare on SERVER01.

```
New-MailboxExportRequest -Mailbox Kweku -FilePath "\
\SERVER01\PSTFileShare\Kweku_Archive.pst" -IsArchive
```

EXAMPLE 3

This example exports messages that contain the words "company" and "profit" in the body of the message for the user Tony received before January 1, 2012.

```
New-MailboxExportRequest -Mailbox Tony -ContentFilter
{(body -like "*company*") -and (body -like "*profit*") -and
(Received -lt "01/01/2012")} -FilePath "\\SERVER01
\PSTFileShare\Tony_CompanyProfits.pst"
```

EXAMPLE 4

This example exports all messages from Kweku's Inbox to the .pst file InPlaceHold.

```
New-MailboxExportRequest -Mailbox Kweku -IncludeFolders
"#Inbox#" -FilePath \\SERVER01\PSTFileShare\Kweku
\InPlaceHold.pst
```

Detailed Description

You can create more than one mailbox export request per mailbox, and each mailbox export request must have a unique name. Microsoft Exchange automatically generates up to 10 unique names for a mailbox export request. However, to create more than 10 export requests for a mailbox, you need to specify a unique name when creating the export request. You can remove existing export requests with the Remove-MailboxExportRequest cmdlet before starting a new request with the default request name <*alias*>\MailboxExport*X* (where *X* = 0–9).

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

You need to grant the following permission to the group Exchange Trusted Subsystem to the network share where you want to export or import PST files:

- To import PST files from the share: Read permission
- To save exported PST files to the share: Read/Write permission.

If you don't grant this permission, you will receive an error message stating that Exchange is unable to establish a connection to the PST file on the network share.

Parameters

Parameter	Required	Type	Description
FilePath	Required	Microsoft.Exchange.Data.LongPath	The *FilePath* parameter specifies the network share path of the .pst file to which data is exported, for example, \\SERVER01\PST Files\exported.pst. UNRESOLVED_TOKEN_VAL(GENL_ImportExport_Exc

			hangeTrustedSubsystemPerms)
Mailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxOrMailUserIdParameter	The *Mailbox* parameter specifies the mailbox or mail-enabled user from which to export contents. You can use the following values: • Alias • SMTP address • Display name
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51 or higher. Items are considered corrupted if the item can't be read from the source database or can't be written to the target database. Corrupted items won't be available in the destination mailbox or .pst file.
AssociatedMessagesCopyOption	Optional	Microsoft.Exchange.MailboxReplicationService.FAICopyOption	The *AssociatedMessagesCopyOption* parameter specifies whether associated messages are copied when the request

			is processed. Associated messages are special messages that contain hidden data with information about rules, views, and forms. By default, associated messages are copied. This parameter accepts the following values: • `DoNotCopy` The associated messages aren't copied. • `MapByMessageClass` This option finds the corresponding associated message by looking up the **MessageClass** attribute of the source message. If there's an associated message of this class in both source and target folders, it overwrites the associated message in the target. If there isn't an associated message in the target, it creates a copy in the target. • `Copy` This option copies associated messages from the source to the target. If the same message type exists both in the source and the target location, these associated messages are duplicated. This is the default option. **Note:**

			Content filtering doesn't apply to associated messages.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use 0 to not skip bad items. The valid input range for this parameter is from 0 through 2147483647. The default value is 0. We recommend that you keep the default value 0 and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the *BadItemLimit* parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process

			is complete.
BatchName	Optional	System.String	The *BatchName* parameter specifies a descriptive name for exporting a batch of mailboxes. You can use the name in the *BatchName* parameter as a string search when you use the **Get-MailboxExportRequest** cmdlet.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the request will be kept after it has completed before being automatically removed. The default *CompletedRequestAgeLimit* is 30 days.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

ConflictResolutionOption	Optional	Microsoft.Exchange.MailboxReplicationService.ConflictResolutionOption	The *ConflictResolutionOption* parameter specifies the action for the Microsoft Exchange Mailbox Replication service (MRS) to take if there are multiple matching messages in the target. This parameter takes the following values: • `KeepSourceItem` • `KeepLatestItem` • `KeepAll` The default value is `KeepSourceItem`.
ContentFilter	Optional	System.String	The *ContentFilter* parameter specifies message content to search for. Only contents that match the *ContentFilter* parameter will be exported into the .pst file.
ContentFilterLanguage	Optional	System.Globalization.CultureInfo	The *ContentFilterLanguage* parameter specifies the language being used in the *ContentFilter* parameter for string searches. The valid input for the *ContentFilterLanguage*

			parameter is the string names listed in the **Culture Name** column in the Microsoft .NET Class Library class reference available at CultureInfo Class.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExcludeDumpster	Optional	System.Management.Automation.SwitchParameter	The *ExcludeDumpster* parameter specifies whether to exclude the Recoverable Items folder. You don't have to include a value with this parameter. If you don't specify this parameter, the Recoverable Items folder is copied with the following subfolders: • Deletions • Versions • Purges
ExcludeFolders	Optional	System.String[]	The *ExcludeFolders* parameter specifies the list of folders to exclude

			during the export. Folder names aren't case-sensitive, and there are no character restrictions. Use the following syntax: `<FolderName>/*` Use this syntax to denote a personal folder under the folder specified in the *SourceRootFolder* parameter, for example, "MyProjects" or "MyProjects/FY2010". `#<FolderName>#/*` Use this syntax to denote a well-known folder regardless of the folder's name in another language. For example, `#Inbox#` denotes the Inbox folder even if the Inbox is localized in Turkish, which is Gelen Kutusu. Well-known folders include the following types: • Inbox • SentItems • DeletedItems • Calendar • Contacts • Drafts • Journal • Tasks • Notes • JunkEmail • CommunicationHistory • Voicemail

			• Fax • Conflicts • SyncIssues • LocalFailures • ServerFailures If the user creates a personal folder with the same name as a well-known folder and the # symbol surrounding it, you can use a back slash (\) as an escape character to specify that folder. For example, if a user creates a folder named #Notes# and you want to specify that folder, but not the well-known Notes folder, use the following syntax: `\#Notes\#`. **Note:** Wildcard characters can't be used in folder names.
IncludeFolders	Optional	System.String[]	The *IncludeFolders* parameter specifies the list of folders to include during the export. Folder names aren't case-sensitive, and there are no character restrictions. Use the following syntax: `<FolderName>/*` Use this syntax to denote a

			personal folder under the folder specified in the *SourceRootFolder* parameter, for example, "MyProjects" or "MyProjects/FY2010". `#<FolderName>#/*` Use this syntax to denote a well-known folder regardless of the folder's name in another language. For example, `#Inbox#` denotes the Inbox folder even if the Inbox is localized in Turkish, which is Gelen Kutusu. Well-known folders include the following types: • Inbox • SentItems • DeletedItems • Calendar • Contacts • Drafts • Journal • Tasks • Notes • JunkEmail • CommunicationHistory • Voicemail • Fax • Conflicts • SyncIssues • LocalFailures • ServerFailures If the user creates a personal folder with the

			same name as a well-known folder and the # symbol surrounding it, you can use a back slash (\) as an escape character to specify that folder. For example, if a user creates a folder named #Notes# and you want to specify that folder, but not the well-known Notes folder, use the following syntax: `\#Notes\#`. **Note:** Wildcard characters can't be used in folder names.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
IsArchive	Optional	System.Management.Automation.SwitchParameter	The *IsArchive* switch specifies that you're exporting from the user's archive.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of large items to skip if the request encounters such items in the mailbox. Use 0 to not

			skip any large items. If any number above 50 is specified then the *AcceptLargeDataLoss* parameter must also be specified. The default value is 0. We recommend that you use the default value of 0 and increase the *LargeItemLimit* only when large items are encountered.
Name	Optional	System.String	The *Name* parameter specifies the name of the specific request for tracking and display purposes. Because you can have multiple export requests per mailbox, Exchange precedes the name with the mailbox's alias. For example, if you create an export request for a user's mailbox that has the alias Kweku and specify the value of this parameter as PC1toArchive, the identity of this export request is Kweku\PC1toArchive. If you don't specify a name using this

			parameter, Exchange generates up to 10 request names per mailbox, which is MailboxExport*X* (where *X* = 0–9). The identity of the request is displayed and searchable as <*alias*>\MailboxExport*X*.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	The *Priority* parameter specifies the order in which this request should be processed in the request queue. Requests are processed in order, based on server health, status, priority, and last update time.
SkipMerging	Optional	Microsoft.Exchange.Management.RecipientTasks.SkippableMergeComponent[]	The *SkipMerging* parameter specifies steps in the export that should be skipped. This parameter is used primarily for debugging purposes.
SourceRootFolder	Optional	System.String	The *SourceRootFolder* parameter specifies the root folder of the mailbox from which data is exported. If this parameter isn't specified, the

			command exports all folders.
Suspend	Optional	System.Management.Automation.SwitchParameter	The *Suspend* switch specifies whether to suspend the request. If you use this switch, the request is queued, but the request won't reach the status of **InProgress** until you resume the request with the relevant resume cmdlet. You don't have to specify a value with this switch.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description about why the request was suspended. You can only use this parameter if you specify the *Suspend* parameter.
TargetRootFolder	Optional	System.String	The *TargetRootFolder* parameter specifies the top-level folder in which to export data. If you don't specify this parameter, the command exports folders to the top of the folder structure in the target .pst file. Content is merged under existing folders, and

			new folders are created if they don't already exist in the target folder structure.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WorkloadType	Optional	Microsoft.Exchange.MailboxReplicationService.RequestWorkloadType	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailboxExportRequest

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-MailboxExportRequest** cmdlet to remove fully or partially completed export requests. You can create multiple export requests for a specified mailbox provided that you specify a distinct name. Completed export requests aren't cleared automatically; they need to be removed by using this cmdlet.

Note:
This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

Note:
When a partially completed export request is removed, content already exported isn't removed from the PST file. If you want to start a new export request to the same file name and start with an empty PST file, you need to rename or delete the previous PST file.

```
Remove-MailboxExportRequest -Identity <MailboxExportRequestIdParameter>
<COMMON PARAMETERS>


Remove-MailboxExportRequest -RequestGuid <Guid> -RequestQueue
<DatabaseIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the second export request Ayla\MailboxExport1.

```
Remove-MailboxExportRequest -Identity "Ayla\MailboxExport1"
```

EXAMPLE 2

This example removes all export requests that have the status of Completed.

```
Get-MailboxExportRequest -Status Completed | Remove-
MailboxExportRequest
```

EXAMPLE 3

This example cancels the export request by using the *RequestGuid* parameter for a mailbox or archive on MBXDB01.

```
Remove-MailboxExportRequest -RequestQueue MBXDB01 -
RequestGuid 25e0eaf2-6cc2-4353-b83e-5cb7b72d441f
```

Detailed Description

The parameter set that requires the *Identity* parameter allows you to remove a fully or partially completed export request.

The parameter set that requires the *RequestGuid* and *RequestQueue* parameters is used for Microsoft Exchange Mailbox Replication service (MRS) debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxExportRequestIdParameter	The *Identity* parameter specifies the identity of the export request. By default, export requests are named <*alias*>\MailboxExport*X* (where *X* = 0–9). Use the following syntax: <*alias*>\<*name*>. You can't use this parameter in conjunction with the *RequestGuid* parameter.
RequestGuid	Required	System.Guid	The *RequestGuid* parameter specifies the unique identifier for the

			export request. To find the export request GUID, use the **Get-MailboxExportRequest** cmdlet. If you specify the *RequestGuid* parameter, you must also specify the *RequestQueue* parameter. You can't use this parameter in conjunction with the *Identity* parameter.
RequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies the mailbox database on which the request is being performed. This parameter accepts the following values: • GUID of the database • Database name
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Da	The *DomainController*

		ta.Fqdn	parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Resume-MailboxExportRequest

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Resume-MailboxExportRequest** cmdlet to resume an export request that was suspended or failed.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Resume-MailboxExportRequest -Identity <MailboxExportRequestIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resumes the export request kweku\export.

```
Resume-MailboxExportRequest -Identity kweku\export
```

EXAMPLE 2

This example resumes any failed export move requests.

```
Get-MailboxExportRequest -Status Failed | Resume-
MailboxExportRequest
```

Detailed Description

The **Resume-MailboxExportRequest** cmdlet can be pipelined with the **Get-MailboxExportRequest** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxExportRequestIdParameter	The *Identity* parameter specifies the identity of the export request. By default, export requests are named <*alias*>\MailboxExport*X* (where *X* = 0–9). Use the following syntax: <*alias*>\<*name*>.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxExportRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-MailboxExportRequest** cmdlet to change export request options after the request has been created. You can use the **Set-MailboxExportRequest** cmdlet to recover from failed export requests.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxExportRequest [-AcceptLargeDataLoss <SwitchParameter>] [-
BadItemLimit <Unlimited>] [-BatchName <String>] [-CompletedRequestAgeLimit
<Unlimited>] [-InternalFlags <InternalMrsFlag[]>] [-LargeItemLimit
<Unlimited>] [-Priority <Lowest | Lower | Low | Normal | High | Higher |
Highest | Emergency>] [-SkipMerging <SkippableMergeComponent[]>] <COMMON
PARAMETERS>


Set-MailboxExportRequest -RehomeRequest <SwitchParameter> <COMMON
PARAMETERS>


COMMON PARAMETERS: -Identity <MailboxExportRequestIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the second export request Ayla\MailboxExport1 to accept up to 10 corrupt mailbox items.

```
Set-MailboxExportRequest -Identity "Ayla\MailboxExport1\" -
BadItemLimit 10
```

Detailed Description

You can pipeline the **Set-MailboxExportRequest** cmdlet from the Get-MailboxExportRequest cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxExportRequestIdParameter	The *Identity* parameter specifies the identity of the export request. By default, export requests are named <*alias*>\MailboxExport*X* (where

			X = 0–9). If you specify a name for the export request, use the following syntax: *<alias>**<name>*.
RehomeRequest	Required	System.Management.Automation.SwitchParameter	The *RehomeRequest* parameter specifies to the Microsoft Exchange Mailbox Replication service (MRS) that the request needs to be moved to the same database as the mailbox that's being exported. This parameter is used primarily for debugging purposes.
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51 or higher. Items are considered corrupted if the item can't be read from the source database or can't be written to the target database. Corrupted items won't be available in the destination

			mailbox or .pst file.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use 0 to not skip bad items. The valid input range for this parameter is from 0 through 2147483647. The default value is 0. We recommend that you keep the default value 0 and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the *BadItemLimit* parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and

			any corrupted items aren't available after the process is complete.
BatchName	Optional	System.String	The *BatchName* parameter specifies the name of the batch.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the request will be kept after it has completed before being automatically removed. The default value is 30 days.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of large items to skip if the request encounters such items in the mailbox. Use 0 to not skip any large items. If any number above 50 is specified, the *AcceptLargeDataLoss* parameter must also be specified. The default value is 0. We recommend that you use the default value of 0 and increase the *LargeItemLimit* only when large items are encountered.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	The *Priority* parameter specifies the order in which this request is processed in the request queue. Requests are

			processed in order, based on server health, status, priority, and last update time.
SkipMerging	Optional	Microsoft.Exchange.Management.RecipientTasks.SkippableMergeComponent[]	The *SkipMerging* parameter specifies steps in the export that should be skipped. This parameter is used primarily for debugging purposes.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Suspend-MailboxExportRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Suspend-MailboxExportRequest** cmdlet to suspend an export request any time after the request was created, but before the request reaches the status of `Completed`. You can resume the request by using the Resume-MailboxExportRequest cmdlet.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Suspend-MailboxExportRequest -Identity <MailboxExportRequestIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-SuspendComment
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example suspends the second export request for Ayla's mailbox with the identity Ayla \MailboxExport1.

```
Suspend-MailboxExportRequest -Identity "Ayla
\MailboxExport1"
```

EXAMPLE 2

This example suspends all export requests that are in progress by using the Get-MailboxExportRequest cmdlet to retrieve all requests with a status of `InProgress`, and then pipelining the output to the **Suspend-MailboxExportRequest** cmdlet with the suspend comment "Resume after 22:00 (10 P.M.)".

```
Get-MailboxExportRequest -Status InProgress | Suspend-
```

```
MailboxExportRequest -SuspendComment "Resume after 22:00
(10 P.M.)"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxExportRequestIdParameter	The *Identity* parameter specifies the identity of the export request. By default, export requests are named <*alias*>\MailboxExport*X* (where *X* = 0–9). Use the following syntax: <*alias*>\<*name*>.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Dat	The *DomainController*

		a.Fqdn	parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description about why the request was suspended. You can only use this parameter if you specify the *Suspend* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxExportRequestStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxExportRequestStatistics** cmdlet to view detailed information about export requests.

Note:
This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxExportRequestStatistics -Identity
<MailboxExportRequestIdParameter> <COMMON PARAMETERS>


Get-MailboxExportRequestStatistics -RequestQueue <DatabaseIdParameter> [-
RequestGuid <Guid>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Diagnostic <SwitchParameter>] [-DomainController
<Fqdn>] [-IncludeReport <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns the default statistics for the second export request for Tony Smith. The type of information returned by default includes name, mailbox, and status.

```
Get-MailboxExportRequestStatistics -Identity Tony
\MailboxExport1
```

EXAMPLE 2

This example returns statistics for Tony Smith's mailbox and exports the report to a .csv file.

```
Get-MailboxExportRequestStatistics -Identity Tony
\MailboxExport | Export-CSV \\SERVER01
\ExportRequest_Reports\Tony_Exportstats.csv
```

EXAMPLE 3

This example returns additional information about the export request for Tony Smith's mailbox by using the *IncludeReport* parameter and by pipelining the results to the **Format-List** command. (The export request was created using the **New-MailboxExportRequest**.)

```
Get-MailboxExportRequestStatistics -Identity Tony\LegalHold
-IncludeReport | Format-List
```

EXAMPLE 4

This example returns default statistics for an export request being processed by the instance of MRS running on the server CAS01. This command only returns information for export requests currently being processed by an instance of MRS. If the request is already finished, it won't be returned.

```
Get-MailboxExportRequestStatistics -RequestQueue
MailboxDatabase01
```

EXAMPLE 5

This example returns additional information for all the export requests that have a status of `Failed` by using the *IncludeReport* parameter, and then saves the information to the text file AllExportReports.txt.

```
Get-MailboxExportRequest -Status Failed | Get-
MailboxExportRequestStatistics -IncludeReport | Format-List
> AllExportReports.txt
```

Detailed Description

You can pipeline the **Get-MailboxExportRequestStatistics** cmdlet from the Get-MailboxExportRequest cmdlet.

The *RequestQueue* parameter syntax set is for debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxExportRequestIdParameter	The *Identity* parameter specifies the identity of the export request. By default, export requests are named <*alias*>\MailboxExport*X* (where *X* = 0–9). If you specified a name for the export request when it was created by using the New-MailboxExportRequest cmdlet, use the following syntax: <*alias*>\<*name*>. This parameter can't be used with the *RequestGuid*, or *RequestQueue* parameters.
RequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies the mailbox database on which the mailbox or archive of the request resides. You can use one of the following values: • GUID of the database • Database name This parameter can't be used in conjunction with

			the *Identity* parameter.
Diagnostic	Optional	System.Management.Automation.SwitchParameter	The *Diagnostic* switch specifies whether to retrieve extremely detailed information about the mailbox export request.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
IncludeReport	Optional	System.Management.Automation.SwitchParameter	The *IncludeReport* switch specifies whether to return additional details, which can be used for troubleshooting.
RequestGuid	Optional	System.Guid	The *RequestGuid* parameter specifies the unique identifier for the export request. To find the export request GUID, use the Get-MailboxExportRequest cmdlet. If you specify the *RequestGuid* parameter, you must also specify the *RequestQueue* parameter. You can't use this

			parameter in conjunction with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxFolder

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxFolder** cmdlet to retrieve folders for a specified mailbox when the mailbox owner runs the command.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxFolder [-Identity <MailboxFolderIdParameter>] <COMMON
PARAMETERS>


Get-MailboxFolder -GetChildren <SwitchParameter> [-Identity
<MailboxFolderIdParameter>] [-MailFolderOnly <SwitchParameter>] [-
ResultSize <Unlimited>] <COMMON PARAMETERS>


Get-MailboxFolder -Recurse <SwitchParameter> [-Identity
<MailboxFolderIdParameter>] [-MailFolderOnly <SwitchParameter>] [-
ResultSize <Unlimited>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves the Inbox folder in Tony's mailbox. The *Identity* parameter is supplied in the format of <*Mailbox Identity*>:<*Folder*>.

```
Get-MailboxFolder -Identity Tony:\Inbox
```

EXAMPLE 2

This example returns the root folders in Tony's mailbox. The *Identity* parameter is supplied in the format of <*Mailbox Identity*>.

```
Get-MailboxFolder -Identity Tony
```

EXAMPLE 3

This example returns the first level of mail folders in Tony's mailbox.

```
Get-MailboxFolder -Identity Tony -GetChildren -
MailFolderOnly
```

EXAMPLE 4

This example returns information about all the subfolders under Inbox in Tony's mailbox.

```
Get-MailboxFolder -Identity Tony:\Inbox -GetChildren
```

EXAMPLE 5

This example returns all levels of folders under Inbox in Tony's mailbox.

```
Get-MailboxFolder -Identity Tony:\Inbox -Recurse
```

Detailed Description

If the mailbox isn't specified, the cmdlet returns the specified folders in the mailbox of the mailbox owner currently running the command. This command checks that the mailbox specified in the *Identity* parameter is a valid Exchange mailbox before retrieving the requested folders. The cmdlet returns all folders if the *MailFolderOnly* parameter isn't specified.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox folders" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
GetChildren	Required	System.Management.Automation.SwitchParameter	The *GetChildren* parameter specifies whether to return only the first level of subfolders under the specified parent folder. You can't specify both this parameter and the *Recurse* parameter at the same time. The default value is `$false`.
Recurse	Required	System.Management.Automation.SwitchParameter	The *Recurse* parameter specifies whether to return the specified parent folder and all of its subfolders. You can't specify both this parameter and the *GetChildren* parameter at the same time. The default value is `$false`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active

			Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *Identity* parameter specifies the mailbox and the folder that the command returns information about. If the folder isn't specified, the command returns information about folders in the root hierarchy of the specified mailbox. You can specify values in the following format: *<Mailbox Identity>:<Parent>* Valid values for *<Mailbox Identity>* include: • GUID • ADObjectID • Distinguished name (DN) • *Domain\username* • User principal name (UPN) • LegacyExchangeDN • SMTP address • Alias • Store object ID of the parent folder Values for *<Parent>* can be both the store object ID and a path string such as

			\Inbox\Personal.
MailFolderOnly	Optional	System.Management.Automation.SwitchParameter	The *MailFolderOnly* parameter specifies whether to return only the mail folders in the specified mailbox. The default value is `$false`.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of settings to return. If you want to return all settings that match the command, use `unlimited` for the value of this parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailboxFolder

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **New-MailboxFolder** cmdlet to create a folder in a mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MailboxFolder -Name <String> -Parent <MailboxFolderIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the folder Personal under the Inbox folder of Tony's mailbox.

```
New-MailboxFolder -Parent Tony:\Inbox -Name Personal
```

EXAMPLE 2

This example creates the folder Personal in the root folder hierarchy of Tony's mailbox.

```
New-MailboxFolder -Parent Tony -Name Personal
```

EXAMPLE 3

This example creates the folder Personal under the Inbox folder in the mailbox for Tony who's running the command.

```
New-MailboxFolder -Parent :\Inbox -Name Personal
```

Detailed Description

If no parent folder is specified, the cmdlet creates a mail folder in the root folder hierarchy of the mailbox. If the mailbox isn't specified, the cmdlet creates the folder in the mailbox of the user currently running the task. When run, the cmdlet returns the new folder name and the folder path as the output.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox folders" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description

Name	Required	System.String	The *Name* parameter specifies the name of the new folder.
Parent	Required	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *Parent* parameter specifies values of the mailbox identity and the parent folder under which the new folder is to be created. If the parent folder isn't specified, the cmdlet creates the folder in the root folder hierarchy of the specified mailbox. You can specify the values in this format: <*Mailbox Identity*>:<*Parent*> Valid values for <*Mailbox Identity*> include: • GUID • ADObjectID • Distinguished name (DN) • *Domain**username* • User principal name (UPN) • LegacyExchangeDN • SMTP address • Alias • Store object ID of the parent folder Values for <*Parent*> can

			be both the store object ID and a path string such as "\Inbox\Personal".
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-MailboxFolderPermission

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-19*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Add-MailboxFolderPermission** cmdlet to manage folder-level permissions for all folders within a user's mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-MailboxFolderPermission -Identity <MailboxFolderIdParameter> -
AccessRights <MailboxFolderAccessRight[]> -User
<MailboxFolderUserIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example assigns permissions for Ed to access Ayla's Marketing mailbox folder and applies the Owner role to his access of that folder.

```
Add-MailboxFolderPermission -Identity ayla@contoso.com:
\Marketing -User Ed@contoso.com -AccessRights Owner
```

EXAMPLE 2

This example assigns permissions for a user to access specific folders in another user's mailbox in an Exchange Online or Office 365 environment. Do the following:

1. Connect to Exchange Online by using remote PowerShell. For info about how to do this, go to the following Microsoft website:**Connect to Exchange Online using remote PowerShell**.
2. Enter a command using the following syntax to assign access permissions to the specific folder:

```
Add-MailboxFolderPermission -Identity <SMTP address or
alias of recipient>:<Folder path> -AccessRights <Permission
you want to grant the recipient> -User < SMTP address or
alias of recipient to be granted access>
```

This example assigns permissions for Ed to access Ayla's Marketing mailbox folder and applies the Owner role to his access of that folder.

```
Add-MailboxFolderPermission -Identity ayla@contoso.com:
\Marketing -User Ed@contoso.com -AccessRights Owner
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox folder permissions" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccessRights	Required	Microsoft.Exchange.Management.StoreTasks.MailboxFolderAccessRight[]	The *AccessRights* parameter specifies the permissions for the user with the following access rights: • **ReadItems** The user has the right to read items within the

			specified folder. • **CreateItems** The user has the right to create items within the specified folder. • **EditOwnedItems** The user has the right to edit the items that the user owns in the specified folder. • **DeleteOwnedItems** The user has the right to delete items that the user owns in the specified folder. • **EditAllItems** The user has the right to edit all items in the specified folder. • **DeleteAllItems** The user has the right to delete all items in the specified folder. • **CreateSubfolders** The user has the right to create subfolders in the specified folder. • **FolderOwner** The user is the owner of the specified folder. The user has the right to view and move the folder and create subfolders. The user

			can't read items, edit items, delete items, or create items. • **FolderContact** The user is the contact for the specified public folder. • **FolderVisible** The user can view the specified folder, but can't read or edit items within the specified public folder. The *AccessRights* parameter also specifies the permissions for the user with the following roles, which are a combination of the rights listed previously: • **None** FolderVisible • **Owner** CreateItems, ReadItems, CreateSubfolders, FolderOwner, FolderContact, FolderVisible, EditOwnedItems, EditAllItems, DeleteOwnedItems, DeleteAllItems • **PublishingEditor** CreateItems,

			ReadItems, CreateSubfolders, FolderVisible, EditOwnedItems, EditAllItems, DeleteOwnedItems, DeleteAllItems • **Editor** CreateItems, ReadItems, FolderVisible, EditOwnedItems, EditAllItems, DeleteOwnedItems, DeleteAllItems • **PublishingAuthor** CreateItems, ReadItems, CreateSubfolders, FolderVisible, EditOwnedItems, DeleteOwnedItems • **Author** CreateItems, ReadItems, FolderVisible, EditOwnedItems, DeleteOwnedItems • **NonEditingAuthor** CreateItems, ReadItems, FolderVisible • **Reviewer** ReadItems, FolderVisible • **Contributor** CreateItems, FolderVisible

			The following roles apply specifically to calendar folders: • **AvailabilityOnly** View only availability data • **LimitedDetails** View availability data with subject and location
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *Identity* parameter specifies the recipient and folder that you want to change the permissions for. This parameter takes the following format: *<SMTP Address or Alias of Recipient>*:*<Folder path>*. The following is an example: john@contoso.com:\Calendar
User	Required	Microsoft.Exchange.Management.StoreTasks.MailboxFolderUserIdParameter	The *User* parameter specifies who's granted permission to view or modify the folder contents of the user specified in the *Identity* parameter. This parameter accepts only users and distribution lists that have SMTP addresses. Security Groups are not allowed. The following values are

			acceptable: • Alias • SMTP address
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxFolderPermission

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxFolderPermission** cmdlet to view the folder-level permissions for a folder or a specific user's permissions for a folder.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxFolderPermission -Identity <MailboxFolderIdParameter> [-
DomainController <Fqdn>] [-User <MailboxFolderUserIdParameter>]
```

Examples

EXAMPLE 1

This example returns the current list of user permissions for John's Reports mailbox folder under the Marketing folder.

```
Get-MailboxFolderPermission -Identity john@contoso.com:
\Marketing\Reports
```

EXAMPLE 2

This example returns the permissions that Ayla has to view John's Marketing Reports folder.

```
Get-MailboxFolderPermission -Identity john@contoso.com:
\Marketing\Reports -User Ayla@contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox folder permissions" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *Identity* parameter specifies the mailbox and folder for which you want to view permissions. This parameter takes the following format: <*SMTP Address or Alias of the mailbox*>:<*Folder path*>, for example, john@contoso.com:\Calendar.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
User	Optional	Microsoft.Exchange.Management.StoreTasks.MailboxFolderUserIdParameter	The *User* parameter specifies who's granted permission to view or modify folder contents of the user and folder specified in the *Identity* parameter. You can use the following values: • Alias • SMTP address

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailboxFolderPermission

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MailboxFolderPermission** cmdlet to remove folder-level permissions for a user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MailboxFolderPermission -Identity <MailboxFolderIdParameter> -User
<MailboxFolderUserIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes John's permission to modify Kim's mailbox folder Training.

```
Remove-MailboxFolderPermission -Identity kim@contoso.com:
\Training -User john@contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox folder permissions" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *Identity* parameter specifies the user and folder for whom you want to modify permissions. This parameter takes the following format: *<SMTP Address or Alias of the recipient>:<Folder path>*, for example, john@contoso.com:\Calendar.
User	Required	Microsoft.Exchange.M	The *User* parameter specifies who's granted

		anagement.StoreTasks.MailboxFolderUserIdParameter	permissions to view or modify folder contents of the user specified in the *Identity* parameter. The following values are acceptable: • Alias • SMTP address
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxFolderPermission

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailboxFolderPermission** cmdlet to update folder-level permissions for all folders within a user's mailbox. The cmdlet differs from the Add-MailboxFolderPermission cmdlet in that it edits an existing permission entry.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxFolderPermission -Identity <MailboxFolderIdParameter> -
AccessRights <MailboxFolderAccessRight[]> -User
<MailboxFolderUserIdParameter> [-Confirm [<SwitchParameter>]] [-
```

```
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example assigns permissions for Ed to access Ayla's Marketing mailbox folder and applies the Owner role to his access of that folder.

```
Set-MailboxFolderPermission -Identity ayla@contoso.com:
\Marketing -User Ed@contoso.com -AccessRights Owner
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox folders" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccessRights	Required	Microsoft.Exchange.Management.StoreTasks.MailboxFolderAccessRight[]	The *AccessRights* parameter specifies the permissions for the user with the following access rights: • **ReadItems** The user has the right to read items within the specified folder. • **CreateItems** The user has the right to create items within the specified folder. • **EditOwnedItems** The user has the right to edit the items that the user

			owns in the specified folder. • **DeleteOwnedItems** The user has the right to delete items that the user owns in the specified folder. • **EditAllItems** The user has the right to edit all items in the specified folder. • **DeleteAllItems** The user has the right to delete all items in the specified folder. • **CreateSubfolders** The user has the right to create subfolders in the specified folder. • **FolderOwner** The user is the owner of the specified folder. The user has the right to view and move the folder and create subfolders. The user can't read items, edit items, delete items, or create items. • **FolderContact** The user is the contact for the specified folder. • **FolderVisible** The user can view the

			specified folder, but can't read or edit items within the specified folder. The *AccessRights* parameter also specifies the permissions for the user with the following roles, which are a combination of the rights listed previously: • **None** FolderVisible • **Owner** CreateItems, ReadItems, CreateSubfolders, FolderOwner, FolderContact, FolderVisible, EditOwnedItems, EditAllItems, DeleteOwnedItems, DeleteAllItems • **PublishingEditor** CreateItems, ReadItems, CreateSubfolders, FolderVisible, EditOwnedItems, EditAllItems, DeleteOwnedItems, DeleteAllItems • **Editor** CreateItems, ReadItems,

			FolderVisible, EditOwnedItems, EditAllItems, DeleteOwnedItems, DeleteAllItems • **PublishingAuthor** CreateItems, ReadItems, CreateSubfolders, FolderVisible, EditOwnedItems, DeleteOwnedItems • **Author** CreateItems, ReadItems, FolderVisible, EditOwnedItems, DeleteOwnedItems • **NonEditingAuthor** CreateItems, ReadItems, FolderVisible • **Reviewer** ReadItems, FolderVisible • **Contributor** CreateItems, FolderVisible The following roles apply specifically to calendar folders: • **AvailabilityOnly** View only availability data • **LimitedDetails** View availability data with subject and location

Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxFolderIdParameter	The *Identity* parameter specifies the recipient and folder that you want to change the permissions for. This parameter takes the following format: *<SMTP Address or Alias of Recipient>*:*<Folder path>*. The following is an example: john@contoso.com:\Calendar
User	Required	Microsoft.Exchange.Management.StoreTasks.MailboxFolderUserIdParameter	The *User* parameter specifies who's granted permission to view or modify the folder contents of the user specified in the *Identity* parameter. The following values are acceptable: • Alias • SMTP address
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxFolderStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxFolderStatistics** cmdlet to retrieve information about the folders in a specified mailbox, including the number and size of items in the folder, the folder name and ID, and other information.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxFolderStatistics -Identity <MailboxOrMailUserIdParameter> [-
Archive <SwitchParameter>] [-DomainController <Fqdn>] [-FolderScope
<Calendar | Contacts | DeletedItems | Drafts | Inbox | JunkEmail | Journal
| Notes | Outbox | SentItems | Tasks | All | ManagedCustomFolder |
RssSubscriptions | SyncIssues | ConversationHistory | Personal |
RecoverableItems | NonIpmRoot | LegacyArchiveJournals>] [-IncludeAnalysis
<SwitchParameter>] [-IncludeOldestAndNewestItems <SwitchParameter>]
```

Examples

EXAMPLE 1

This example doesn't specify the *FolderScope* parameter and retrieves all the information about the user Chris in the Contoso domain.

```
Get-MailboxFolderStatistics -Identity contoso\chris
```

EXAMPLE 2

This example uses the *FolderScope* parameter to view the statistics for calendar folders for the user Chris.

```
Get-MailboxFolderStatistics -Identity Chris -FolderScope
Calendar
```

EXAMPLE 3

This example uses the *Archive* parameter to view the statistics for Ayla's archive.

```
Get-MailboxFolderStatistics -Identity Ayla@contoso.com -
Archive
```

EXAMPLE 4

This example uses the *IncludeAnalysis* parameter to view the statistics of Tony's Recoverable Items folder.

```
Get-MailboxFolderStatistics -Identity "Tony" -FolderScope RecoverableItems -IncludeAnalysis
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox folder permissions" entry in the Recipients Permissions topic.

Note:

A mailbox can have hidden items that are never visible to the user and are only used by applications. The **Get-MailboxFolderStatistics** cmdlet can return hidden items for the following values: FolderSize, FolderAndSubfolderSize, ItemsInFolder, and ItemsInFolderAndSubfolders.

Note:

The **Get-MailboxFolderStatistics** cmdlet shouldn't be confused with the **Get-MailboxStatistics** cmdlet. For more information, see Get-MailboxStatistics.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxOrMailUserIdParameter	The *Identity* parameter specifies the identity of the mailbox or mail-enabled user. You can use one of the following values: • GUID • ADObjectID • Distinguished name (DN) • *Domain\Account* • User principal name

			(UPN) • LegacyExchangeDN • SMTP address • Alias
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* parameter specifies whether to return the usage statistics of the archive associated with the mailbox or mail-enabled user. You don't need to provide a value with this parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
FolderScope	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ElcFolderType	This parameter is available only in on-premises Exchange 2013. The *FolderScope* parameter specifies the scope of the search by folder type. Valid parameter values include: • `All` • `Calendar` • `Contacts`

			• ConversationHistory • DeletedItems • Drafts • Inbox • JunkEmail • Journal • LegacyArchiveJournals • ManagedCustomFolder • NonIpmRoot • Notes • Outbox • Personal • RecoverableItems • RssSubscriptions • SentItems • SyncIssues • Tasks If the ManagedCustomFolder value is entered, the command returns the output for all managed custom folders. If the RecoverableItems value is entered, the command returns the output for the Recoverable Items folder and the Deletions, Purges, and Versions subfolders.
IncludeAnalysis	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IncludeAnalysis* parameter specifies whether to scan all items within a folder and return statistics related to the folder and item size. This parameter should be used for troubleshooting

			purposes, and it may take a long time to complete. You don't need to provide a value with this parameter.
IncludeOldestAndNewestItems	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IncludeOldestAndNewestItems* parameter specifies whether to return the dates of the oldest and newest items in each folder.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxImportRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxImportRequest** cmdlet to view the detailed status of an ongoing import request that was initiated using the **New-MailboxImportRequest** cmdlet.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxImportRequest [-AccountPartition <AccountPartitionIdParameter>]
[-Identity <MailboxImportRequestIdParameter>] [-Organization
<OrganizationIdParameter>] <COMMON PARAMETERS>


Get-MailboxImportRequest [-AccountPartition <AccountPartitionIdParameter>]
[-BatchName <String>] [-HighPriority <$true | $false>] [-Mailbox
<MailboxOrMailUserIdParameter>] [-Name <String>] [-Organization
<OrganizationIdParameter>] [-RequestQueue <DatabaseIdParameter>] [-Status
<None | Queued | InProgress | AutoSuspended | CompletionInProgress |
Synced | Completed | CompletedWithWarning | Suspended | Failed>] [-Suspend
<$true | $false>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns the default information regarding the status of the ongoing import request with the identity tony\Recovered. The type of information returned by default includes name, mailbox, and status.

```
Get-MailboxImportRequest -Identity "tony\Recovered"
```

EXAMPLE 2

This example returns the status of import requests in the ImportingDB1PSTs batch that completed.

```
Get-MailboxImportRequest -BatchName "ImportingDB1PSTs" -
Status Completed
```

EXAMPLE 3

This example returns all import requests that have the name Recovered where the import has been suspended.

```
Get-MailboxImportRequest -Name "Recovered" -Suspend $true
```

Detailed Description

The search criteria for the **Get-MailboxImportRequest** cmdlet is a Boolean **And** statement. If you use multiple parameters, you narrow your search and reduce your search results.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
BatchName	Optional	System.String	The *BatchName* parameter specifies the name given to a batch import request. You can't use this parameter in conjunction with the *Identity* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active

			Directory.
HighPriority	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *HighPriority* parameter specifies that the cmdlet should return requests that were created with the *HighPriority* flag. The *HighPriority* flag indicates that the request should be processed before other lower priority requests in the queue. You can't use this parameter in conjunction with the *Identity* parameter.
Identity	Optional	Microsoft.Exchange.MailboxReplicationService.MailboxImportRequestIdParameter	The *Identity* parameter specifies the identity of the import request. By default, import requests are named <*alias*>\MailboxImport*X* (where *X* = 0–9). If you specify a name for the import request, use the following syntax: <*alias*>\<*name*>. Microsoft Exchange Server 2013 automatically precedes the request with

			the mailbox's alias. This parameter can't be used in conjunction with the following parameters: • *BatchName* • *Mailbox* • *Name* • *Status* • *Suspend* • *HighPriority*
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxOrMailUserIdParameter	The *Mailbox* parameter specifies the identity of the mailbox or mail user into which content is being imported. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias You can't use this parameter in conjunction with the *Identity* parameter.
Name	Optional	System.String	The *Name* parameter specifies that import requests that have the

			specified name are returned. Use this parameter to search on the name that you provided when you created the import request. If you didn't specify a name when the request was created, the default name is MailboxImport*X* (where *X* = 0-9). You can't use this parameter in conjunction with the *Identity* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter specifies the organization you want configuration data from.
RequestQueue	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *RequestQueue* parameter specifies that the cmdlet retrieves requests that are based in a queue on the specified database. By default, the request is created in a queue on the same

			database as where the mailbox is hosted.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Status	Optional	Microsoft.Exchange.Data.Directory.Recipient.RequestStatus	The *Status* parameter specifies that import requests with the specified status are returned. You can use the following values: • `AutoSuspended` • `Completed` • `CompletionInProgress` • `CompletedWithWarning` • `Failed` • `InProgress` • `None` • `Queued` • `Suspended` **Note:** `CompletionInProgress` and `AutoSuspended` don't apply to import requests and won't return any information. You can't use this parameter in conjunction with the *Identity* parameter.

Suspend	Optional	System.Boolean	The *Suspend* parameter specifies whether to return mailboxes with import requests that have been suspended. This parameter accepts `$true` or `$false`. You can't use this parameter in conjunction with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailboxImportRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-28*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-MailboxImportRequest** cmdlet to begin the process of importing a .pst file to a mailbox or archive. You can create more than one mailbox import request per mailbox and each mailbox import request must have a unique name. Microsoft Exchange automatically generates up to 10 unique names for a mailbox import request. However, to create more than 10 import requests for a mailbox, you need to specify a unique name when creating the import request, or you can remove existing import requests with the Remove-MailboxExportRequest cmdlet before starting a

new import request with the default request *<Alias>*\MailboxImport*X* (where *X* = 0–9).

By default, the import checks for duplication of items and doesn't copy the data from the .pst file into the mailbox or archive if a matching item exists in the target mailbox or target archive.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MailboxImportRequest -FilePath <LongPath> -Mailbox
<MailboxOrMailUserIdParameter> [-AssociatedMessagesCopyOption <DoNotCopy |
MapByMessageClass | Copy>] [-ConflictResolutionOption <KeepSourceItem |
KeepLatestItem | KeepAll>] [-ContentCodePage <Int32>] [-ExcludeDumpster
<SwitchParameter>] [-ExcludeFolders <String[]>] [-IncludeFolders <String[]
>] [-IsArchive <SwitchParameter>] [-RemoteCredential <PSCredential>] [-
RemoteHostName <Fqdn>] [-SourceRootFolder <String>] [-TargetRootFolder
<String>] [-AcceptLargeDataLoss <SwitchParameter>] [-BadItemLimit
<Unlimited>] [-BatchName <String>] [-CompletedRequestAgeLimit <Unlimited>]
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-InternalFlags
<InternalMrsFlag[]>] [-LargeItemLimit <Unlimited>] [-Name <String>] [-
Priority <Lowest | Lower | Low | Normal | High | Higher | Highest |
Emergency>] [-SkipMerging <SkippableMergeComponent[]>] [-Suspend
<SwitchParameter>] [-SuspendComment <String>] [-WhatIf
[<SwitchParameter>]] [-WorkloadType <None | Local | Onboarding |
Offboarding | TenantUpgrade | LoadBalancing | Emergency |
RemotePstIngestion | SyncAggregation>]
```

Examples

EXAMPLE 1

This example imports a recovered .pst file on SERVER01 into the user Ayla's primary mailbox. Only data in the .pst file's Inbox is imported. The data is imported into the RecoveredFiles folder of the target mailbox for Ayla.

```
New-MailboxImportRequest -Mailbox Ayla -FilePath \\SERVER01
\PSTFiles\Recovered.pst -TargetRootFolder "RecoveredFiles"
-IncludeFolders "#Inbox#"
```

EXAMPLE 2

This example imports a .pst file into Kweku's archive folder. The *TargetRootFolder* isn't specified; therefore, content is merged under existing folders and new folders are created if they don't already exist in the target folder structure.

```
New-MailboxImportRequest -Mailbox Kweku -IsArchive -
FilePath \\SERVER01\PSTFiles\Archives\Kweku\Archive2012.pst
```

EXAMPLE 3

This example imports all of the .pst files on a shared folder. Each .pst file name is named after a corresponding user's alias. The command creates an import request for all the .pst files and imports the data into the matching mailbox.

```
Dir \\SERVER01\PSTshareRO\Recovered\*.pst | %{ New-
MailboxImportRequest -Name RecoveredPST -BatchName
Recovered -Mailbox $_.BaseName -FilePath $_.FullName -
TargetRootFolder SubFolderInPrimary}
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

You need to grant the following permission to the group Exchange Trusted Subsystem to the network share where you want to export or import PST files:

- To import PST files from the share: Read permission
- To save exported PST files to the share: Read/Write permission.

If you don't grant this permission, you will receive an error message stating that Exchange is unable to establish a connection to the PST file on the network share.

Parameters

Parameter	Required	Type	Description
FilePath	Required	Microsoft.Exchange.Data.LongPath	The *FilePath* parameter specifies the network share path of the .pst file from which data is imported, for example, \\SERVER01\PST Files\ToImport.pst. You need to grant the following permission to the group Exchange Trusted Subsystem to the network share where you

			want to export or import PST files: • To import PST files from the share: Read permission • To save exported PST files to the share: Read/Write permission. If you don't grant this permission, you will receive an error message stating that Exchange is unable to establish a connection to the PST file on the network share.
Mailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxOrMailUserIdParameter	The *Mailbox* parameter specifies the mailbox or mail-enabled user into which to import contents. You can use the following values: • Alias • SMTP address • Display name
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51 or higher. Items are considered corrupted if the item can't be read

			from the source database or can't be written to the target database. Corrupted items won't be available in the destination mailbox or .pst file.
AssociatedMessagesCopyOption	Optional	Microsoft.Exchange.MailboxReplicationService.FAICopyOption	The *AssociatedMessagesCopyOption* parameter specifies whether associated messages are copied when the request is processed. Associated messages are special messages that contain hidden data with information about rules, views, and forms. By default, associated messages are copied. This parameter accepts the following values: • `DoNotCopy` The associated messages aren't copied. • `MapByMessageClass` This option finds the corresponding associated message by looking up the **MessageClass** attribute of the source message. If there's an associated message of this class in both source and target

			folders, it overwrites the associated message in the target. If there isn't an associated message in the target, it creates a copy in the target. • `Copy` This option copies associated messages from the source to the target. If the same message type exists both in the source and the target location, these associated messages are duplicated. This is the default option. **Note:** Content filtering doesn't apply to associated messages.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use `0` to not skip bad items. The valid input range for this parameter is from `0` through `2147483647`. The default value is `0`. We recommend that you keep the default value `0` and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the

			BadItemLimit parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process is complete.
BatchName	Optional	System.String	The *BatchName* parameter specifies a descriptive name for importing a batch of mailboxes. You can use the name in the *BatchName* parameter as a string search when you use the **Get-MailboxImportRequest** cmdlet.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the request will be kept after it has completed before being automatically removed. The default value of the

			CompletedRequestAgeLimit parameter is 30 days.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ConflictResolutionOption	Optional	Microsoft.Exchange.MailboxReplicationService.ConflictResolutionOption	The *ConflictResolutionOption* parameter specifies the action for the Microsoft Exchange Mailbox Replication service (MRS) to take if there are multiple matching messages in the target. This parameter takes the following values: • `KeepSourceItem` • `KeepLatestItem` • `KeepAll` The default value is `KeepSourceItem`.
ContentCodePage	Optional	System.Int32	The *ContentCodePage* parameter specifies the specific code page to use for an ANSI pst file. The

			ANSI pst file is the Outlook 97 to Outlook 2002 pst format files. You can find the valid values in the Code Page Identifiers topic.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExcludeDumpster	Optional	System.Management.Automation.SwitchParameter	The *ExcludeDumpster* parameter specifies whether to exclude the Recoverable Items folder. You don't have to include a value with this parameter. If you don't specify this parameter, the Recoverable Items folder is copied with the following subfolders: • Deletions • Versions • Purges
ExcludeFolders	Optional	System.String[]	The *ExcludeFolders* parameter specifies the list of folders to exclude during the import.

			Folder names aren't case-sensitive, and there are no character restrictions. Use the following syntax: `<FolderName>/*` Use this syntax to denote a personal folder under the folder specified in the *SourceRootFolder* parameter, for example, "MyProjects" or "MyProjects/FY2010". `#<FolderName>#/*` Use this syntax to denote a well-known folder regardless of the folder's name in another language. For example, `#Inbox#` denotes the Inbox folder even if the Inbox is localized in Turkish, which is Gelen Kutusu. Well-known folders include the following types: • Inbox • SentItems • DeletedItems • Calendar • Contacts • Drafts • Journal • Tasks • Notes • JunkEmail • CommunicationHistory • Voicemail • Fax

			• Conflicts • SyncIssues • LocalFailures • ServerFailures If the user creates a personal folder with the same name as a well-known folder and the # symbol surrounding it, you can use a back slash (\) as an escape character to specify that folder. For example, if a user creates a folder named #Notes# and you want to specify that folder, but not the well-known Notes folder, use the following syntax: `\#Notes\#`. **Note:** Wildcard characters can't be used in folder names. If the *TargetRootFolder* parameter isn't specified when the Recoverable Items folder is imported, the recoverable item content is placed in the Recoverable Items folder of the target mailbox or archive.
IncludeFolders	Optional	System.String[]	The *IncludeFolders* parameter specifies the

			list of folders to include during the import. Folder names aren't case-sensitive, and there are no character restrictions. Use the following syntax: `<FolderName>/*` Use this syntax to denote a personal folder under the folder specified in the *SourceRootFolder* parameter, for example, "MyProjects" or "MyProjects/FY2010". `#<FolderName>#/*` Use this syntax to denote a well-known folder regardless of the folder's name in another language. For example, `#Inbox#` denotes the Inbox folder even if the Inbox is localized in Turkish, which is Gelen Kutusu. Well-known folders include the following types: • Inbox • SentItems • DeletedItems • Calendar • Contacts • Drafts • Journal • Tasks • Notes • JunkEmail • CommunicationHistory

			• Voicemail • Fax • Conflicts • SyncIssues • LocalFailures • ServerFailures If the user creates a personal folder with the same name as a well-known folder and the # symbol surrounding it, you can use a back slash (\) as an escape character to specify that folder. For example, if a user creates a folder named #Notes# and you want to specify that folder, but not the well-known Notes folder, use the following syntax: `\#Notes\#`. **Note:** Wildcard characters can't be used in folder names.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
IsArchive	Optional	System.Management.Automation.SwitchPar	The *IsArchive* switch specifies that you're

		ameter	importing the .pst file into the user's archive.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of large items to skip if the request encounters such items in the mailbox. Use 0 to not skip any large items. If any number above 50 is specified, the *AcceptLargeDataLoss* parameter must also be specified. The default value is 0. We recommend that you use the default value of 0 and increase the *LargeItemLimit* only when large items are encountered.
Name	Optional	System.String	The *Name* parameter specifies the name of the specific request for tracking and display purposes. Because you can have multiple import requests per mailbox, Exchange precedes the name with the mailbox's alias. For example, if you create an import request for a user's mailbox that

			has the alias Kweku and specify the value of this parameter as PC1toArchive, the identity of this import request is Kweku\PC1toArchive. If you don't specify a name using this parameter, Exchange generates up to 10 request names per mailbox, which is MailboxImport*X* (where *X* = 0–9). The identity of the request is displayed and searchable as <*alias*>\MailboxImport*X*.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	The *Priority* parameter specifies the order in which this request should be processed in the request queue. Requests are processed in order, based on server health, status, priority, and last update time.
RemoteCredential	Optional	System.Management.Automation.PSCredential	The *RemoteCredential* parameter specifies an administrator who has permission to perform the mailbox import request. For example,

			Administrator@humongousinsurance.com.
RemoteHostName	Optional	Microsoft.Exchange.Data.Fqdn	The *RemoteHostName* parameter specifies the FQDN of the cross-forest organization from which you're making the import request.
SkipMerging	Optional	Microsoft.Exchange.Management.RecipientTasks.SkippableMergeComponent[]	The *SkipMerging* parameter specifies steps in the import that should be skipped. This parameter is used primarily for debugging purposes.
SourceRootFolder	Optional	System.String	The *SourceRootFolder* parameter specifies the root folder of the .pst file from which data is imported. When specified, the folder hierarchy outside the value of the *SourceRootFolder* parameter isn't imported, and the *SourceRootFolder* parameter is mapped to the *TargetRootFolder* parameter. If this parameter isn't specified, the command imports all

			folders.
Suspend	Optional	System.Management.Automation.SwitchParameter	The *Suspend* switch specifies whether to suspend the request. If you use this switch, the request is queued, but the request won't reach the status of **InProgress** until you resume the request with the relevant resume cmdlet. You don't have to specify a value with this switch.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description about why the request was suspended. You can only use this parameter if you specify the *Suspend* parameter.
TargetRootFolder	Optional	System.String	The *TargetRootFolder* parameter specifies the top-level mailbox folder that the imported content is placed in. If you don't specify this parameter, the command imports folders to the top of the folder structure in the target mailbox or archive. If the folder already exists,

			content is merged under existing folders, and new folders are created if they don't already exist in the target folder structure.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WorkloadType	Optional	Microsoft.Exchange.MailboxReplicationService.RequestWorkloadType	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailboxImportRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-MailboxImportRequest** cmdlet to remove fully or partially completed import requests. Completed import requests aren't automatically cleared. Requests need to be removed by using the **Remove-MailboxImportRequest** cmdlet. Multiple import requests can exist against the same mailbox if you provide a distinct import request name.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

Note:

Removing a partially completed import request removes the request from the Microsoft Exchange Mailbox Replication service (MRS) job queue. Any import progress that was made until the removal won't be reverted.

```
Remove-MailboxImportRequest -Identity <MailboxImportRequestIdParameter>
<COMMON PARAMETERS>


Remove-MailboxImportRequest -RequestGuid <Guid> -RequestQueue
<DatabaseIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the second import request for Ayla's mailbox Ayla\MailboxImport1.

```
Remove-MailboxImportRequest -Identity "Ayla\MailboxImport1"
```

EXAMPLE 2

This example cancels the import request by using the *RequestGuid* parameter for a mailbox or archive on MBXDB01.

```
Remove-MailboxImportRequest -RequestQueue MBXDB01 -
RequestGuid 25e0eaf2-6cc2-4353-b83e-5cb7b72d441f
```

EXAMPLE 3

This example removes all completed import requests.

```
Get-MailboxImportRequest -Status Completed | Remove-
MailboxImportRequest
```

Detailed Description

The parameter set that requires the *Identity* parameter allows you to remove a fully or partially completed import request.

The parameter set that requires the *RequestGuid* and *RequestQueue* parameters is used for MRS debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxImportRequestIdParameter	The *Identity* parameter specifies the identity of the import request. By default, import requests are named <*alias*>\MailboxImport*X* (where *X* = 0–9). If you created the request using the *Name* parameter, use the following syntax: <*alias*>\<*name*>. You can't use this parameter in conjunction

			with the *RequestGuid* parameter.
RequestGuid	Required	System.Guid	The *RequestGuid* parameter specifies the unique identifier for the import request. To find the import request GUID, use the Get-MailboxImportRequest cmdlet. If you specify the *RequestGuid* parameter, you must also specify the *RequestQueue* parameter. You can't use this parameter in conjunction with the *Identity* parameter.
RequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies the mailbox database on which the request is being performed. This parameter accepts the following values: • GUID of the database • Database name
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Resume-MailboxImportRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Resume-MailboxImportRequest** cmdlet to resume an import request that was suspended or failed.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Resume-MailboxImportRequest -Identity <MailboxImportRequestIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resumes the second import request for Kweku's mailbox kweku\MailboxImport1

```
Resume-MailboxImportRequest -Identity kweku\MailboxImport1
```

EXAMPLE 2

This example resumes all failed import requests.

```
Get-MailboxImportRequest -Status Failed | Resume-
MailboxImportRequest
```

Detailed Description

This cmdlet can be pipelined with the **Get-MailboxImportRequest** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxImportRequestIdParameter	The *Identity* parameter specifies the identity of the import request. By default, import requests are named <*alias*>\MailboxImport*X* (where *X* = 0–9). If you created the request using the *Name* parameter, use the following syntax: <*alias*>\<*name*>.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Dat	The *DomainController*

		a.Fqdn	parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxImportRequest

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-MailboxImportRequest** cmdlet to change import request options after the request has been created. You can use the **Set-MailboxImportRequest** cmdlet to recover from failed import requests.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxImportRequest [-AcceptLargeDataLoss <SwitchParameter>] [-
BadItemLimit <Unlimited>] [-BatchName <String>] [-CompletedRequestAgeLimit
<Unlimited>] [-InternalFlags <InternalMrsFlag[]>] [-LargeItemLimit
<Unlimited>] [-Priority <Lowest | Lower | Low | Normal | High | Higher |
Highest | Emergency>] [-RemoteCredential <PSCredential>] [-RemoteHostName
<Fqdn>] [-SkipMerging <SkippableMergeComponent[]>] <COMMON PARAMETERS>


Set-MailboxImportRequest -RehomeRequest <SwitchParameter> <COMMON
PARAMETERS>


COMMON PARAMETERS: -Identity <MailboxImportRequestIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the import request Kweku\Import to accept up to five corrupted mailbox items.

```
Set-MailboxImportRequest -Identity "Kweku\Import" -
BadItemLimit 5
```

EXAMPLE 2

This example finds all import requests that have a status of Suspended, and then gives them a batch name of April14.

```
Get-MailboxImportRequest -Status Suspended | Set-
MailboxImportRequest -BatchName April14
```

Detailed Description

You can pipeline the **Set-MailboxImportRequest** cmdlet from the Get-MailboxImportRequest cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxImportRequestIdParameter	The *Identity* parameter specifies the identity of the import request. By default, import requests are named <*alias*>\MailboxImport*X* (where *X* = 0–9). If you specified a name for the import request with the New-MailboxImportRequest cmdlet, use the following syntax: <*alias*>\<*name*>.
RehomeRequest	Required	System.Management.Automation.SwitchParameter	The *RehomeRequest* parameter specifies to the Microsoft Exchange Mailbox Replication service (MRS) that the request needs to be moved to the same database as the mailbox

			being imported. This parameter is used primarily for debugging purposes.
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51 or higher. Items are considered corrupted if the item can't be read from the source database or can't be written to the target database. Corrupted items won't be available in the destination mailbox or .pst file.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use `0` to not skip bad items. The valid input range for this parameter is from `0` through `2147483647`. The default value is `0`. We recommend that

			you keep the default value 0 and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the *BadItemLimit* parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process is complete.
BatchName	Optional	System.String	The *BatchName* parameter specifies the name of the batch.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the request is kept after it has completed before being automatically removed.

			The default *CompletedRequestAgeLimit* parameter value is 30 days.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of large items

			to skip if the request encounters such items in the mailbox. Use 0 to not skip any large items. If any number above 50 is specified, the *AcceptLargeDataLoss* parameter must also be specified. The default value is 0. We recommend that you use the default value of 0 and increase the *LargeItemLimit* only when large items are encountered.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	The *Priority* parameter specifies the order in which this request is processed in the request queue. Requests are processed in order, based on server health, status, priority, and last update time.
RemoteCredential	Optional	System.Management.Automation.PSCredential	The *RemoteCredential* parameter specifies an administrator who has permission to perform the mailbox import request. For example, Administrator@

			humongousinsurance.com.
RemoteHostName	Optional	Microsoft.Exchange.Data.Fqdn	The *RemoteHostName* parameter specifies the FQDN of the cross-forest organization from which you're configuring the import request.
SkipMerging	Optional	Microsoft.Exchange.Management.RecipientTasks.SkippableMergeComponent[]	The *SkipMerging* parameter specifies the steps in the import that should be skipped. This parameter is used primarily for debugging purposes.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Suspend-MailboxImportRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Suspend-MailboxImportRequest** cmdlet to suspend an import request any time after the request was created, but before the request reaches the status of `Completed`. You can resume the move request by using the Resume-MailboxImportRequest cmdlet.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Suspend-MailboxImportRequest -Identity <MailboxImportRequestIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-SuspendComment
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example suspends the second import request for Ayla's mailbox with the identity Ayla \MailboxImport1.

```
Suspend-MailboxImportRequest -Identity "Ayla
\MailboxImport1"
```

EXAMPLE 2

This example suspends all import requests that are in progress by using the Get-MailboxImportRequest cmdlet to retrieve all requests with a Status of `InProgress`, and then pipelining the output to the **Suspend-MailboxImportRequest** cmdlet with the suspend comment "Resume after 22:00 (10 P.M.)".

```
Get-MailboxImportRequest -Status InProgress | Suspend-MailboxImportRequest -SuspendComment "Resume after 22:00 (10 P.M.)"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxImportRequestIdParameter	The *Identity* parameter specifies the identity of the import request. By default, import requests are named <*alias*>\MailboxImport*X* (where *X* = 0–9). If you created the request by using the *Name* parameter, use the following syntax: <*alias*>\<*name*>.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description about why the request was suspended. You can only use this parameter if you specify the *Suspend* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxImportRequestStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxImportRequestStatistics** cmdlet to view detailed information about import requests.

Note:

This cmdlet is available only in the Mailbox Import Export role, and by default, that role isn't assigned to a role group. To use this cmdlet, you need to add the Mailbox Import Export role to a role group (for example, to the Organization Management role group). For more information, see the "Add a role to a role group" section in Manage role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxImportRequestStatistics -Identity
<MailboxImportRequestIdParameter> <COMMON PARAMETERS>

Get-MailboxImportRequestStatistics -RequestQueue <DatabaseIdParameter> [-
RequestGuid <Guid>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-Diagnostic <SwitchParameter>] [-DomainController
```

```
<Fqdn>] [-IncludeReport <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns the default statistics for the second import request for Tony Smith. The type of information returned by default includes name, mailbox, and status.

```
Get-MailboxImportRequestStatistics -Identity Tony
\MailboxImport1
```

EXAMPLE 2

This example returns the detailed statistics for the second import request for Tony Smith's mailbox and exports the report to a .csv file.

```
Get-MailboxImportRequestStatistics -Identity Tony
\MailboxImport1 | Export-CSV \\SERVER01
\ImportRequest_Reports\Tony_Importstats.csv
```

EXAMPLE 3

This example returns additional information about the import request for Tony Smith's mailbox by using the *IncludeReport* parameter and by pipelining the results to the **Format-List** command.

```
Get-MailboxImportRequestStatistics -Identity Tony\LegalHold
-IncludeReport | Format-List
```

EXAMPLE 4

This example returns additional information for all the import requests that have a status of `Failed` by using the *IncludeReport* parameter, and then saves the information to the text file AllImportReports.txt.

```
Get-MailboxImportRequest -Status Failed | Get-
MailboxImportRequestStatistics -IncludeReport | Format-List
> AllImportReports.txt
```

Detailed Description

You can pipeline the **Get-MailboxImportRequestStatistics** cmdlet from the **Get-MailboxImportRequest** cmdlet.

The *RequestQueue* parameter syntax set is for debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Import Export" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxImportRequestIdParameter	The *Identity* parameter specifies the identity of the import request. By default, import requests are named <*alias*>\MailboxImport*X* (where *X* = 0–9). If you specified a name when you created the import request, use the following syntax: <*alias*>\<*name*>.
RequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies the mailbox database on which the mailbox or archive of the request resides. You can use one of the following values: • GUID of the database • Database name This parameter can't be used in conjunction with the *Identity* or *MRSInstance* parameters.
Diagnostic	Optional	System.Management.	The *Diagnostic* switch

		Automation.SwitchParameter	specifies whether to retrieve extremely detailed information about the mailbox import request statistics.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
IncludeReport	Optional	System.Management.Automation.SwitchParameter	The *IncludeReport* switch specifies whether to return additional details, which can be used for troubleshooting.
RequestGuid	Optional	System.Guid	The *RequestGuid* parameter specifies the unique identifier for the import request. To find the import request GUID, use the **Get-MailboxImportRequest** cmdlet. If you specify the *RequestGuid* parameter, you must also specify the *RequestQueue* parameter. You can't use this parameter in conjunction with the *Identity*

			parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-MailboxPermission

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-19*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Add-MailboxPermission** cmdlet to add permissions to a mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-MailboxPermission -Identity <MailboxIdParameter> -Owner
<SecurityPrincipalIdParameter> <COMMON PARAMETERS>


Add-MailboxPermission -AccessRights <MailboxRights[]> -Identity
<MailboxIdParameter> -User <SecurityPrincipalIdParameter> [-AutoMapping
<$true | $false>] [-Deny <SwitchParameter>] [-InheritanceType <None | All
| Descendents | SelfAndChildren | Children>] <COMMON PARAMETERS>


Add-MailboxPermission -Instance <MailboxAcePresentationObject> [-
AccessRights <MailboxRights[]>] [-AutoMapping <$true | $false>] [-Deny
<SwitchParameter>] [-Identity <MailboxIdParameter>] [-InheritanceType
<None | All | Descendents | SelfAndChildren | Children>] [-User
<SecurityPrincipalIdParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-IgnoreDefaultScope <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example grants Kevin Kelly full access to Terry Adams's mailbox.

Note:
The *Identity* parameter requires the full name of the user to be enclosed in quotation marks (").

```
Add-MailboxPermission -Identity "Terry Adams" -User
KevinKelly -AccessRights FullAccess -InheritanceType All
```

EXAMPLE 2

This example sets Tony Smith as the owner of the resource mailbox Room 222.

```
Add-MailboxPermission -Identity "Room 222" -Owner "Tony
Smith"
```

EXAMPLE 3

This example grants the user Mark Steele Full Access permission to Jeroen Cool's mailbox and disables the auto-mapping feature.

```
Add-MailboxPermission -Identity JeroenC -User 'Mark Steele'
-AccessRights FullAccess -InheritanceType All -AutoMapping
$false
```

EXAMPLE 4

This example assigns full access permissions to all user mailboxes in an Exchange Online or Office 365 environment.

1. Connect to Exchange Online by using remote PowerShell. For info about how to do this, go to the following Microsoft website:**Connect to Exchange Online using remote PowerShell**.
2. Enter a command using the following syntax to assign full access permissions to all user mailboxes:

```
Get-Mailbox -ResultSize unlimited -Filter
{(RecipientTypeDetails -eq 'UserMailbox') -and (Alias -ne
'Admin')} | Add-MailboxPermission -User <user, role group
or security group> -AccessRights fullaccess -
InheritanceType all
```

For example, to assign full access permissions to all user mailboxes for the administrator account admin@contoso.com, run the following command.

```
Get-Mailbox -ResultSize unlimited -Filter
```

```
{(RecipientTypeDetails -eq 'UserMailbox') -and (Alias -ne
'Admin')} | Add-MailboxPermission -User admin@contoso.com -
AccessRights fullaccess -InheritanceType all
```

Detailed Description

Running this cmdlet updates the Active Directory object specified by the *Identity* parameter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Permissions and delegation" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccessRights	Required	Microsoft.Exchange.Management.RecipientTasks.MailboxRights[]	The *AccessRights* parameter specifies the rights needed to perform the operation. Valid values include: • `FullAccess` • `ExternalAccount` • `DeleteItem` • `ReadPermission` • `ChangePermission` • `ChangeOwner`
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identity of the mailbox that's getting permissions added. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name`

			Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Instance	Required	Microsoft.Exchange.Management.RecipientTasks.MailboxAcePresentationObject	The *Instance* parameter is no longer used and will be deprecated.
Owner	Required	Microsoft.Exchange.Configuration.Tasks.SecurityPrincipalIdParameter	The *Owner* parameter specifies the owner of the mailbox object.

User	Required	Microsoft.Exchange.Configuration.Tasks.SecurityPrincipalIdParameter	The *User* parameter specifies the user mailbox that the permissions are being granted to on the other mailbox.
AutoMapping	Optional	System.Boolean	The *AutoMapping* parameter specifies whether to ignore the auto-mapping feature in Microsoft Outlook. If a user is granted Full Access permissions to another user's mailbox or to a shared mailbox, Outlook, through Autodiscover, automatically loads all mailboxes to which the user has full access. This parameter accepts `$true` or `$false` values.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
Deny	Optional	System.Management.	The *Deny* switch specifies

		Automation.SwitchParameter	whether to deny permissions to the user on the mailbox.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController*

			parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
InheritanceType	Optional	System.DirectoryServices.ActiveDirectorySecurityInheritance	The *InheritanceType* parameter specifies whether permissions are inherited by folders within the mailbox.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxPermission

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxPermission** cmdlet to retrieve permissions on a mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxPermission [-User <SecurityPrincipalIdParameter>] <COMMON
PARAMETERS>

Get-MailboxPermission [-Owner <SwitchParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: -Identity <MailboxIdParameter> [-Credential
<PSCredential>] [-DomainController <Fqdn>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns permissions on the mailbox by its SMTP address john@contoso.com.

```
Get-MailboxPermission -Identity john@contoso.com | Format-
List
```

EXAMPLE 2

This example returns permissions that the user Ayla has on John's mailbox.

```
Get-MailboxPermission -Identity john@contoso.com -User
"Ayla"
```

EXAMPLE 3

This example returns the owner information for the resource mailbox Room222.

```
Get-MailboxPermission -Identity Room222 -Owner
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Permissions and delegation" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter identifies the mailbox. You can use one of the following values: • GUID • ADObjectID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This

			credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Owner	Optional	System.Management.Automation.SwitchParameter	The *Owner* parameter returns the owner information for the mailbox identified in the *Identity* parameter. This parameter can't be used with the *User* parameter.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient

			scope to include all recipients in the forest, and if you don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Exchange.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of recipient objects returned.
User	Optional	Microsoft.Exchange.Configuration.Tasks.SecurityPrincipalIdParameter	The *User* parameter specifies the UPN, *domain**user*, or the alias of the user. This parameter can't be used with the *Owner* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailboxPermission

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MailboxPermission** cmdlet to remove permissions from a user's mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MailboxPermission -AccessRights <MailboxRights[]> -Identity
<MailboxIdParameter> -User <SecurityPrincipalIdParameter> [-Deny
<SwitchParameter>] [-InheritanceType <None | All | Descendents |
SelfAndChildren | Children>] <COMMON PARAMETERS>


Remove-MailboxPermission -Instance <MailboxAcePresentationObject> [-
AccessRights <MailboxRights[]>] [-Deny <SwitchParameter>] [-Identity
<MailboxIdParameter>] [-InheritanceType <None | All | Descendents |
SelfAndChildren | Children>] [-User <SecurityPrincipalIdParameter>]
<COMMON PARAMETERS>


Remove-MailboxPermission -Identity <MailboxIdParameter> <COMMON
PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-IgnoreDefaultScope <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes user Test2's full access rights to Test1's mailbox.

```
Remove-MailboxPermission -Identity Test1 -User Test2 -
AccessRights FullAccess -InheritanceType All
```

Detailed Description

The **Remove-MailboxPermission** cmdlet allows you to remove permissions from a user's mailbox, for example, removing full access to another user's mailbox.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Permissions and delegation" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccessRights	Required	Microsoft.Exchange.Management.RecipientTasks.MailboxRights[]	The *AccessRights* parameter specifies the rights required to perform the operation. You can use the following values: • `FullAccess` • `SendAs` • `ExternalAccount` • `DeleteItem` • `ReadPermission` • `ChangePermission` • `ChangeOwner`
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter identifies the mailbox. You can use the following values: • GUID • ADObjectID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
Instance	Required	Microsoft.Exchange.M	This parameter is

		anagement.RecipientTasks.MailboxAcePresentationObject	available only in on-premises Exchange 2013. The *Instance* parameter enables you to pass an entire object to the command to be processed. It's mainly used in scripts where an entire object must be passed to the command.
User	Required	Microsoft.Exchange.Configuration.Tasks.SecurityPrincipalIdParameter	The *User* parameter specifies the user mailbox that will get permissions removed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Deny	Optional	System.Management.Automation.SwitchParameter	The *Deny* parameter denies permissions to the user on the Active Directory object.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity*

			parameter. Other forms of identification, such as alias or GUID, aren't accepted.
InheritanceType	Optional	System.DirectoryServices.ActiveDirectorySecurityInheritance	The *InheritanceType* parameter specifies whether permissions are inherited to folders within the mailbox.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxRestoreRequest

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxRestoreRequest** cmdlet to view detailed status of an ongoing restore request that was initiated by using the New-MailboxRestoreRequest cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxRestoreRequest [-AccountPartition
<AccountPartitionIdParameter>] [-Identity
<MailboxRestoreRequestIdParameter>] [-Organization
<OrganizationIdParameter>] <COMMON PARAMETERS>


Get-MailboxRestoreRequest [-AccountPartition
<AccountPartitionIdParameter>] [-BatchName <String>] [-HighPriority <$true
| $false>] [-Name <String>] [-Organization <OrganizationIdParameter>] [-
RequestQueue <DatabaseIdParameter>] [-SourceDatabase
<DatabaseIdParameter>] [-Status <None | Queued | InProgress |
AutoSuspended | CompletionInProgress | Synced | Completed |
CompletedWithWarning | Suspended | Failed>] [-Suspend <$true | $false>] [-
TargetMailbox <MailboxOrMailUserIdParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns the status of the in-progress and queued restore request with the identity ayla \MailboxRestore.

```
Get-MailboxRestoreRequest -Identity "ayla\MailboxRestore"
```

EXAMPLE 2

This example returns the status of in-progress and queued restore requests that are being restored to the mailbox database MBD01.

```
Get-MailboxRestoreRequest -RequestQueue MBD01
```

EXAMPLE 3

This example returns all restore requests that have the name RestoreToMBD01 where the restore request has been suspended.

```
Get-MailboxRestoreRequest -Name "RestoreToMBD01" -Suspend
$true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox restore request" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
BatchName	Optional	System.String	The *BatchName* parameter specifies the name given to a batch of restore requests. You can't use this parameter with the *Identity* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
HighPriority	Optional	System.Boolean	The *HighPriority* parameter specifies that restore requests with the specified priority range are returned. Use one of

			the following values: • `$true` Returns restore requests that were created with a priority of Emergency, Highest, Higher, or High • `$false` Returns restore requests that were created with a priority of Normal, Low, Lower, or Lowest
Identity	Optional	Microsoft.Exchange.MailboxReplicationService.MailboxRestoreRequestIdParameter	The *Identity* parameter specifies the identity of the restore request. The *Identity* parameter consists of the alias of the mailbox to be restored and the name that was specified when the restore request was created. The identity of the restore request uses the following syntax: <*alias*>\<*name*>. If you didn't specify a name for the restore request when it was created, Exchange automatically generated the default name MailboxRestore. Exchange generates up to 10 names, starting with MailboxRestore and then MailboxRestore*X* (where *X* = 1–9).

			You can't use this parameter with the *Name* parameter.
Name	Optional	System.String	The *Name* parameter specifies that any restore request that has the specified name is returned. Use this parameter to search on the name you provided when you created the restore request. If you didn't specify a name for the restore request when it was created, Exchange automatically generated the default name MailboxRestore. Exchange generates up to 10 names, starting with MailboxRestore and then MailboxRestore*X* (where *X* = 1–9). You can't use this parameter with the *Identity* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

RequestQueue	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies the mailbox database on which the mailbox or archive of the request resides. You can use one of the following values: • GUID of the database • Database name
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
SourceDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *SourceDatabase* parameter specifies that the cmdlet should only return restore requests for mailboxes that are being restored from the specified source database. You can use the following values: • GUID of the database • Database name You can't use this parameter with the *Identity* parameter.

Status	Optional	Microsoft.Exchange.Data.Directory.Recipient.RequestStatus	The *Status* parameter specifies that restore requests with the specified status are returned. You can use the following values: • `AutoSuspended` • `Completed` • `CompletedWithWarning` • `CompletionInProgress` • `Failed` • `InProgress` • `None` • `Queued` • `Suspended` You can't use this parameter with the *Identity* parameter.
Suspend	Optional	System.Boolean	The *Suspend* parameter specifies that the cmdlet should only return restore requests that have been suspended. This parameter accepts `$true` or `$false`. You can't use this parameter with the *Identity* parameter.
TargetMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxOrMailUserIdParameter	The *TargetMailbox* parameter specifies the identity of the target mailbox. You can use the following values: • GUID • Distinguished name (DN)

			• *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias You can't use this parameter with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailboxRestoreRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-02*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-MailboxRestoreRequest** cmdlet to restore a soft-deleted or disconnected mailbox. This cmdlet starts the process of moving content from the soft-deleted mailbox, disabled mailbox, or any mailbox in a recovery database into a connected primary or archive mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MailboxRestoreRequest -SourceDatabase <DatabaseIdParameter> [-
AllowLegacyDNMismatch <SwitchParameter>] [-AssociatedMessagesCopyOption
<DoNotCopy | MapByMessageClass | Copy>] [-ConflictResolutionOption
<KeepSourceItem | KeepLatestItem | KeepAll>] [-ExcludeDumpster
<SwitchParameter>] [-ExcludeFolders <String[]>] [-IncludeFolders <String[]
>] <COMMON PARAMETERS>
```

```
New-MailboxRestoreRequest -RemoteDatabaseGuid <Guid> -RemoteHostName
<Fqdn> -RemoteRestoreType <None | RecoveryDatabase | DisconnectedMailbox |
SoftDeletedRecipient> [-AllowLegacyDNMismatch <SwitchParameter>] [-
AssociatedMessagesCopyOption <DoNotCopy | MapByMessageClass | Copy>] [-
ConflictResolutionOption <KeepSourceItem | KeepLatestItem | KeepAll>] [-
ExcludeDumpster <SwitchParameter>] [-ExcludeFolders <String[]>] [-
IncludeFolders <String[]>] [-RemoteCredential <PSCredential>] <COMMON
PARAMETERS>


COMMON PARAMETERS: -SourceStoreMailbox <StoreMailboxIdParameter> -
TargetMailbox <MailboxOrMailUserIdParameter> [-AcceptLargeDataLoss
<SwitchParameter>] [-BadItemLimit <Unlimited>] [-BatchName <String>] [-
CompletedRequestAgeLimit <Unlimited>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-InternalFlags <InternalMrsFlag[]>] [-
LargeItemLimit <Unlimited>] [-Name <String>] [-Priority <Lowest | Lower |
Low | Normal | High | Higher | Highest | Emergency>] [-SkipMerging
<SkippableMergeComponent[]>] [-SourceRootFolder <String>] [-Suspend
<SwitchParameter>] [-SuspendComment <String>] [-TargetIsArchive
<SwitchParameter>] [-TargetRootFolder <String>] [-WhatIf
[<SwitchParameter>]] [-WorkloadType <None | Local | Onboarding |
Offboarding | TenantUpgrade | LoadBalancing | Emergency |
RemotePstIngestion | SyncAggregation>]
```

Examples

EXAMPLE 1

To create a restore request, you must provide the **DisplayName**, **LegacyDN**, or **MailboxGUID** for the soft-deleted or disabled mailbox. This example uses the Get-MailboxStatistics cmdlet to return the **DisplayName**, **LegacyDN**, **MailboxGUID**, and **DisconnectReason** for all mailboxes on mailbox database MBD01 that have a disconnect reason of `SoftDeleted` or `Disabled`.

```
Get-MailboxStatistics -Database MBD01 | Where
{ $_.DisconnectReason -eq "SoftDeleted" -or
$_.DisconnectReason -eq "Disabled" } | Format-List
LegacyDN, DisplayName, MailboxGUID, DisconnectReason
```

This example restores the source mailbox with the MailboxGUID 1d20855f-fd54-4681-98e6-e249f7326ddd on mailbox database MBD01 to the target mailbox with the alias Ayla.

```
New-MailboxRestoreRequest -SourceDatabase "MBD01" -
SourceStoreMailbox 1d20855f-fd54-4681-98e6-e249f7326ddd -
TargetMailbox Ayla
```

EXAMPLE 2

This example restores the content of the source mailbox with the **DisplayName** of Tony Smith on mailbox database MBD01 to the archive mailbox for Tony@contoso.com.

```
New-MaiboxRestoreRequest -SourceDatabase "MBD01" -
SourceStoreMailbox "Tony Smith" -TargetMailbox
Tony@contoso.com -TargetIsArchive
```

Detailed Description

When mailboxes are moved from a Microsoft Exchange Server 2013 or Exchange Server 2010 Service Pack 1 or later versions database to any other database, Exchange doesn't fully delete the mailbox from the source database immediately upon completion of the move. Instead, the mailbox in the source mailbox database is switched to a *soft-deleted* state, which allows mailbox data to be accessed during a mailbox restore operation by using the new **MailboxRestoreRequest** cmdlet set. The soft-deleted mailboxes are retained in the source database until either the deleted mailbox retention period expires or you use the **Remove-StoreMailbox** cmdlet to purge the mailbox.

To view soft-deleted mailboxes, run the Get-MailboxStatistics cmdlet against a database and look for results that have a **DisconnectReason** with a value of `SoftDeleted`. For more information, see EXAMPLE 1 later in this topic.

A mailbox is marked as Disabled a short time after the **Disable-Mailbox** or **Remove-Mailbox** command completes.

Note:

The mailbox won't be marked as Disabled until the Microsoft Exchange Information Store service determines that Active Directory has been updated with the disabled mailbox's information. You can expedite the process by running the **Update-StoreMailboxState** cmdlet against that database.

Exchange retains disabled mailboxes in the mailbox database based on the deleted mailbox retention settings configured for that mailbox database. After the specified period of time, the mailbox is permanently deleted.

To view disabled mailboxes, run the **Get-MailboxStatistics** cmdlet against a database and look for results that have a **DisconnectReason** with a value of `Disabled`. For more information, see EXAMPLE 1 later in this topic.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox restore request" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
RemoteDatabaseGuid	Required	System.Guid	This parameter is reserved for internal Microsoft use.
RemoteHostName	Required	Microsoft.Exchange.Data.Fqdn	This parameter is reserved for internal Microsoft use.

RemoteRestoreType	Required	Microsoft.Exchange.Management.RecipientTasks.RemoteRestoreType	This parameter is reserved for internal Microsoft use.
SourceDatabase	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *SourceDatabase* parameter specifies the identity of the database from which you're restoring the soft-deleted or disconnected mailbox.
SourceStoreMailbox	Required	Microsoft.Exchange.Configuration.Tasks.StoreMailboxIdParameter	The *SourceStoreMailbox* parameter specifies the identity of the mailbox from which you want to restore content. This parameter accepts the following values: • MailboxGUID • LegacyExchangeDN • DisplayName You can find this information by running the Get-MailboxStatistics cmdlet.
TargetMailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxOrMailUserIdParameter	The *TargetMailbox* parameter specifies the identity of the mailbox or mail user to which you want to restore content. The target mailbox or mail user needs to exist before

			you can run this command successfully. This parameter accepts the following values: • GUID • Alias • LegacyExchangeDN • *Domain\Account Name* • SMTP address
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51 or higher. Items are considered corrupted if the item can't be read from the source database or can't be written to the target database. Corrupted items won't be available in the destination mailbox or .pst file.
AllowLegacyDNMismatch	Optional	System.Management.Automation.SwitchParameter	The *AllowLegacyDNMismatch* parameter specifies that if the LegacyExchangeDN of the source physical mailbox and the target mailbox don't match, continue the operation. By

			default, this cmdlet checks to make sure that the LegacyExchangeDN on the source physical mailbox is present on the target user in the form of the LegacyExchangeDN or an X500 proxy address that corresponds to the LegacyExchangeDN. This check prevents you from accidentally restoring a source mailbox into the incorrect target mailbox. You don't have to provide a value with this parameter.
AssociatedMessagesCopyOption	Optional	Microsoft.Exchange.MailboxReplicationService.FAICopyOption	The *AssociatedMessagesCopyOption* parameter specifies whether associated messages are copied when the request is processed. Associated messages are special messages that contain hidden data with information about rules, views, and forms. By default, associated messages are copied. This parameter accepts the

			following values: • `DoNotCopy` The associated messages aren't copied. • `MapByMessageClass` This option finds the corresponding associated message by looking up the **MessageClass** attribute of the source message. If there's an associated message of this class in both source and target folders, it overwrites the associated message in the target. If there isn't an associated message in the target, it creates a copy in the target. • `Copy` This option copies associated messages from the source to the target. If the same message type exists both in the source and the target location, these associated messages are duplicated. This is the default option. **Note:** Content filtering doesn't apply to associated messages.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use `0` to not

			skip bad items. The valid input range for this parameter is from 0 through 2147483647. The default value is 0. We recommend that you keep the default value 0 and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the *BadItemLimit* parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process is complete.
BatchName	Optional	System.String	The *BatchName* parameter specifies a descriptive name for restoring a batch of mailboxes. You can use the name in the *BatchName* parameter as

			a string search when you use the **Get-MailboxRestoreRequest** cmdlet.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the status of a completed restore request is set to `Completed`. If this parameter is set to a value of `0`, the status is cleared immediately instead of being changed to `Completed`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ConflictResolutionOption	Optional	Microsoft.Exchange.MailboxReplicationService.ConflictResolutionOption	The *ConflictResolutionOption* parameter specifies the action for the Microsoft Exchange Mailbox Replication service (MRS) to take if there are

			multiple matching messages in the target. This parameter takes the following values: • `KeepSourceItem` • `KeepLatestItem` • `KeepAll` The default value is `KeepSourceItem`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExcludeDumpster	Optional	System.Management.Automation.SwitchParameter	The *ExcludeDumpster* parameter specifies whether to exclude the Recoverable Items folder. You don't have to include a value with this parameter. If you don't specify this parameter, the Recoverable Items folder is copied with the following subfolders: • Deletions • Versions • Purges
ExcludeFolders	Optional	System.String[]	The *ExcludeFolders* parameter specifies the

			list of folders to exclude during the restore request. Folder names aren't case-sensitive, and there are no character restrictions. Use the following syntax: `<FolderName>/*` Use this syntax to denote a personal folder under the folder specified in the *SourceRootFolder* parameter, for example, "MyProjects" or "MyProjects/FY2010". `#<FolderName>#/*` Use this syntax to denote a well-known folder regardless of the folder's name in another language. For example, `#Inbox#` denotes the Inbox folder even if the Inbox is localized in Turkish, which is Gelen Kutusu. Well-known folders include the following types: • Inbox • SentItems • DeletedItems • Calendar • Contacts • Drafts • Journal • Tasks • Notes • JunkEmail

			• CommunicationHistory • Voicemail • Fax • Conflicts • SyncIssues • LocalFailures • ServerFailures If the user creates a personal folder with the same name as a well-known folder and the # symbol surrounding it, you can use a back slash (\) as an escape character to specify that folder. For example, if a user creates a folder named #Notes# and you want to specify that folder, but not the well-known Notes folder, use the following syntax: `\#Notes\#`. **Note:** Wildcard characters can't be used in folder names.
IncludeFolders	Optional	System.String[]	The *IncludeFolders* parameter specifies the list of folder to include during the restore request. Folder names aren't case-sensitive, and there are no character restrictions. Use

			the following syntax: `<FolderName>/*` Use this syntax to denote a personal folder under the folder specified in the *SourceRootFolder* parameter, for example, "MyProjects" or "MyProjects/FY2010". `#<FolderName>#/*` Use this syntax to denote a well-known folder regardless of the folder's name in another language. For example, `#Inbox#` denotes the Inbox folder even if the Inbox is localized in Turkish, which is Gelen Kutusu. Well-known folders include the following types: • Inbox • SentItems • DeletedItems • Calendar • Contacts • Drafts • Journal • Tasks • Notes • JunkEmail • CommunicationHistory • Voicemail • Fax • Conflicts • SyncIssues • LocalFailures

			• ServerFailures If the user creates a personal folder with the same name as a well-known folder and the # symbol surrounding it, you can use a back slash (\) as an escape character to specify that folder. For example, if a user creates a folder named #Notes# and you want to specify that folder, but not the well-known Notes folder, use the following syntax: `\#Notes\#`. **Note:** Wildcard characters can't be used in folder names.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of items to skip in the mailbox because these items exceed the item size limit for the target mailbox database. Use 0 to not skip any large

			items. **Note:** If you set the *LargeItemLimit* parameter to 51 or higher, you have to include the *AcceptLargeDataLoss* parameter.
Name	Optional	System.String	The *Name* parameter specifies the name of the specific request for tracking and display purposes. Because you can have multiple restore requests per mailbox, Exchange precedes the name with the mailbox's alias. For example, if you create an export request for a user's mailbox that has the alias Kweku and specify the value of this parameter as RestoreFailedMoves, the identity of this export request is Kweku \RestoreFailedMoves. If you didn't specify a name for the restore request when it was created, Exchange automatically generates the default name MailboxRestore. Exchange

			generates up to 10 names, starting with MailboxRestore and then MailboxRestore*X* (where *X* = 1–9).
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	The *Priority* parameter specifies the priority of the mailbox restore request. Use one of the following values: • `Emergency` • `Highest` • `Higher` • `High` • `Normal` • `Low` • `Lower` • `Lowest`
RemoteCredential	Optional	System.Management.Automation.PSCredential	This parameter is reserved for internal Microsoft use.
SkipMerging	Optional	Microsoft.Exchange.Management.RecipientTasks.SkippableMergeComponent[]	The *SkipMerging* parameter specifies folder-related items to skip when restoring the mailbox. Use one of the following values: • `FolderRules` • `FolderACLs` • `InitialConnectionValidation` Use this parameter only if a restore request fails because of folder rules, folder access control lists (ACLs), or initial connection validation.

SourceRootFolder	Optional	System.String	The *SourceRootFolder* parameter specifies the root folder of the mailbox from which data is restored. If this parameter isn't specified, the command restores all folders.
Suspend	Optional	System.Management.Automation.SwitchParameter	The *Suspend* switch specifies whether to suspend the request. If you use this switch, the request is queued, but the request won't reach the status of **InProgress** until you resume the request with the relevant resume cmdlet. You don't have to specify a value with this switch.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description about why the request was suspended. You can only use this parameter if you specify the *Suspend* parameter.
TargetIsArchive	Optional	System.Management.Automation.SwitchParameter	The *TargetIsArchive* parameter specifies that the content is restored into the specified target

			mailbox's archive.
TargetRootFolder	Optional	System.String	The *TargetRootFolder* parameter specifies the top-level folder in which to restore data. If you don't specify this parameter, the command restores folders to the top of the folder structure in the target mailbox or archive. Content is merged under existing folders, and new folders are created if they don't already exist in the target folder structure.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WorkloadType	Optional	Microsoft.Exchange.MailboxReplicationService.RequestWorkloadT	The *WorkLoadType* parameter specifies the type of restore request based on the type of

		ype	Exchange deployment or the purpose of the restore request. Use one of the following values: • None • Local • Onboarding • Offboarding • TenantUpgrade • LoadBalancing • Emergency

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailboxRestoreRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-MailboxRestoreRequest** cmdlet to remove fully or partially completed restore requests.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MailboxRestoreRequest -Identity <MailboxRestoreRequestIdParameter>
<COMMON PARAMETERS>


Remove-MailboxRestoreRequest -RequestGuid <Guid> -RequestQueue
<DatabaseIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the second restore request Ayla\MailboxRestore1.

```
Remove-MailboxRestoreRequest -Identity "Ayla
\MailboxRestore1"
```

EXAMPLE 2

This example removes all restore requests that have the status of Completed.

```
Get-MailboxRestoreRequest -Status Completed | Remove-
MailboxRestoreRequest
```

EXAMPLE 3

This example cancels the restore request by using the *RequestGuid* parameter for a request stored on MBXDB01. The parameter set that requires the *RequestGuid* and *RequestQueue* parameters is used for MRS debugging purposes only. You should only use this parameter set if instructed by Microsoft Customer Service and Support.

```
Remove-MailboxRestoreRequest -RequestQueue MBXDB01 -
RequestGuid 25e0eaf2-6cc2-4353-b83e-5cb7b72d441f
```

Detailed Description

The parameter set that requires the *Identity* parameter allows you to remove a fully or partially completed restore request.

The parameter set that requires the *RequestGuid* and *RequestQueue* parameters is used for Microsoft Exchange Mailbox Replication service (MRS) debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox restore request" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.M	The *Identity* parameter

		ailboxReplicationService.MailboxRestoreRequestIdParameter	specifies the identity of the restore request. The *Identity* parameter consists of the alias of the mailbox to be restored and the name that was specified when the restore request was created. The identity of the restore request uses the following syntax: <*alias*>\<*name*>. If you didn't specify a name for the restore request when it was created, Exchange automatically generated a default name MailboxRestore. Exchange generates up to 10 names, starting with MailboxRestore and then MailboxRestore*X* (where *X* = 1–9).
RequestGuid	Required	System.Guid	The *RequestGuid* parameter specifies the unique identifier for the restore request. To find the GUID, use the Get-MailboxRestoreRequest cmdlet. If you specify the *RequestGuid* parameter, you must also specify the

			RequestQueue parameter. This parameter can't be used in conjunction with the *Identity* parameter.
RequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies the target mailbox database on which the mailbox or archive of the request resides. You can use one of the following values: • GUID of the database • Database name If you specify the *RequestQueue* parameter, you must also specify the *RequestGuid* parameter. This parameter can't be used in conjunction with the *Identity* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Da	The *DomainController*

		ta.Fqdn	parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Resume-MailboxRestoreRequest

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Resume-MailboxRestoreRequest** cmdlet to resume a restore request that was suspended or failed.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Resume-MailboxRestoreRequest -Identity <MailboxRestoreRequestIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resumes the restore request with the identity kweku\RestoreFromDB01.

```
Resume-MailboxRestoreRequest -Identity "kweku
\RestoreFromDB01"
```

EXAMPLE 2

This example resumes any restore request with the status of Failed.

```
Get-MailboxRestoreRequest -Status Failed | Resume-
MailboxRestoreRequest
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox restore request" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxRestoreRequestIdParameter	The *Identity* parameter specifies the identity of the restore request. The *Identity* parameter

			consists of the alias of the mailbox to be restored and the name that was specified when the restore request was created. The identity of the restore request uses the following syntax: *<alias>**<name>*. If you didn't specify a name for the restore request when it was created, Exchange automatically generated the default name MailboxRestore. Exchange generates up to 10 names, starting with MailboxRestore and then MailboxRestore*X* (where *X* = 1–9).
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value

			with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxRestoreRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-MailboxRestoreRequest** cmdlet to change restore request options after the request has been created. You can use this cmdlet to recover from failed restore requests.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxRestoreRequest [-AcceptLargeDataLoss <SwitchParameter>] [-
BadItemLimit <Unlimited>] [-BatchName <String>] [-CompletedRequestAgeLimit
<Unlimited>] [-InternalFlags <InternalMrsFlag[]>] [-LargeItemLimit
<Unlimited>] [-Priority <Lowest | Lower | Low | Normal | High | Higher |
Highest | Emergency>] [-SkipMerging <SkippableMergeComponent[]>] <COMMON
PARAMETERS>


Set-MailboxRestoreRequest -RehomeRequest <SwitchParameter> <COMMON
PARAMETERS>


COMMON PARAMETERS: -Identity <MailboxRestoreRequestIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-RemoteCredential
<PSCredential>] [-RemoteHostName <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the second restore request for Ayla\MailboxRestore1 to skip 10 corrupt mailbox items.

```
Set-MailboxRestoreRequest -Identity "Ayla\MailboxRestore1"
-BadItemLimit 10
```

EXAMPLE 2

This example changes the first restore request for Kweku's mailbox to skip 100 corrupt items. Because the *BadItemLimit* is greater than 50, the *AcceptLargeDataLoss* parameter must be specified.

```
Set-MailboxRestoreRequest -Identity "Kweku\MailboxRestore"
-BadItemLimit 100 -AcceptLargeDataLoss
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox restore request" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxRestoreRequestIdParameter	The *Identity* parameter specifies the identity of the restore request. The *Identity* parameter consists of the alias of the mailbox to be restored and the name that was specified when the restore request was created. The identity of the restore request uses the following syntax: <*alias*>\<*name*> If you didn't specify a name for the restore request when it was created, Exchange automatically generated the default name MailboxRestore. Exchange generates up to 10 names, starting with MailboxRestore and then MailboxRestore*X* (where *X* = 1–9).
RehomeRequest	Required	System.Management.Automation.SwitchPar	The *RehomeRequest* switch specifies that the

		ameter	mailbox restore request be moved to a different mailbox database. Use this parameter to edit a mailbox restore request in the case where the source mailbox database from the original move request has to be removed.
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51 or higher. Items are considered corrupted if the item can't be read from the source database or can't be written to the target database. Corrupted items won't be available in the destination mailbox or .pst file.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use 0 to not skip bad items. The valid input range for this

			parameter is from 0 through 2147483647. The default value is 0. We recommend that you keep the default value 0 and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the *BadItemLimit* parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process is complete.
BatchName	Optional	System.String	The *BatchName* parameter specifies the name of the batch. Use this parameter to change, create, or remove a batch name. To remove a batch name, set the *BatchName* parameter value to an

			empty string or to null, for example, `-BatchName ""` or `-BatchName $null`.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the status of a completed restore request is set to `Completed`. If this parameter is set to a value of `0`, the status is cleared immediately instead of changing it to `Completed`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of items to skip in the mailbox because these items exceed the item size limit in the target mailbox data. Use `0` to not skip any large items. **Note:** If you set the *LargeItemLimit* parameter to 51 or higher, you must also include the *AcceptLargeDataLoss* parameter.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	The *Priority* parameter specifies the priority of the mailbox restore request. Use one of the following values: • `Emergency` • `Highest` • `Higher` • `High` • `Normal` • `Low` • `Lower` • `Lowest`
RemoteCredential	Optional	System.Management.Automation.PSCredential	This parameter is reserved for internal Microsoft use.

RemoteHostName	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is reserved for internal Microsoft use.
SkipMerging	Optional	Microsoft.Exchange.Management.RecipientTasks.SkippableMergeComponent[]	The *SkipMerging* parameter specifies folder-related items to skip when restoring the mailbox. Use one of the following values: • `FolderRules` • `FolderACLs` • `InitialConnectionValidation` Use this parameter only if a restore request fails because of folder rules, folder access control lists (ACLs), or initial connection validation.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Suspend-MailboxRestoreRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Suspend-MailboxRestoreRequest** cmdlet to suspend a restore request any time after the request was created, but before the request reaches the status of Completed. You can resume the restore request by using the Resume-MailboxRestoreRequest cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Suspend-MailboxRestoreRequest -Identity <MailboxRestoreRequestIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-SuspendComment
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example suspends the second restore request for Ayla's mailbox with the identity Ayla \MailboxRestore1.

```
Suspend-MailboxRestoreRequest -Identity "Ayla
\MailboxRestore1"
```

EXAMPLE 2

This example suspends all restore requests that are in progress by using the Get-MailboxRestoreRequest cmdlet to retrieve all requests with a status of `InProgress`, and then pipelines the output to the **Suspend-MailboxRestoreRequest** cmdlet with the suspend comment "Resume after 10:00 PM."

```
Get-MailboxRestoreRequest -Status InProgress | Suspend-
```

```
MailboxRestoreRequest -SuspendComment "Resume after 10:00
PM"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox restore request" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxRestoreRequestIdParameter	The *Identity* parameter specifies the identity of the restore request. The *Identity* parameter consists of the alias of the mailbox to be restored and the name that was specified when the restore request was created. The identity of the restore request uses the following syntax: *<alias>**<name>*. If you didn't specify a name for the restore request when it was created, Exchange automatically generated the default name MailboxRestore. Exchange generates up

			to 10 names, starting with MailboxRestore and then MailboxRestore*X* (where *X* = 1–9).
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description about why the request was suspended.
WhatIf	Optional	System.Management.Automation.SwitchPara	The *WhatIf* switch instructs the command

		meter	to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxRestoreRequestStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxRestoreRequestStatistics** cmdlet to view detailed information about restore requests.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxRestoreRequestStatistics -Identity
<MailboxRestoreRequestIdParameter> <COMMON PARAMETERS>

Get-MailboxRestoreRequestStatistics -RequestQueue <DatabaseIdParameter> [-
```

```
RequestGuid <Guid>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-Diagnostic <SwitchParameter>] [-DomainController
<Fqdn>] [-IncludeReport <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns the default statistics for the restore request with the identity Tony \MailboxRestore1. The type of information returned by default includes name, mailbox, status, and percent complete.

```
Get-MailboxRestoreRequestStatistics -Identity "Tony
\MailboxRestore1"
```

EXAMPLE 2

This example returns the statistics for Tony Smith's mailbox and exports the report to a CSV file.

```
Get-MailboxRestoreRequestStatistics -Identity Tony
\MailboxRestore | Export-CSV \\SERVER01
\RestoreRequest_Reports\Tony_Restorestats.csv
```

EXAMPLE 3

This example returns additional information about the restore request for Tony Smith's mailbox by using the *IncludeReport* parameter and by pipelining the results to the **Format-List** command.

```
Get-MailboxRestoreRequestStatistics -Identity Tony
\MailboxRestore -IncludeReport | Format-List
```

EXAMPLE 4

This example returns default statistics for a restore request being processed by the instance of MRS running on the server CAS01. This command only returns information for restore requests currently being processed by an instance of MRS. If the Client Access server is finished processing all restore requests, no information is returned. This command is for debugging purposes only and should only be performed if requested by Microsoft Customer Service and Support.

```
Get-MailboxRestoreRequestStatistics -MRSInstance
CAS01.contoso.com
```

EXAMPLE 5

This example returns additional information for all the restore requests that have a status of `Failed`

by using the *IncludeReport* parameter, and then saves the information to the text file AllRestoreReports.txt in the location where the command is being run.

```
Get-MailboxRestoreRequest -Status Failed | Get-MailboxRestoreRequestStatistics -IncludeReport | Format-List > AllRestoreReports.txt
```

Detailed Description

The *RequestQueue* and *MRSInstance* parameter syntax sets are for debugging purposes only and should only be used when directed by Microsoft Customer Service and Support.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox restore request" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MailboxRestoreRequestIdParameter	The *Identity* parameter specifies the identity of the restore request. The *Identity* parameter consists of the alias of the mailbox to be restored and the name that was specified when the restore request was created. The identity of the restore request uses the following syntax: <*alias*>\<*name*>. If you didn't specify a name for the restore request when it was created, Exchange automatically generated

			the default name MailboxRestore. Exchange generates up to 10 names, starting with MailboxRestore and then MailboxRestore*X* (where *X* = 1–9). This parameter can't be used in conjunction with the *MRSInstance* or *RequestQueue* parameters.
RequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies the target mailbox database on which the mailbox or archive of the request resides. You can use one of the following values: • GUID of the database • Database name This parameter can't be used in conjunction with the *Identity* or *MRSInstance* parameters.
Diagnostic	Optional	System.Management.Automation.SwitchParameter	The *Diagnostic* switch specifies whether to retrieve extremely detailed information about the mailbox restore request.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
IncludeReport	Optional	System.Management.Automation.SwitchParameter	The *IncludeReport* switch specifies whether to return additional details, which can be used for troubleshooting.
RequestGuid	Optional	System.Guid	The *RequestGuid* parameter specifies the unique identifier for the restore request. To find the GUID, use the Get-MailboxRestoreRequest cmdlet.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxStatistics

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-23*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxStatistics** cmdlet to obtain information about a mailbox, such as the size of the mailbox, the number of messages it contains, and the last time it was accessed. In addition, you can get the move history or a move report of a completed move request.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxStatistics -Identity <GeneralMailboxOrMailUserIdParameter> [-
Archive <SwitchParameter>] [-CopyOnServer <ServerIdParameter>] <COMMON
PARAMETERS>


Get-MailboxStatistics -Database <DatabaseIdParameter> [-CopyOnServer
<ServerIdParameter>] [-Filter <String>] [-StoreMailboxIdentity
<StoreMailboxIdParameter>] <COMMON PARAMETERS>


Get-MailboxStatistics -Server <ServerIdParameter> [-Filter <String>] [-
IncludePassive <SwitchParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-IncludeMoveHistory
<SwitchParameter>] [-IncludeMoveReport <SwitchParameter>] [-NoADLookup
<SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves the mailbox statistics for the mailbox of the user Ayla Kol by using its associated alias AylaKol.

```
Get-MailboxStatistics -Identity AylaKol
```

EXAMPLE 2

This example retrieves the mailbox statistics for all mailboxes on the server MailboxServer01.

```
Get-MailboxStatistics -Server MailboxServer01
```

EXAMPLE 3

This example retrieves the mailbox statistics for the specified mailbox.

```
Get-MailboxStatistics -Identity contoso\chris
```

EXAMPLE 4

This example retrieves the mailbox statistics for all mailboxes in the specified mailbox database.

```
Get-MailboxStatistics -Database "Mailbox Database"
```

EXAMPLE 5

This example retrieves the mailbox statistics for the disconnected mailboxes for all mailbox databases in the organization. The **-ne** operator means not equal.

```
Get-MailboxDatabase | Get-MailboxStatistics -Filter
'DisconnectDate -ne $null'
```

EXAMPLE 6

This example retrieves the mailbox statistics for a single disconnected mailbox. The value for the *StoreMailboxIdentity* parameter is the mailbox GUID of the disconnected mailbox. You can also use the LegacyDN.

```
Get-MailboxStatistics -Database "Mailbox Database" -
StoreMailboxIdentity 3b475034-303d-49b2-9403-ae022b43742d
```

EXAMPLE 7

This example returns the summary move history for the completed move request for Ayla Kol's mailbox. If you don't pipeline the output to the **Format-List** cmdlet, the move history doesn't display.

```
Get-MailboxStatistics -Identity AylaKol -IncludeMoveHistory
| Format-List
```

EXAMPLE 8

This example returns the detailed move history for the completed move request for Ayla Kol's mailbox. This example uses a temporary variable to store the mailbox statistics object. If the mailbox has been moved multiple times, there are multiple move reports. The last move report is always `MoveReport[0]`.

```
$temp=Get-MailboxStatistics -Identity AylaKol -
IncludeMoveHistory
$temp.MoveHistory[0]
```

EXAMPLE 9

This example returns the detailed move history and a verbose detailed move report for Ayla Kol's mailbox. This example uses a temporary variable to store the move request statistics object and outputs the move report to a CSV file.

```
$temp=Get-MailboxStatistics -Identity AylaKol -
IncludeMoveReport
$temp.MoveHistory[0] | Export-CSV C:\MoveReport_AylaKol.csv
```

Detailed Description

On Mailbox servers only, you can use the **Get-MailboxStatistics** cmdlet without parameters. In this case, the cmdlet returns the statistics for all mailboxes on all databases on the local server.

Note:

The **Get-MailboxStatistics** cmdlet requires at least one of the following parameters to complete successfully: *Server*, *Database*, or *Identity*.

You can use the **Get-MailboxStatistics** cmdlet to return detailed move history and a move report for completed move requests to troubleshoot a move request. To view the move history, you must pass this cmdlet as an object. Move histories are retained in the mailbox database and are numbered incrementally, and the last executed move request is always numbered 0. For more information, see "EXAMPLE 7," "EXAMPLE 8," and "EXAMPLE 9" in this topic.

Note:

You can only see move reports and move history for completed move requests.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Database	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *Database* parameter specifies the name of the mailbox database. When you specify a value for the *Database* parameter, the Exchange Management Shell

			returns statistics for all the mailboxes on the database specified. You can use the following values: • GUID • Database This parameter accepts pipeline input from the **Get-MailboxDatabase** cmdlet.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.GeneralMailboxOrMailUserIdParameter	The *Identity* parameter specifies a mailbox. When you specify a value for the *Identity* parameter, the command looks up the mailbox specified in the *Identity* parameter, connects to the server where the mailbox resides, and returns the statistics for the mailbox. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)`

			Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	This parameter is available only in on-premises Exchange 2013. The *Server* parameter specifies the server from which you want to obtain mailbox statistics. You can use one of the following values: • Fully qualified domain name (FQDN)

			• NetBIOS name When you specify a value for the *Server* parameter, the command returns statistics for all the mailboxes on all the databases, including recovery databases, on the specified server. If you don't specify this parameter, the command returns logon statistics for the local server.
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* switch parameter specifies whether to return mailbox statistics for the archive mailbox associated with the specified mailbox. You don't have to specify a value with this parameter.
CopyOnServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	This parameter is available only in on-premises Exchange 2013. The *CopyOnServer* parameter is used to retrieve statistics from a specific database copy on the server specified with the *Server* parameter.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *Filter* parameter specifies a filter to filter the results of the **Get-MailboxStatistics** cmdlet. For example, to display all disconnected mailboxes on a specific mailbox database, use the following syntax for this parameter: `-Filter 'DisconnectDate -ne $null'`
IncludeMoveHistory	Optional	System.Management.Automation.SwitchParameter	The *IncludeMoveHistory* switch specifies whether to return additional information about the mailbox that includes the history of a completed move request, such as

			status, flags, target database, bad items, start times, end times, duration that the move request was in various stages, and failure codes.
IncludeMoveReport	Optional	System.Management.Automation.SwitchParameter	The *IncludeMoveReport* switch specifies whether to return a verbose detailed move report for a completed move request, such as server connections and move stages. **Note:** Because the output of this command is verbose, you should send the output to a .CSV file for easier analysis.
IncludePassive	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. Without the *IncludePassive* parameter, the cmdlet retrieves statistics from active database copies only. Using the *IncludePassive* parameter, you can have the cmdlet return statistics from all active and passive database copies.

NoADLookup	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *NoADLookup* switch specifies that information is retrieved from the mailbox database, and not from Active Directory. This helps improve cmdlet performance when querying a mailbox database that contains a large number of mailboxes.
StoreMailboxIdentity	Optional	Microsoft.Exchange.Configuration.Tasks.StoreMailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *StoreMailboxIdentity* parameter specifies the mailbox identity when used with the *Database* parameter to return statistics for a single mailbox on the specified database. You can use one of the following values: • MailboxGuid • LegacyDN Use this syntax to retrieve information about disconnected mailboxes, which don't have a

			corresponding Active Directory object or that has a corresponding Active Directory object that doesn't point to the disconnected mailbox in the mailbox database.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailMessage

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MailMessage** cmdlet to create an email message for the specified user mailbox and place the email message in the Drafts folder of the user's mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MailMessage [-Body <String>] [-BodyFormat <PlainText | Html | Rtf>] [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Subject
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates an email message in the Drafts folder with the subject and body specified by the *Subject* and *Body* parameters. The message body is rendered in plain text because no format for the message body is specified.

```
New-MailMessage -Subject "Delivery Report" -Body "Click
here to view this report"
```

EXAMPLE 2

This example creates an empty email message in the Drafts folder because no subject or message body is specified.

```
New-MailMessage
```

EXAMPLE 3

This example creates an email message in the Drafts folder with the subject and body specified by the *Subject* and *Body* parameters. The message body is rendered in HTML format.

```
New-MailMessage -Subject "Delivery Information" -Body
"Click here to see details" -BodyFormat Html
```

Detailed Description

If the cmdlet is run without specifying the *Subject* or *Body* parameters, an empty email message is placed in the user's Drafts folder.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "User mailboxes" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Body	Optional	System.String	The *Body* parameter specifies the content of the body section of the new email message.
BodyFormat	Optional	Microsoft.Exchange.Data.Providers.MailBodyFo	The *BodyFormat* parameter specifies the

		rmat	format of the message body. The values can be `PlainText`, `Rtf` (Rich Text Format), or `Html`. By default, if the *BodyFormat* parameter isn't specified when the *Body* parameter is used, the message body is rendered in plain text.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this

			configuration change to Active Directory.
Subject	Optional	System.String	The *Subject* parameter specifies the content of the subject field of the new email message.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MessageCategory

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MessageCategory** cmdlet to retrieve a message category from the specified mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MessageCategory [-Identity <MessageCategoryIdParameter>] [-
DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves message categories from the mailbox User1.

```
Get-MessageCategory -Mailbox "User1"
```

Detailed Description

The **Get-MessageCategory** cmdlet is used by the web management interface in Microsoft Exchange to populate fields that display message category information.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Message categories" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MessageCategoryIdParameter	The *Identity* parameter specifies the name of the message category to be retrieved.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of a mailbox user to retrieve the message category from.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-RecipientDataProperty

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Export-RecipientDataProperty** cmdlet to download a user's picture or spoken name sound file. The picture and audio files display in the **Global Address List** property dialog box, contact card, reading pane, and meeting requests in Microsoft Outlook and Microsoft Office

Outlook Web App.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Export-RecipientDataProperty [-Picture <SwitchParameter>] <COMMON
PARAMETERS>


Export-RecipientDataProperty [-SpokenName <SwitchParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: -Identity <MailboxUserContactIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example exports Tony Smith's spoken name audio file and saves it to the local computer.

```
Export-RecipientDataProperty -Identity tony@contoso.com -
SpokenName | ForEach { $_.FileData | Add-Content C:
\tonysmith.wma -Encoding Byte}
```

EXAMPLE 2

This example exports Ayla Kol's picture file to the local computer.

```
Export-RecipientDataProperty -Identity "Ayla" -Picture |
ForEach { $_.FileData | Add-Content C:\aylakol.jpg -
Encoding Byte}
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient data properties" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxUserContactIdPara	The *Identity* parameter specifies the mailbox or mail contact from which

		meter	you want to export the recipient data. You can use the following values: • Distinguished name (DN) • Canonical name • GUID • Name • Display name • Alias • Exchange DN • Primary SMTP email address
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Picture	Optional	System.Management.	The *Picture* switch

		Automation.SwitchParameter	specifies that the file you're exporting is the user's picture file. You can't use this parameter in conjunction with the *SpokenName* switch. You can only export one file type at a time.
SpokenName	Optional	System.Management.Automation.SwitchParameter	The *SpokenName* switch specifies that the file you're exporting is the user's audio file. This cmdlet exports the WMA 9-voice format. You can't use this parameter in conjunction with the *Picture* switch. You can only export one file type at a time.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Import-RecipientDataProperty

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Import-RecipientDataProperty** cmdlet to add a picture or an audio file of a spoken name to a mailbox or contact. The picture and audio files display on the **Global Address List** property dialog box, contact card, reading pane, and meeting requests in Microsoft Outlook and Microsoft Office Outlook Web App.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Import-RecipientDataProperty [-Picture <SwitchParameter>] <COMMON
PARAMETERS>


Import-RecipientDataProperty [-SpokenName <SwitchParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: -FileData <Byte[]> -Identity
<MailboxUserContactIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example imports the audio file for Tony Smith's spoken name.

```
Import-RecipientDataProperty -Identity "Tony Smith" -
SpokenName -FileData ([Byte[]]$(Get-Content -Path "M:
```

```
\AudioFiles\TonySmith.wma" -Encoding Byte -ReadCount 0))
```

EXAMPLE 2

This example imports the picture file for Ayla Kol.

```
Import-RecipientDataProperty -Identity Ayla -Picture -
FileData ([Byte[]]$(Get-Content -Path "M:\Employee Photos
\AylaKol.jpg" -Encoding Byte -ReadCount 0))
```

Detailed Description

Importing and exporting files require a specific syntax because importing and exporting use Remote PowerShell.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient data properties" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
FileData	Required	System.Byte[]	The *FileData* parameter specifies the location and file name of the picture or audio file.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxUserContactIdParameter	The *Identity* parameter specifies the mailbox or contact that you're adding the picture or spoken name file to. You can use one of the following values: • Alias • Canonical name • Display name • Distinguished name

			(DN) • Exchange DN • GUID • Name • Primary SMTP email address
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Picture	Optional	System.Management.Automation.SwitchParameter	The *Picture* switch specifies that the file you're importing is a picture file. The picture must be a JPEG file and shouldn't be larger than 10 kilobytes (KB). You can't use this parameter in

			conjunction with the *SpokenName* switch. You can only import one file type at a time.
SpokenName	Optional	System.Management.Automation.SwitchParameter	The *SpokenName* switch specifies that the file you're importing is an audio file. The maximum file size should be less than 32 KB. You can use one of the following formats: • WMA 9-voice • PCM 8-KHz, 16-bits, mono format You can't use this switch in conjunction with the *Picture* switch. You can only import one file type at a time.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ResourceConfig

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ResourceConfig** cmdlet to get resource property schema data from Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ResourceConfig [-Identity <OrganizationIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example gets the resource property schema data from Active Directory, where test.Contoso.com is the domain controller for that site.

```
Get-ResourceConfig -DomainController test.Contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Resource mailbox schema configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter is reserved for internal use only.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ResourceConfig

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ResourceConfig** cmdlet to set resource property schema and resource locations on the **Resource Config** object in Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ResourceConfig [-Identity <OrganizationIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ResourcePropertySchema
<MultiValuedProperty>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds equipment and room resources to the custom **Resource Config** list in Active Directory.

Note:

All entries must start with either `Room/` or `Equipment/`. Setting a new entry using the **Set-ResourceConfig** cmdlet overwrites all existing entries; it doesn't add a new entry to the list. Use the **Get-ResourceConfig** cmdlet to query the existing entries, and then append to the list. To add or remove entries without overwriting the current entries, see "EXAMPLE 2" and "EXAMPLE 3" in the Examples section.

```
Set-ResourceConfig -DomainController server1.contoso.com -
ResourcePropertySchema ("Room/16Seats","Equipment/
Projector","Room/8Seats","Equipment/Whiteboard")
```

EXAMPLE 2

This example creates three custom resource properties for room mailboxes and two custom resource properties for equipment mailboxes. This example also demonstrates two methods for adding new custom resource properties. The syntax of both commands can be used interchangeably. Perform the following steps:

1. Read the current resource configuration and store it in a temporary variable called `$ResourceConfiguration`.

```
$ResourceConfiguration = Get-ResourceConfig
```

2. Create the custom properties AV, TV, and Whiteboard for room mailboxes.

Note:

This example assumes that some of the meeting rooms in your organization have audio-visual equipment, TV, or whiteboards. It also assumes that you want to differentiate the rooms that have the specified features from others that don't have these features.

```
$ResourceConfiguration.ResourcePropertySchema+=("Room/AV")
$ResourceConfiguration.ResourcePropertySchema.Add("Room/
TV")
$ResourceConfiguration.ResourcePropertySchema+=("Room/
Whiteboard")
```

3. Create the custom properties Car and Van for equipment mailboxes by running the following

commands.

Note:

This example assumes that your organization uses equipment mailboxes to track the scheduling of company vehicles, and you plan to use the custom resource properties to specify the vehicle type.

```
$ResourceConfiguration.ResourcePropertySchema.Add("Equipmen
t/Car")
$ResourceConfiguration.ResourcePropertySchema+=("Equipment/
Van")
```

4. Update the resource configuration of your organization by using the modified resource property schema.

```
Set-ResourceConfig -ResourcePropertySchema
$ResourceConfiguration.ResourcePropertySchema
```

EXAMPLE 3

This example removes two of the custom resource properties for room mailboxes that were created in the previous example. The commands also demonstrate two methods for removing a custom resource property. The syntax of both commands can be used interchangeably.

1. Read the current resource configuration and store it in a temporary variable called `$ResourceConfiguration`.

```
$ResourceConfiguration = Get-ResourceConfig
```

2. Remove the custom properties AV and TV for room mailboxes.

```
$ResourceConfiguration.ResourcePropertySchema-=("Room/AV")
$ResourceConfiguration.ResourcePropertySchema.Remove("Room/
TV")
```

3. Update the resource configuration of your organization by using the modified resource property schema.

```
Set-ResourceConfig -ResourcePropertySchema
$ResourceConfiguration.ResourcePropertySchema
```

Detailed Description

Custom resource properties are features for room or equipment mailboxes. Administrators can indicate that a resource has a specific feature by assigning the corresponding custom resource property to that resource mailbox.

Important:
Custom resource properties can't include spaces.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Resource mailbox schema configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
ResourcePropertySche	Optional	Microsoft.Exchange.Dat	The

ma		a.MultiValuedProperty	*ResourcePropertySchema* parameter specifies a list of custom strings that you can use to tag resources.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-ServiceEmailChannel

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-ServiceEmailChannel** cmdlet to disable the .NET service channel for a specific user. The .NET service channel enables Microsoft Exchange to store information that it later forwards to applications or devices that aren't permanently connected to the server running Exchange.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-ServiceEmailChannel -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the .NET service channel for the user Jeff Hay.

```
Disable-ServiceEmailChannel -Identity JeffHay
```

EXAMPLE 2

This example disables the .NET service channel for the user Jeff Hay after confirmation is given.

```
Disable-ServiceEmailChannel -Identity JeffHay -Confirm
$true
```

EXAMPLE 3

This example disables the .NET service channel for the user Jeff Hay without requiring confirmation.

```
Disable-ServiceEmailChannel -Identity JeffHay -Confirm
$false
```

Detailed Description

The **Disable-ServiceEmailChannel** cmdlet deletes the receive folder in the user's mailbox under the root folder.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access service email channel settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identity of the user for which you want to enable the .NET service channel. The user specified must be a valid user in Active Directory who has an Exchange mailbox.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.A	The *WhatIf* switch

		utomation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-ServiceEmailChannel

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-ServiceEmailChannel** cmdlet to enable the .NET service channel for a specific user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-ServiceEmailChannel -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the .NET service channel for the user Tony Smith.

```
Enable-ServiceEmailChannel -Identity "fourthcoffee\tony"
```

EXAMPLE 2

This example enables the .NET service channel for the user Tony Smith.

```
Enable-ServiceEmailChannel -Identity "tony@contoso.com"
```

EXAMPLE 3

This example enables the .NET service channel for the user Tony Smith.

```
Enable-ServiceEmailChannel -Identity "TonySmith"
```

Detailed Description

The .NET service channel enables Microsoft Exchange to store information that it later forwards to applications or devices that aren't permanently connected to the server running Exchange. This cmdlet creates a receive folder in the user's mailbox under the root folder named Service E-mail.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access service email channel settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identity of the user for which you want to enable the .NET service channel. The user specified must be a valid user in Active

			Directory who has an Exchange mailbox.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UserPhoto

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-UserPhoto** cmdlet to view information about the user photos feature that allows users to associate a picture with their account. User photos are stored in Active Directory and appear in on-premises and cloud-based client applications, such as Microsoft Office Outlook Web App, Microsoft Office Lync, and Microsoft SharePoint.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UserPhoto [-Anr <String>] <COMMON PARAMETERS>

Get-UserPhoto [-Identity <MailboxIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-Filter <String>]
[-IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-Preview <SwitchParameter>] [-
ReadFromDomainController <SwitchParameter>] [-ResultSize <Unlimited>] [-
SortBy <String>]
```

Examples

EXAMPLE 1

This example displays information about the user photo configured for Susan Burk.

```
Get-UserPhoto "Susan Burk"
```

EXAMPLE 2

This example displays information about the user photo that was uploaded to Pilar Pinilla's account, but wasn't saved.

```
Get-UserPhoto "Pilar Pinilla" -Preview
```

Detailed Description

The user photos feature allows users to associate a picture with their account. User photos are stored in the user's Active Directory account and in the root directory of the user's Exchange mailbox. The user photo feature must be set for a user before you can run the **Get-UserPhoto** cmdlet to view information about the user's photo. Otherwise, you get an error message saying the user photo doesn't exist for the specified users. Administrators use the **Set-UserPhoto** cmdlet or the Exchange Administration Center to configure user photos. Users can upload, preview, and save a user photo to their account by using the Outlook Web App Options page.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient provisioning permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous

			name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the account used to read Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active

			Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identity of the user. You can use one of the following values: • GUID • Distinguished name (DN) • Display name • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to

			access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParam	The *OrganizationalUnit* parameter returns objects only from the specified

		eter	organizational unit (OU).
Preview	Optional	System.Management.Automation.SwitchParameter	The *Preview* parameter retrieves information about the preview photo for the user account. The preview photo is the photo object that was uploaded to the user's account, but wasn't saved, for example, if a user uploads a photo in Outlook Web App Options, but doesn't save it. If you use the *Preview* parameter after a user photo is saved, this cmdlet returns an error saying the preview photo doesn't exist.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest, and if you don't use this parameter, it's possible that the user information is read from a global

			catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers running Exchange.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute to sort by. This parameter sorts by a single attribute in ascending order.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-UserPhoto

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-UserPhoto** cmdlet to delete the photo associated with a user's account. The user photo feature allows users to associate a picture with their account. User photos appear in on-premises and cloud-based client applications, such as Microsoft Office Outlook Web App, Microsoft Office Lync, and Microsoft SharePoint. This cmdlet deletes the photo associated with a user's account.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-UserPhoto -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreDefaultScope
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the photo associated with Ann Beebe's user account.

```
Remove-UserPhoto "Ann Beebe"
```

Detailed Description

Use the **Remove-UserPhoto** cmdlet to delete the user photo currently associated with a user's account. User photos are stored in the user's Active Directory account and in the root directory of the user's Exchange mailbox, both of which are deleted when you run this cmdlet. Administrators can also use the Exchange Administration Center to delete user photos by accessing the user's Outlook Web App Options page.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient provisioning permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identity of the user. You can use one of the following values:

			• GUID • Distinguished name (DN) • Display name • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.	The *IgnoreDefaultScope*

		Automation.SwitchParameter	parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-UserPhoto

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-UserPhoto** cmdlet to configure the user photos feature that allows users to associate a picture with their account. User photos are stored in Active Directory and appear in on-premises and cloud-based client applications, such as Microsoft Office Outlook Web App, Microsoft Office Lync, and Microsoft SharePoint.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-UserPhoto -Identity <MailboxIdParameter> -PictureData <Byte[]> <COMMON
PARAMETERS>

Set-UserPhoto -Cancel <SwitchParameter> -Identity <MailboxIdParameter>
<COMMON PARAMETERS>

Set-UserPhoto -Identity <MailboxIdParameter> -Preview <SwitchParameter> [-
```

```
PictureData <Byte[]>] [-PictureStream <Stream>] <COMMON PARAMETERS>

Set-UserPhoto -Identity <MailboxIdParameter> -PictureStream <Stream>
<COMMON PARAMETERS>

Set-UserPhoto -Identity <MailboxIdParameter> -Save <SwitchParameter>
<COMMON PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-IgnoreDefaultScope <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example uploads and saves a photo to Paul Cannon's user account using a single command.

```
Set-UserPhoto "Paul Cannon" -PictureData
([System.IO.File]::ReadAllBytes("C:\Users\Administrator
\Desktop\PaulCannon.jpg"))
```

EXAMPLE 2

This example shows how to use two commands to upload and save a preview photo to Ann Beebe's user account.

This command uploads a preview photo to Ann Beebe's user account.

```
Set-UserPhoto "Ann Beebe" -PictureData
([System.IO.File]::ReadAllBytes("C:\Users\Administrator
\Desktop\AnnBeebe.jpg")) -Preview
```

To save the preview photo that was uploaded using the previous command, run the following command.

```
Set-UserPhoto "Ann Beebe" -Save
```

To delete the preview photo that was uploaded using the first command in this example, run the following command.

```
Set-UserPhoto "Ann Beebe" -Cancel
```

Detailed Description

The user photos feature allows users to associate a picture with their account. User photos are stored in the user's Active Directory account and in the root directory of the user's Exchange

mailbox. Administrators use the **Set-UserPhoto** cmdlet or the Exchange Administration Center (EAC) to configure user photos. Users can upload, preview, and save a user photo to their account by using the Outlook Web App Options page. When a user uploads a photo, a preview of the photo is displayed on the Outlook Web App Options page. This is the preview state, and creates the same result as running the **Set-UserPhoto** cmdlet using the *Preview* parameter. If the user clicks **Save**, the preview photo is saved as the user's photo. This is the same result as running the `Set-UserPhoto -Save` command or running both the `Set-UserPhoto -Preview` and `Set-UserPhoto -Save` commands. If the user cancels the preview photo on the Outlook Web App Options page, then the `Set-UserPhoto -Cancel` command is called.

A user photo must be set for a user before you can run the **Get-UserPhoto** cmdlet to view information about the user's photo. Otherwise, you'll get an error message saying the user photo doesn't exist for the specified user. Alternatively, you can run the `Get-UserPhoto -Preview` command to view information about a preview photo.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient provisioning permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Cancel	Required	System.Management.Automation.SwitchParameter	The *Cancel* parameter deletes the photo currently uploaded as the preview photo. To delete the photo currently associated with a user's account, use the **Remove-UserPhoto** command. The *Cancel* parameter only deletes the preview photo.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identity of the user. You can use one of the following values:

			• GUID • Distinguished name (DN) • Display name • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
PictureData	Required	System.Byte[]	The *PictureData* parameter specifies the photo file that will be uploaded to the user's account. Use the following syntax for this parameter. `([System.IO.File]::ReadAllBytes("<`**file name and path**`>"))`. The following is an example. `([System.IO.File]::ReadAllBytes("C:\Documents\Pictures\MyPhoto.jpg"))`.
PictureStream	Required	System.IO.Stream	The *PictureStream* parameter specifies the photo that will be uploaded to the user's account. This parameter is used by client applications such as Outlook Web App when users add a photo. To upload a photo using Windows PowerShell, use

			the *PictureData* parameter to specify the photo file.
Preview	Required	System.Management.Automation.SwitchParameter	The *Preview* parameter uploads a preview photo for the user account. A preview photo is the photo object that is uploaded to the user's account, but isn't saved. For example, if a user uploads a photo in Outlook Web App Options to preview before saving it. If you use the *Preview* parameter to upload a preview photo, you have to run the `Set-UserPhoto -Save` command to save it as the user's photo.
Save	Required	System.Management.Automation.SwitchParameter	The *Save* parameter specifies that the photo that's uploaded to the user's account will be saved as the user's photo.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before

			processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the

			DomainController parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Mailbox database cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-09*

Mailbox database cmdlets

Dismount-Database

Mount-Database

Move-DatabasePath

Update-DatabaseSchema

Get-MailboxDatabase

New-MailboxDatabase

Remove-MailboxDatabase

Set-MailboxDatabase

Disable-MailboxQuarantine

Enable-MailboxQuarantine

Get-MailboxRepairRequest

New-MailboxRepairRequest

Remove-MailboxRepairRequest

Test-MapiConnectivity

Remove-StoreMailbox

Update-StoreMailboxState

Get-StoreUsageStatistics

Exchange search cmdlets

Test-ExchangeSearch

Get-FailedContentIndexDocuments

Get-SearchDocumentFormat

New-SearchDocumentFormat

Remove-SearchDocumentFormat

Set-SearchDocumentFormat

Dismount-Database

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Dismount-Database** cmdlet to dismount a database on a computer running Microsoft Exchange Server 2013 that has the Mailbox server role installed. You can run this command only if the Microsoft Exchange Information Store service is running.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Dismount-Database -Identity <DatabaseIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example dismounts the database `MyDatabase`.

```
Dismount-Database -Identity MyDatabase
```

Note:

Regardless of where you run this cmdlet, it operates against the server hosting the active copy of the database.

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox databases" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Identity* parameter specifies the GUID or distinguished name (DN) that represents a specific database.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes

			would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Mount-Database

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Mount-Database** cmdlet to mount a database on a server running Microsoft Exchange Server 2013 with the Mailbox server role installed. The cmdlet mounts the database only if the Microsoft Exchange Information Store service and Microsoft Exchange Replication service are running.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Mount-Database -Identity <DatabaseIdParameter> [-AcceptDataLoss
<SwitchParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Force <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example mounts the database MyDatabase.

```
Mount-Database -Identity ExchangeServer1.Contoso.com
\MyDatabase
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox databases" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Identity* parameter specifies the GUID or distinguished name (DN) that represents a specific database. The *Identity* parameter label is optional.
AcceptDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptDataLoss* parameter specifies that the command accepts the data loss caused by missing committed transaction log files without asking for user confirmation.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* parameter specifies a forced mount of an empty database. The parameter also overrides any errors or warnings during the database mount.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of

			those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Move-DatabasePath

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Move-DatabasePath** cmdlet to set a new path to the location of a database on the specified Mailbox server and to move the related files to that location.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Move-DatabasePath -Identity <DatabaseIdParameter> [-ConfigurationOnly
<SwitchParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-EdbFilePath <EdbFilePath>] [-Force <SwitchParameter>] [-
LogFolderPath <NonRootLocalLongFullPath>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets a new path for the mailbox database specified by the mailbox database name. To perform the move operation, the database must be temporarily dismounted, making it inaccessible to all users. If the database is currently dismounted, it isn't remounted upon completion.

```
Move-DatabasePath -Identity MyDatabase01 -EdbFilePath C:
\NewFolder\MyDatabase01.edb
```

Detailed Description

When you use the **Move-DatabasePath** cmdlet, consider the following:

- This cmdlet fails if it's run while the database is being backed up.
- If the specified database is mounted when this cmdlet is run, the database is automatically dismounted and then remounted, and is unavailable to users while it's dismounted.
- This cmdlet normally can be run on the affected Mailbox server only. An exception is that this cmdlet can be run on an administrator's workstation when using the *ConfigurationOnly* parameter with a value of `$true`.
- This cmdlet can't be run against replicated mailbox databases. To move the path of a replicated database, you must first remove all replicated copies, and then you can perform the move operation. After the move operation is complete, you can add copies of the mailbox database.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox databases" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or name of the database.
ConfigurationOnly	Optional	System.Management.Automation.SwitchParameter	The *ConfigurationOnly* switch specifies whether the configuration of the database changes without moving any files. A value of `$true` changes the

			configuration. A value of `$false` changes the configuration and moves the files. The default value is `$false`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EdbFilePath	Optional	Microsoft.Exchange.Data.EdbFilePath	The *EdbFilePath* parameter specifies a new file path for the database. All current database files are moved to this location. The default location is

			<ExchangeInstallDirectory>\Mailbox\LocalCopies\MBDatabase.edb. This file path can't be the same as the path for the backup copy of the database.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
LogFolderPath	Optional	Microsoft.Exchange.Data.NonRootLocalLongFullPath	The *LogFolderPath* parameter specifies the folder where log files are stored.
WhatIf	Optional	System.Management.A	The *WhatIf* switch

		utomation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-DatabaseSchema

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Update-DatabaseSchema** cmdlet to upgrade the database schema for one or more databases after an Exchange software update that includes database schema updates has been installed on Mailbox servers in a database availability group (DAG). Some software updates for Exchange may include database schema updates. After such an update has been installed on all members of a DAG, the administrator must run the **Update-DatabaseSchema** cmdlet for each

database in the DAG to trigger the database schema update. The in-place database schema upgrade engine ensures that no schema updates occur until all members of the DAG have compatible versions of the software.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-DatabaseSchema -Identity <DatabaseIdParameter> <COMMON PARAMETERS>

Update-DatabaseSchema -Identity <DatabaseIdParameter> -MajorVersion
<UInt16> -MinorVersion <UInt16> <COMMON PARAMETERS>

COMMON PARAMETERS: [-AllowFileRestore <$true | $false>] [-
AutoDagExcludeFromMonitoring <$true | $false>] [-AutoDatabaseMountDial
<Lossless | GoodAvailability | BestAvailability>] [-
BackgroundDatabaseMaintenance <$true | $false>] [-CircularLoggingEnabled
<$true | $false>] [-Confirm [<SwitchParameter>]] [-DatabaseGroup <String>]
[-DataMoveReplicationConstraint <None | SecondCopy | SecondDatacenter |
AllDatacenters | AllCopies | CINoReplication | CISecondCopy |
CISecondDatacenter | CIAllDatacenters | CIAllCopies>] [-
DeletedItemRetention <EnhancedTimeSpan>] [-DomainController <Fqdn>] [-
EventHistoryRetentionPeriod <EnhancedTimeSpan>] [-IssueWarningQuota
<Unlimited>] [-MaintenanceSchedule <Schedule>] [-MountAtStartup <$true |
$false>] [-Name <String>] [-QuotaNotificationSchedule <Schedule>] [-
RetainDeletedItemsUntilBackup <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example updates the database schema for database DB1.

```
Update-DatabaseSchema DB1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox databases" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Identity* parameter specifies the mailbox database for which you want to set one or more attributes. You can use the

			following values: • GUID • Distinguished name (DN) • Name of the mailbox database
MajorVersion	Required	System.UInt16	This parameter is reserved for internal Microsoft use.
MinorVersion	Required	System.UInt16	This parameter is reserved for internal Microsoft use.
AllowFileRestore	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
AutoDagExcludeFromMonitoring	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
AutoDatabaseMountDial	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AutoDatabaseMountDial	This parameter is reserved for internal Microsoft use.
BackgroundDatabaseMaintenance	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
CircularLoggingEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You

			don't have to specify a value with the *Confirm* switch.
DatabaseGroup	Optional	System.String	This parameter is reserved for internal Microsoft use.
DataMoveReplicationConstraint	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DataMoveReplicationConstraintParameter	This parameter is reserved for internal Microsoft use.
DeletedItemRetention	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EventHistoryRetentionPeriod	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
IssueWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
MaintenanceSchedule	Optional	Microsoft.Exchange.Data.Schedule	This parameter has been deprecated in Exchange 2013.
MountAtStartup	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.

Name	Optional	System.String	This parameter is reserved for internal Microsoft use.
QuotaNotificationSchedule	Optional	Microsoft.Exchange.Data.Schedule	This parameter has been deprecated in Exchange 2013.
RetainDeletedItemsUntilBackup	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-ExchangeSearch

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-ExchangeSearch** cmdlet to test that Exchange Search is currently enabled and is indexing new email messages in a timely manner.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-ExchangeSearch [-Archive <SwitchParameter>] [-Identity
<MailboxIdParameter>] <COMMON PARAMETERS>

Test-ExchangeSearch [-Server <ServerIdParameter>] <COMMON PARAMETERS>

Test-ExchangeSearch [-MailboxDatabase <DatabaseIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-IndexingTimeoutInSeconds <Int32>] [-MonitoringContext
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests Exchange Search results for the mailbox database on which the specified mailbox resides.

```
Test-ExchangeSearch -Identity john@contoso.com
```

EXAMPLE 2

This example tests Exchange Search against the mailbox database EXCH01-SG1-MDB1 with an indexing time-out of 30 seconds.

```
Test-ExchangeSearch -MailboxDatabase "EXCH01-SG1-MDB1" -
IndexingTimeoutInSeconds 30
```

EXAMPLE 3

This example tests Exchange Search results for the mailbox database on which the specified mailbox resides. The *Verbose* switch is used to display detailed information.

```
Test-ExchangeSearch -Identity john@contoso.com -Verbose
```

Detailed Description

The **Test-ExchangeSearch** cmdlet creates a hidden message and an attachment visible only to Exchange Search. Unless a mailbox is specified in the *Identity* parameter, the hidden message is stored in the System Attendant mailbox. The command waits for the message to be indexed and then searches for the content. It reports success or failure depending on whether the message is found after the interval set in the *IndexingTimeoutInSeconds* parameter has elapsed.

You can use the *Verbose* switch to get detailed information about each step performed by the cmdlet as part of the test.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Search" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* switch specifies that the test be run against the archive mailbox for the mailbox user specified in the *Identity* parameter. When the *Archive* switch is used, you must also use the *Identity* parameter to specify the mailbox.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value

			with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox that you want to test Exchange Search against. If this parameter isn't specified, the System Attendant mailbox is used. The *Identity* and *MailboxDatabase* parameters can't be used together.
IndexingTimeoutInSeconds	Optional	System.Int32	The *IndexingTimeoutInSeconds* parameter specifies, in seconds, the maximum amount of time to wait between adding the new email message to the test mailbox and waiting for it to be returned in a

			search result. The default value is 120 seconds. If this parameter isn't specified, the default interval is used.
MailboxDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *MailboxDatabase* parameter specifies the mailbox database to test Exchange Search against. The *MailboxDatabase* and *Identity* parameters can't be used together.
MonitoringContext	Optional	System.Management.Automation.SwitchParameter	The *MonitoringContext* switch includes the associated monitoring events and performance counters in the results. You don't need to specify a value with this switch. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations

			Manager.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server for the recipient that you want to test Exchange Search against.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-FailedContentIndexDocuments

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-FailedContentIndexDocuments** cmdlet to retrieve a list of documents for a mailbox, mailbox database, or Mailbox server that couldn't be indexed by Exchange Search.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-FailedContentIndexDocuments -Identity <MailboxIdParameter> [-Archive
<SwitchParameter>] <COMMON PARAMETERS>


Get-FailedContentIndexDocuments -Server <ServerIdParameter> <COMMON
PARAMETERS>


Get-FailedContentIndexDocuments -MailboxDatabase <DatabaseIdParameter>
<COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-Credential
<PSCredential>] [-DomainController <Fqdn>] [-EndDate <DateTime>] [-
ErrorCode <Int32>] [-FailureMode <Transient | Permanent | All>] [-
ReadFromDomainController <SwitchParameter>] [-ResultSize <Unlimited>] [-
StartDate <DateTime>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example retrieves a list of items that couldn't be indexed by Exchange Search from the mailbox of user Terry Adams.

```
Get-FailedContentIndexDocuments -Identity "Terry Adams"
```

EXAMPLE 2

This example retrieves a list of items that couldn't be indexed by Exchange Search from the mailbox database Mailbox Database MDB2.

```
Get-FailedContentIndexDocuments -MailboxDatabase "Mailbox
Database MDB2"
```

Detailed Description

The **Get-FailedContentIndexDocuments** cmdlet returns a list of documents that couldn't be indexed. The most common reason is that there was no filter available for that document type or

there was an attachment within the document. For example, the PDF filter isn't available by default. If an email message contains a PDF document, because there is no PDF filter, the document is marked as failed content indexing.

After a new filter is installed, only new messages with attachments of the type for which the filter is installed are indexed. If you want to index older messages for the document type, the mailbox has to be moved.

The cmdlet output provides details about items in a mailbox that couldn't be indexed, including an error code and the reason for failure.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Get unsearchable items" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox. You can use one of the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
MailboxDatabase	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *MailboxDatabase* parameter specifies the database from which to

			get the mailbox. You can use the following values: • GUID of the database • Database name • DN
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	This parameter is available only in on-premises Exchange 2013. The *Server* parameter specifies a Mailbox server. You can use the following values: • Name • GUID • DN
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* switch restricts the scope of the cmdlet to the user's archive. When using the *Archive* switch, you must also specify the *Identity* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

Credential	Optional	System.Management.Automation.PSCredential	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
EndDate	Optional	System.DateTime	The *EndDate* parameter specifies the end date of the date range. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must

			enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
ErrorCode	Optional	System.Int32	The *ErrorCode* parameter allows you to retrieve documents that failed indexing with a specific error code. You can use the cmdlet without this parameter to list all failed documents for a mailbox, a mailbox database or a Mailbox server. The output includes the error codes and reason for failure. If required, you can then restrict the output to a specific error code from the results.
FailureMode	Optional	Microsoft.Exchange.Search.Core.Abstraction.FailureMode	The *FailureMode* parameter specifies the type of error. Use the following values. • `Transient` Returns items that couldn't be indexed due to transient errors. Exchange Search attempts to index these items again. • `Permanent` Returns items that couldn't be indexed due to a permanent error. Exchange Search does

			not attempt to index these items again. • All Returns items that couldn't be indexed regardless of nature of failure.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
StartDate	Optional	System.DateTime	The *StartDate* parameter specifies the start date of the date range. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxDatabase

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-23*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxDatabase** cmdlet to retrieve one or more mailbox database objects from a server or organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxDatabase -Server <ServerIdParameter> <COMMON PARAMETERS>
```

```
Get-MailboxDatabase [-Identity <DatabaseIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-DumpsterStatistics
<SwitchParameter>] [-IncludePreExchange2013 <SwitchParameter>] [-Status
<SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves information about all the mailbox databases in the Exchange organization, including the mailbox databases that reside on computers running Exchange 2013 and earlier versions of Exchange.

```
Get-MailboxDatabase -IncludePreExchange2013
```

EXAMPLE 2

This example retrieves information about MailboxDatabase01 on Server01. This example also retrieves the status information, and pipes the output to the **Format-List** cmdlet so that you can view all the information about the mailbox database.

```
Get-MailboxDatabase -Identity MailboxDatabase01 -Server
Server01 -Status | Format-List
```

Detailed Description

If you use the **Get-MailboxDatabase** cmdlet with no parameters, it retrieves information about all mailbox databases in the Exchange organization. If you use the **Get-MailboxDatabase** cmdlet with the *Server* parameter, it retrieves information about all mailbox databases on the server that you specify.

The following list describes the properties that are returned in the results.

- **Name** Name of the database.
- **Server** Server hosting the database.
- **Recovery** Specifies whether the new database is designated as a recovery database.
- **ReplicationType** Replication type of the database.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Database Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name of the server from which to retrieve mailbox database information. If you specify this parameter, the command retrieves information about all of the mailbox databases on the server that you specify.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
DumpsterStatistics	Optional	System.Management.Automation.SwitchParameter	The *DumpsterStatistics* switch specifies that transport dumpster statistics be returned with the database status.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Identity* parameter specifies a mailbox database. You can use the following values: • GUID • Distinguished name (DN) • Database name

			If you have multiple databases with the same name, the command retrieves all databases with the same name in the specified scope.
IncludePreExchange2013	Optional	System.Management.Automation.SwitchParameter	The *IncludePreExchange2013* switch parameter specifies whether to return information about the mailbox databases that reside on computers running Microsoft Exchange Server 2013 and earlier versions of Exchange.
Status	Optional	System.Management.Automation.SwitchParameter	The *Status* switch specifies whether to retrieve the available free space in the database root and information about the following attributes: • **BackupInProgress** • **Mounted** • **OnlineMaintenanceInProgress** You don't need to specify a value with this switch. If you specify this switch, you should format the

			output in such a way that you can view the additional attributes, for example, pipe the output to the **Format-List** cmdlet.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailboxDatabase

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-MailboxDatabase** cmdlet to create a mailbox database, or a recovery database. Each database you create must have a unique name in the organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MailboxDatabase -Name <String> [-AutoDagExcludeFromMonitoring <$true |
$false>] [-IsExcludedFromInitialProvisioning <SwitchParameter>] [-
IsExcludedFromProvisioning <$true | $false>] [-IsSuspendedFromProvisioning
<$true | $false>] [-OfflineAddressBook <OfflineAddressBookIdParameter>] [-
PublicFolderDatabase <DatabaseIdParameter>] <COMMON PARAMETERS>


New-MailboxDatabase -Recovery <SwitchParameter> [-Name <String>] <COMMON
PARAMETERS>


COMMON PARAMETERS: -Server <ServerIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-EdbFilePath
<EdbFilePath>] [-LogFolderPath <NonRootLocalLongFullPath>] [-
MailboxProvisioningAttributes <MailboxProvisioningAttributes>] [-
SkipDatabaseLogFolderCreation <SwitchParameter>] [-WhatIf
```

```
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the mailbox database DB1. This example also uses a non-default location for the database file.

```
New-MailboxDatabase -Name "DB1" -EdbFilePath D:
\ExchangeDatabases\DB1\DB1.edb
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox database permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name of the new mailbox database.
Recovery	Required	System.Management.Automation.SwitchParameter	The *Recovery* parameter specifies that the new database is designated as a recovery database.
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server on which you want to create the database.
AutoDagExcludeFrom	Optional	System.Boolean	The

Monitoring			*AutoDagExcludeFromMonitoring* parameter specifies that the database being created should not be monitored by managed availability.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EdbFilePath	Optional	Microsoft.Exchange.Data.EdbFilePath	The *EdbFilePath* parameter specifies the path to the database files. The default value is %programfiles%

			\Microsoft\Exchange Server\V15\Mailbox \<*Database name*>.edb.
IsExcludedFromInitialProvisioning	Optional	System.Management.Automation.SwitchParameter	The *IsExcludedFromInitialProvisioning* parameter specifies that this database is temporarily not considered by the mailbox provisioning load balancer. If the *IsExcludedFromInitialProvisioning* parameter is enabled, new mailboxes aren't added automatically to this database.
IsExcludedFromProvisioning	Optional	System.Boolean	The *IsExcludedFromProvisioning* parameter specifies whether this database is considered by the mailbox provisioning load balancer. If the *IsExcludedFromProvisioning* parameter is set to `$true`, no new mailboxes are automatically added to this database.

IsSuspendedFromProvisioning	Optional	System.Boolean	The *IsSuspendedFromProvisioning* parameter specifies whether this database is temporarily considered by the mailbox provisioning load balancer.
LogFolderPath	Optional	Microsoft.Exchange.Data.NonRootLocalLongFullPath	The *LogFolderPath* parameter specifies the folder location for log files.
MailboxProvisioningAttributes	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MailboxProvisioningAttributes	This parameter is reserved for internal Microsoft use.
OfflineAddressBook	Optional	Microsoft.Exchange.Configuration.Tasks.OfflineAddressBookIdParameter	The *OfflineAddressBook* parameter specifies the associated offline address book (OAB) for the new mailbox database.
PublicFolderDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *PublicFolderDatabase* parameter specifies the associated public folder database for the new mailbox database.
SkipDatabaseLogFolde	Optional	System.Management.A	This parameter is

rCreation		utomation.SwitchParameter	reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailboxDatabase

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-MailboxDatabase** cmdlet to delete a mailbox database object.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MailboxDatabase -Identity <DatabaseIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the mailbox database MailboxDatabase01.

```
Remove-MailboxDatabase -Identity MailboxDatabase01
```

Detailed Description

If the mailbox database has a database copy, the **Remove-MailboxDatabase** cmdlet also removes the copy.

The **Remove-MailboxDatabase** cmdlet removes only the database object from Active Directory. It doesn't remove the physical database files. You must remove the database files manually after you run the **Remove-MailboxDatabase** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Database Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Identity* parameter specifies the mailbox database to remove. You can use one of the following values: • GUID • Distinguished name (DN) • Name of the mailbox database

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxDatabase

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-27*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-MailboxDatabase** cmdlet to configure a variety of properties for a mailbox database.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxDatabase -Identity <DatabaseIdParameter> [-AllowFileRestore
<$true | $false>] [-AutoDagExcludeFromMonitoring <$true | $false>] [-
AutoDatabaseMountDial <Lossless | GoodAvailability | BestAvailability>] [-
BackgroundDatabaseMaintenance <$true | $false>] [-CalendarLoggingQuota
<Unlimited>] [-CircularLoggingEnabled <$true | $false>] [-Confirm
[<SwitchParameter>]] [-DatabaseGroup <String>] [-
DataMoveReplicationConstraint <None | SecondCopy | SecondDatacenter |
AllDatacenters | AllCopies | CINoReplication | CISecondCopy |
CISecondDatacenter | CIAllDatacenters | CIAllCopies>] [-
DeletedItemRetention <EnhancedTimeSpan>] [-DomainController <Fqdn>] [-
EventHistoryRetentionPeriod <EnhancedTimeSpan>] [-IndexEnabled <$true |
$false>] [-IsExcludedFromInitialProvisioning <$true | $false>] [-
IsExcludedFromProvisioning <$true | $false>] [-IssueWarningQuota
<Unlimited>] [-IsSuspendedFromProvisioning <$true | $false>] [-
JournalRecipient <RecipientIdParameter>] [-MailboxLoadBalanceEnabled
<$true | $false>] [-MailboxLoadBalanceMaximumEdbFileSize
<ByteQuantifiedSize>] [-MailboxLoadBalanceOverloadedThreshold <Int32>] [-
MailboxLoadBalanceRelativeLoadCapacity <Int32>] [-
MailboxLoadBalanceUnderloadedThreshold <Int32>] [-
MailboxProvisioningAttributes <MailboxProvisioningAttributes>] [-
MailboxRetention <EnhancedTimeSpan>] [-MaintenanceSchedule <Schedule>] [-
MountAtStartup <$true | $false>] [-Name <String>] [-OfflineAddressBook
<OfflineAddressBookIdParameter>] [-ProhibitSendQuota <Unlimited>] [-
ProhibitSendReceiveQuota <Unlimited>] [-PublicFolderDatabase
<DatabaseIdParameter>] [-QuotaNotificationSchedule <Schedule>] [-
RecoverableItemsQuota <Unlimited>] [-RecoverableItemsWarningQuota
<Unlimited>] [-RetainDeletedItemsUntilBackup <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the length of time that deleted items are retained. If a specific mailbox has its

own item retention set, that value is used instead of this value, which is set on the mailbox database.

```
Set-MailboxDatabase "Mailbox Database01" -
DeletedItemRetention 7.00:00:00
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Database Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Identity* parameter specifies the mailbox database for which you want to set one or more attributes. You can use the following values: • GUID • Distinguished name (DN) • Name of the mailbox database If you don't specify the server name, the command searches for the database on the local server.
AllowFileRestore	Optional	System.Boolean	The *AllowFileRestore* parameter specifies whether to allow restoring

			a database from a backup. The two possible values are `$true` or `$false`. If you specify `$true`, the command allows a database that doesn't match the database entry in Active Directory to be mounted. If you specify `$false`, the command doesn't allow a database that doesn't match the database entry in Active Directory to be mounted, so you won't be able to replace an existing database with a newly created database.
AutoDagExcludeFromMonitoring	Optional	System.Boolean	The *AutoDagExcludedFromMonitoring* is to exclude a mailbox database from the ServerOneCopyMonitor which alerts an administrator when a replicated database has only one healthy copy available. The default setting for this property is False. If this property is not set to True for a

			database and there is only one copy of the database, alerts will be issued.
AutoDatabaseMountDial	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AutoDatabaseMountDial	This parameter is reserved for internal Microsoft use.
BackgroundDatabaseMaintenance	Optional	System.Boolean	The *BackgroundDatabaseMaintenance* parameter specifies whether the Extensible Storage Engine (ESE) performs database maintenance. The two possible values are `$true` or `$false`. If you specify `$true`, the mailbox database reads the object during database mount and initializes the database to perform the background database maintenance. If you specify `$false`, the mailbox database reads the object during database mount and initializes the database without the option to perform the background database maintenance.

CalendarLoggingQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *CalendarLoggingQuota* parameter specifies how much space to allocate for calendar logging information.
CircularLoggingEnabled	Optional	System.Boolean	The *CircularLoggingEnabled* parameter specifies whether circular logging is enabled. If this parameter is set to `$true`, circular logging is enabled. The default value is `$false`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DatabaseGroup	Optional	System.String	This parameter is reserved for internal Microsoft use.
DataMoveReplicationConstraint	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DataMoveReplicationConstraint	The *DataMoveReplicationConstraint* parameter specifies the throttling behavior for

		Parameter	high availability mailbox moves. The possible values include: • `None` Moves shouldn't be throttled to ensure high availability. Use this setting if the database isn't part of a database availability group (DAG). • `SecondCopy` At least one passive mailbox database copy must have the most recent changes synchronized. This is the default value. Use this setting to indicate that the database is replicated to one or more mailbox database copies. • `SecondDatacenter` At least one passive mailbox database copy in another Active Directory site must have the most recent changes replicated. Use this setting to indicate that the database is replicated to database copies in multiple Active Directory sites. • `AllDatacenters` At least one passive mailbox database copy in each Active Directory site must have the most recent changes replicated. Use this setting to indicate that the database is replicated to database

			copies in multiple Active Directory sites. • `AllCopies` All copies of the database must have the most recent changes replicated. Use this setting to indicate that the database is replicated to one or more mailbox database copies. **Note:** Any value other than `None` enables the Microsoft Exchange Mailbox Replication service to coordinate with Active Manager. For more information, see Active Manager.
DeletedItemRetention	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *DeletedItemRetention* parameter specifies the length of time to keep deleted items. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify a 15-hour interval, enter 15:00:00. The maximum length of time to retain deleted items is 24,855 days. By default, deleted items are retained for 14 days. This attribute applies

			to all mailboxes in this mailbox database that don't have their own item retention attribute set.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EventHistoryRetentionPeriod	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *EventHistoryRetentionPeriod* parameter specifies the length of time to keep event data. This event data is stored in the event history table in the Exchange store. It includes information about changes to various objects in the mailbox database. You can use this parameter to prevent the event history table from becoming too large and using too much disk space. To specify a value, enter it as a time span: dd.hh:mm:ss where d =

			days, h = hours, m = minutes, and s = seconds. For example, to specify a 15-hour interval, enter 15:00:00.
IndexEnabled	Optional	System.Boolean	The *IndexEnabled* parameter specifies whether Exchange Search indexes this mailbox database. The two possible values are `$true` or `$false`. The default value is `$true`.
IsExcludedFromInitialProvisioning	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
IsExcludedFromProvisioning	Optional	System.Boolean	The *IsExcludedFromProvisioning* parameter specifies that this database is permanently not considered by the mailbox provisioning load balancer. If the *IsExcludedFromProvisioning* parameter is enabled, new mailboxes aren't added automatically to this database. You can manually add a mailbox if your role permits.
IssueWarningQuota	Optional	Microsoft.Exchange.Da	The *IssueWarningQuota*

		ta.Unlimited	parameter specifies the mailbox size at which a warning message is sent to the user. This attribute applies to all mailboxes in this mailbox database that don't have their own warning quota attribute set. You must specify either an integer or unlimited. The default value is 1.9 gigabytes (GB).
IsSuspendedFromProvisioning	Optional	System.Boolean	The *IsSuspendedFromProvisioning* parameter specifies that this database is temporarily not considered by the mailbox provisioning load balancer.
JournalRecipient	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *JournalRecipient* parameter specifies the mailbox to which journal reports are sent.
MailboxLoadBalanceEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
MailboxLoadBalanceMaximumEdbFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
MailboxLoadBalanceO	Optional	System.Int32	This parameter is reserved

verloadedThreshold			for internal Microsoft use.
MailboxLoadBalanceRelativeLoadCapacity	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
MailboxLoadBalanceUnderloadedThreshold	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
MailboxProvisioningAttributes	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MailboxProvisioningAttributes	This parameter is reserved for internal Microsoft use.
MailboxRetention	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MailboxRetention* parameter specifies the length of time to keep deleted mailboxes. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify a 15-hour interval, enter 15:00:00. The maximum length of time to retain mailboxes is 24,855 days. By default, deleted mailboxes are retained for 30 days. This attribute applies to all mailboxes in this mailbox database.
MaintenanceSchedule	Optional	Microsoft.Exchange.Data.Schedule	This parameter has been deprecated in Exchange

			2013. While it can be used to change the *MaintenanceSchedule* property of a database, that property is ignored in Exchange 2013 because scheduled maintenance no longer exists.
MountAtStartup	Optional	System.Boolean	The *MountAtStartup* parameter specifies whether to mount this mailbox database when the Microsoft Exchange Information Store service starts. The two possible values for this parameter are `$true` or `$false`.
Name	Optional	System.String	The *Name* parameter specifies the name of the mailbox database.
OfflineAddressBook	Optional	Microsoft.Exchange.Configuration.Tasks.OfflineAddressBookIdParameter	The *OfflineAddressBook* parameter specifies the associated address book for this mailbox database.
ProhibitSendQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *ProhibitSendQuota* parameter specifies the mailbox size at which users associated with mailboxes in this mailbox database can no longer send messages. This

			attribute applies to all mailboxes in this mailbox database that don't have their own prohibit send quota attributes set. You must specify either an integer or unlimited.
ProhibitSendReceiveQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *ProhibitSendReceiveQuota* parameter specifies the mailbox size at which the user associated with this mailbox can no longer send or receive messages. This attribute applies to all mailboxes in this mailbox database that don't have their own prohibit send receive quota attributes set. You must specify either an integer or unlimited.
PublicFolderDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter has been deprecated in Exchange 2013 and it no longer does anything.
QuotaNotificationSchedule	Optional	Microsoft.Exchange.Data.Schedule	This parameter has been deprecated in Exchange 2013 and it no longer does anything.

RecoverableItemsQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *RecoverableItemsQuota* parameter specifies the limit for the Recovery Items folder. When you reach the quota limit, you can't put any more items in the Recovery Items folder.
RecoverableItemsWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *RecoverableItemsWarningQuota* parameter specifies the quota for when a warning event is entered in Event Viewer.
RetainDeletedItemsUntilBackup	Optional	System.Boolean	The *RetainDeletedItemsUntilBackup* parameter specifies whether to retain deleted items until the next backup occurs. The two possible values for this parameter are `$true` or `$false`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-MailboxQuarantine

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-MailboxQuarantine** cmdlet to quarantine mailboxes that affect the availability of the mailbox database.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-MailboxQuarantine -Identity <GeneralMailboxIdParameter> [-
AllowMigration <SwitchParameter>] [-Confirm [<SwitchParameter>]] [-
Duration <EnhancedTimeSpan>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example quarantines the mailbox for the user Brian Johnson.

```
Enable-MailboxQuarantine "Brian Johnson"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox databases" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.GeneralMailboxIdParameter	The *Identity* parameter specifies the mailbox. You can use any value that uniquely identifies the mailbox. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-

			4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
AllowMigration	Optional	System.Management.Automation.SwitchParameter	The *AllowMigration* switch allows a quarantined mailbox to be moved to another mailbox database or to the cloud. Moving a mailbox is one method of correcting data corruption that's required before releasing the mailbox from quarantine. You don't have to specify a value with the *AllowMigration* switch.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You

			must include a colon (:) in the syntax.
Duration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *Duration* parameter specifies how long the mailbox should remain quarantined. The default duration is 24 hours. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-MailboxQuarantine

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-MailboxQuarantine** cmdlet to release quarantined mailboxes.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-MailboxQuarantine -Identity <GeneralMailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example releases the mailbox for the user Brian Johnson from quarantine.

```
Disable-MailboxQuarantine "Brian Johnson"
```

Detailed Description

Mailboxes are quarantined when they affect the availability of the mailbox database. Typically a software fix from Microsoft is required before releasing a mailbox from quarantine. If a fix isn't deployed before releasing the mailbox, the quarantine on the mailbox will be re-enabled if the condition recurs. The default quarantine duration is 24 hours.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox databases" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.GeneralMailboxIdParameter	specifies the mailbox. You can use any value that uniquely identifies the mailbox. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name`

			Example: JPhillips@contoso.com
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxRepairRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxRepairRequest** cmdlet to display information about current mailbox repair requests. Mailbox repair requests are created using the **New-MailboxRepairRequest** cmdlet to detect and fix mailbox corruptions.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxRepairRequest -Identity <StoreIntegrityCheckJobIdParameter> [-
Detailed <SwitchParameter>] <COMMON PARAMETERS>


Get-MailboxRepairRequest -Database <DatabaseIdParameter> [-StoreMailbox
<StoreMailboxIdParameter>] <COMMON PARAMETERS>


Get-MailboxRepairRequest -Mailbox <MailboxIdParameter> [-Archive
<SwitchParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example displays the value of the *Identity* property for all mailbox repair requests for all mailbox servers in your organization; the second command displays information about a specific mailbox repair request that was returned by the first command.

```
Get-MailboxDatabase | Get-MailboxRepairRequest | FT
Identity


Get-MailboxRepairRequest -Identity 5b8ca3fa-8227-427f-af04-
9b4f206d611f\335c2b06-321d-4e73-b2f7-3dc2b02d0df5\374289de-
b899-42dc-8391-4f8579935f1f | FL
```

EXAMPLE 2

This example displays repair request information for the mailbox of Ann Beebe using the *Mailbox* parameter.

```
Get-MailboxRepairRequest -Mailbox "Ann Beebe" | FL
```

EXAMPLE 3

This example uses the *Database* and *StoreMailbox* parameters to display the *Identity* property of the repair request for the mailbox of Ann Beebe.

```
$MailboxGuid = Get-MailboxStatistics annb

Get-MailboxRepairRequest -Database $MailboxGuid.Database -
StoreMailbox $MailboxGuid.MailboxGuid | FL Identity
```

Detailed Description

The **Get-MailboxRepairRequest** cmdlet displays information about mailbox repair requests. This information includes:

- The mailbox GUID.
- The type of corruption that was specified when the mailbox repair request was created.
- The progress of the repair request in percentage of completion.
- The number of corruptions detected and fixed.
- The status of the repair request; values are `Queued`, `Running`, `Succeeded` and `Failed`.
- The date and time when the mailbox repair request was created and when it finished.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox repair request" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Database	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Database* parameter specifies the database on which you run this command. If you use this parameter, all mailboxes on the database are searched for corruptions. You can use the following values:

			• GUID of the database • Database name You can't use this parameter with the *Mailbox* parameter.
Identity	Required	Microsoft.Exchange.Management.Tasks.StoreIntegrityCheckJobIdParameter	The *Identity* parameter specifies the mailbox repair request to display information about. Mailbox repair requests are identified by a complex GUID that is created when a new mailbox repair request is created. This GUID consists of a database ID, a Request ID, and a job ID. The format is `<DatabaseGuid>\<RequestGuid>\<JobGuid>`.
Mailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the mailbox that you want to get mailbox repair request information about. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN)

			• LegacyExchangeDN • SMTP address • Alias You can't use this parameter with the *Database* parameter.
Archive	Optional	System.Management.Automation.SwitchParameter	If the associated archive mailbox was included when the mailbox repair request was created, use the *Archive* parameter to display information about the archive mailbox. If you don't specify this parameter, only information about the primary mailbox is returned. You can't use this parameter with the *Database* parameter.
Detailed	Optional	System.Management.Automation.SwitchParameter	Use the *Detailed* parameter to display mailbox-level repair tasks associated with the repair request.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			writes this configuration change to Active Directory.
StoreMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.StoreMailboxIdParameter	The *StoreMailbox* parameter specifies the mailbox GUID of the mailbox that you want to get mailbox repair request information about. Use this parameter with the *Database* parameter. Use the Get-MailboxStatistics cmdlet to find the mailbox GUID for a mailbox.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailboxRepairRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-MailboxRepairRequest** cmdlet to detect and fix mailbox corruptions. You can run this command against a specific mailbox or against a database. While this task is running, mailbox access is disrupted only for the mailbox being repaired. If you're running this command against a database, only the mailbox being repaired is disrupted. All other mailboxes on the database remain operational.

For information about the parameter sets in the Syntax section below, see Syntax.

Note:

After you begin the repair request, it can't be stopped unless you dismount the database.

```
New-MailboxRepairRequest -Mailbox <MailboxIdParameter> [-Archive
<SwitchParameter>] <COMMON PARAMETERS>


New-MailboxRepairRequest -Database <DatabaseIdParameter> [-StoreMailbox
<StoreMailboxIdParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: -CorruptionType <MailboxCorruptionType[]> [-Confirm
[<SwitchParameter>]] [-DetectOnly <SwitchParameter>] [-DomainController
<Fqdn>] [-Force <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example detects and repairs all folder views for the mailbox tony@contoso.com.

```
New-MailboxRepairRequest -Mailbox tony@contoso.com -
CorruptionType FolderView
```

EXAMPLE 2

This example only detects and reports on `ProvisionedFolder` and `SearchFolder` corruption issues to Ayla Kol's mailbox. This command doesn't repair the mailbox.

```
New-MailboxRepairRequest -Mailbox ayla -CorruptionType
ProvisionedFolder,SearchFolder -DetectOnly
```

EXAMPLE 3

This example detects and repairs `AggregateCounts` for all mailboxes on mailbox database MBX-DB01.

```
New-MailboxRepairRequest -Database MBX-DB01 -CorruptionType
AggregateCounts
```

EXAMPLE 4

This example detects and repairs all corruption types for Ayla Kol's mailbox and archive.

```
New-MailboxRepairRequest -Mailbox ayla -CorruptionType
ProvisionedFolder,SearchFolder,AggregateCounts,Folderview -
Archive
```

EXAMPLE 5

This example creates a variable that identifies Ann Beebe's mailbox and then uses the variable to specify the values for the *Database* and *StoreMailbox* parameters to create a request to detect and repair all corruption types.

```
$Mailbox = Get-MailboxStatistics annb

New-MailboxRepairRequest -Database $Mailbox.Database -
StoreMailbox $Mailbox.MailboxGuid -CorruptionType
ProvisionedFolder,SearchFolder,AggregateCounts,Folderview
```

Detailed Description

To avoid any performance problems, there are limits placed on the number of simultaneous repair requests that can be submitted per server. Only one request can be active for a database-level repair, or up to 100 requests can be active for a mailbox-level repair per server.

The **New-MailboxRepairRequest** cmdlet detects and fixes the following types of mailbox corruptions:

- Search folder corruptions (`SearchFolder`)
- Aggregate counts on folders that aren't reflecting correct values (`AggregateCounts`)
- Views on folders that aren't returning correct contents (`FolderView`)
- Provisioned folders that are incorrectly pointing into parent folders that aren't provisioned (`ProvisionedFolder`)

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox repair request" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
CorruptionType	Required	Microsoft.Exchange.Management.Tasks.MailboxCorruptionType[]	The *CorruptionType* parameter specifies the type of corruption that

			you want to detect and repair. You can use the following values: • `SearchFolder` • `AggregateCounts` • `ProvisionedFolder` • `FolderView` You can search for multiple corruption types at a time. Separate multiple types with a comma, for example, `SearchFolder,AggregateCounts`.
Database	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Database* parameter specifies the database on which you run this command. If you use this parameter, all mailboxes on the database are searched for corruptions. To avoid performance issues, you're limited to one active database repair request at a time. You can use the following values: • GUID of the database • Database name You can't use this parameter in conjunction with the *Mailbox* parameter.
Mailbox	Required	Microsoft.Exchange.Configuration.Tasks.Mail	The *Mailbox* parameter specifies the mailbox on

		boxIdParameter	which you run this command. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SMTP address • Alias You can't use this parameter in conjunction with the *Database* parameter.
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* parameter specifies whether to detect corruptions or repair the archive mailbox associated with the specified mailbox. If you don't specify this parameter, only the primary mailbox is repaired. You can't use this parameter in conjunction with the *Database* parameter.
Confirm	Optional	System.Management.	The *Confirm* switch can be

		Automation.SwitchParameter	used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DetectOnly	Optional	System.Management.Automation.SwitchParameter	The *DetectOnly* parameter specifies that you want this command to report errors, but not fix them. You don't have to specify a value with this parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies that the cmdlet should run immediately and not wait to be dispatched by workload management.
StoreMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.Stor	The *StoreMailbox* parameter specifies the

		eMailboxIdParameter	mailbox GUID of the mailbox you want to repair. Use this parameter with the *Database* parameter. Run the Get-MailboxStatistics cmdlet to find the mailbox GUID for a mailbox.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailboxRepairRequest

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-MailboxRepairRequest** cmdlet to remove mailbox repair requests from a mailbox database that were created using the **New-MailboxRepairRequest** cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MailboxRepairRequest -Identity <StoreIntegrityCheckJobIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes all mailbox repair requests for the mailbox database EXCH-MBX-01.

```
Get-MailboxDatabase -Identity "EXCH-MBX-01" | Get-
MailboxRepairRequest | Remove-MailboxRepairRequest
```

EXAMPLE 2

This example removes all related mailbox repair requests that have the same `<DatabaseGuid>\<RequestGuid>`. The example uses the **Get-MailboxRepairRequest** cmdlet to display the value of the *Identity* parameter for all mailbox repair request for EXCH-MBX-02 mailbox database.

```
Get-MailboxDatabase -Identity "EXCH-MBX-02" | Get-
MailboxRepairRequest | FL Identity


Remove-MailboxRepairRequest -Identity 5b8ca3fa-8227-427f-
af04-9b4f206d611f\335c2b06-321d-4e73-b2f7-3dc2b02d0df5
```

EXAMPLE 3

This example deletes a specific mailbox repair request by specifying the unique `<DatabaseGuid>\<RequestGuid>\<JobGuid>` identity value. The example also uses the **Get-MailboxRepairRequest** cmdlet to display the identities of all mailbox repair request for the EXCH-MBX-02 mailbox database.

```
Get-MailboxDatabase -Identity "EXCH-MBX-02" | Get-
MailboxRepairRequest | FL Identity

Remove-MailboxRepairRequest -Identity 5b8ca3fa-8227-427f-
af04-9b4f206d611f\189c7852-49bd-4737-a53e-6e6caa5a183c
\1d8ca58a-186f-4dc6-b481-f835b548a929
```

Detailed Description

You can run the **Remove-MailboxRepairRequest** cmdlet to remove all mailbox repair requests for a specific database, for a group of related mailbox repair requests, or for a specific mailbox repair request. Mailbox repair requests are identified by a complex GUID with the following format: `<DatabaseGuid>\<RequestGuid>\<JobGuid>`. The `DatabaseGuid` identifies the mailbox database where the mailbox being repaired is located. The `RequestGuid` identifies related requests that may contain more than one job if the request runs more than one task or targets more than one mailbox. The `JobGuid` identifies a unique job. See the examples to remove all requests on a mailbox database, remove a group of related requests that share the same `RequestGuid`, or remove a specific request by specifying the complete `<DatabaseGuid>\<RequestGuid>\<JobGuid>` value.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox repair request" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.Tasks.StoreIntegrityCheckJobIdParameter	The *Identity* parameter specifies the mailbox repair request to remove. Mailbox repair requests are identified by a complex GUID that is created when a new mailbox repair request is created. This GUID consists of a database ID, a Request ID, and a

			job ID. The format is `<DatabaseGuid>\<RequestGuid>\<JobGuid>`. Use the Get-MailboxRepairRequest cmdlet to find the identity of a mailbox repair request.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on

			the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-MapiConnectivity

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-17*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Test-MapiConnectivity** cmdlet to verify server functionality by logging on to the mailbox that you specify. If you don't specify a mailbox, the cmdlet logs on to the SystemMailbox on the database that you specify.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-MAPIConnectivity [-IncludePassive <SwitchParameter>] [-Server
<ServerIdParameter>] <COMMON PARAMETERS>

Test-MAPIConnectivity -Database <DatabaseIdParameter> [-CopyOnServer
<ServerIdParameter>] <COMMON PARAMETERS>
```

```
Test-MAPIConnectivity -Identity <MailboxIdParameter> [-Archive
<SwitchParameter>] [-CopyOnServer <ServerIdParameter>] [-
EnableSoftDeletedRecipientLogon <SwitchParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-ActiveDirectoryTimeout <Int32>] [-
AllConnectionsTimeout <Int32>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-MonitoringContext <$true | $false>] [-
PerConnectionTimeout <Int32>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests connectivity to the server Server01.

```
Test-MapiConnectivity -Server "Server01"
```

EXAMPLE 2

This example tests connectivity to a mailbox, specified as a domain name and user name.

```
Test-MapiConnectivity -Identity "midwest\john"
```

Detailed Description

The **Test-MapiConnectivity** cmdlet verifies server functionality. This cmdlet logs on to the mailbox that you specify (or to the SystemMailbox if you don't specify the *Identity* parameter) and retrieves a list of items in the Inbox. Logging on to the mailbox tests two critical protocols used when a client connects to a Mailbox server: MAPI and LDAP. During authentication, the **Test-MapiConnectivity** cmdlet indirectly verifies that the MAPI server, Exchange store, and Directory Service Access (DSAccess) are working.

The cmdlet logs on to the mailbox that you specify using the credentials of the account with which you're logged on to the local computer. After a successful authentication, the **Test-MapiConnectivity** cmdlet accesses the mailbox to verify that the database is working. If a successful connection to a mailbox is made, the cmdlet also determines the time that the logon attempt occurred.

There are three distinct parameters that you can use with the command: *Database*, *Identity*, and *Server*:

- The *Database* parameter takes a database identity and tests the ability to log on to the system mailbox on the specified database.
- The *Identity* parameter takes a mailbox identity and tests the ability to log on to a specific mailbox.
- The *Server* parameter takes a server identity and tests the ability to log on to each system mailbox on the specified server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "MAPI connectivity" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Database	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *Database* parameter specifies the database on which to test the connectivity to the system mailbox. If you don't specify this parameter or the *Identity* parameter, the command tests the SystemMailbox on each active database on the server that you specify, or on the local server if you don't specify the *Server* parameter.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies a mailbox to test. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name

			(UPN) • Legacy Exchange DN • SMTP address • Alias This parameter accepts pipeline input from the **Get-Mailbox** or **Get-Recipient** cmdlet. If an object is piped from the **Get-Mailbox** cmdlet or **Get-Recipient** cmdlet, this parameter isn't required. If you don't specify this parameter, the cmdlet tests the SystemMailbox on the database that you specify.
ActiveDirectoryTimeout	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *ActiveDirectoryTimeout* parameter specifies the amount of time, in seconds, allowed for each Active Directory operation to complete before the operation times out. The default value is 15 seconds.

AllConnectionsTimeout	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *AllConnectionsTimeout* parameter specifies the amount of time, in seconds, allowed for all connections to complete before the cmdlet times out. The time-out countdown doesn't begin until all information necessary to perform the connections is gathered from Active Directory.The default value is 90 seconds.
Archive	Optional	System.Management.Automation.SwitchParameter	The *Archive* parameter specifies whether to test the MAPI connectivity of the personal archive associated with the specified mailbox. If you don't specify this parameter, only the primary mailbox is tested.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To

			suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
CopyOnServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	This parameter is available only in on-premises Exchange 2013. The *CopyOnServer* parameter is used to test MAPI connectivity to a specific database copy on the servers specified with the *Server* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
EnableSoftDeletedRecipientLogon	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IncludePassive	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013.

			Without the *IncludePassive* parameter, the cmdlet tests MAPI connectivity from active database copies only. Using the *IncludePassive* parameter, you can have the cmdlet test MAPI connectivity from all active and passive database copies.
MonitoringContext	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *MonitoringContext* parameter includes or excludes the associated monitoring events and performance counters in the results. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you specify the value `$true`, the monitoring events and performance counters are included in the command results. Typically, you include the monitoring events and performance counters in the results when the output is passed

			to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
PerConnectionTimeout	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *PerConnectionTimeout* parameter specifies the amount of time, in seconds, allowed for each connection to complete before the connection times out. The default value is 10 seconds.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	This parameter is available only in on-premises Exchange 2013. The *Server* parameter specifies the server on which you will test the MAPI connectivity. The command tests the MAPI connectivity to each system mailbox hosted on active databases on the specified server. If you don't specify this parameter, the command

			tests the mailbox on the local server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SearchDocumentFormat

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-SearchDocumentFormat** cmdlet to retrieve details of file formats supported by

Exchange Search.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-SearchDocumentFormat [-Identity <SearchDocumentFormatId>] [-Server
<ServerIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves a list of all file formats supported by Exchange Search.

```
Get-SearchDocumentFormat
```

EXAMPLE 2

This example retrieves all properties of the docx file format.

```
Get-SearchDocumentFormat docx | Format-List *
```

Detailed Description

In Microsoft Exchange Server 2013, Exchange Search includes built-in support for indexing many file formats. Output from the **Get-SearchDocumentFormat** cmdlet includes details about each supported file format, including whether content indexing is enabled for the file format, the format handler, and the file extension (such as .docx).

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Search - diagnostics" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Optional	Microsoft.Exchange.Management.Search.SearchDocumentFormatId	The *Identity* parameter specifies the identity of a file format.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	This parameter is available only in on-premises Exchange

			2013. The *Server* parameter specifies the name of the server against which the command is executed.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-SearchDocumentFormat

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-SearchDocumentFormat** cmdlet to add a format-specific filter to those used by Exchange search.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-SearchDocumentFormat -Extension <String> -Identity
<SearchDocumentFormatId> -MimeType <String> -Name <String> [-Confirm
[<SwitchParameter>]] [-Enabled <$true | $false>] [-Server
<ServerIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example creates a new search document format with an extension of .sct and a MIME type of

text/scriptlet.

```
New-SearchDocumentFormat -Name "Proprietary SCT Formats" -
MimeType text/scriptlet -Extension .sct -Identity
ProprietarySCT1
```

Detailed Description

After running the **New-SearchDocumentFormat** cmdlet, your must run the following cmdlet to restart the search service. There will be a brief search outage.

```
Restart-Service HostControllerService
```

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Search - diagnostics" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Extension	Required	System.String	The *Extension* parameter specifies the file type to be processed by the filter, and is designated by the common file extension associated with the file type. Examples include .MP3, .JPG, and .PNG. Note the leading period.
Identity	Required	Microsoft.Exchange.Management.Search.SearchDocumentFormatId	The *Identity* parameter uniquely identifies the new search document format. For example, an identity of "PropSCT" might specify a proprietary document

			format which is supported by a custom **IFilter**. The *Identity* parameter must be unique within the search document formats.
MimeType	Required	System.String	The *MimeType* parameter specifies the MIME type of the format.
Name	Required	System.String	The *Name* specifies a friendly name for the format, but does not need to be unique. For example, you might have several different formats (supported by custom **IFilters**) that are used to index output from a proprietary system called "My Business Output". You could use the *Name* parameter to create a category of formats called "My Business Output Formats", and uniquely identify each format within that group using the *Identity* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the new format is enabled at creation. Valid input for this parameter is `$true` or `$false`. The default is `$true`.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.
WhatIf	Optional	System.Management.Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-SearchDocumentFormat

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-SearchDocumentFormat** cmdlet to remove a format-specific filter from those used by Exchange search. Only filters added with **New-SearchDocumentFormat** can be removed.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-SearchDocumentFormat -Identity <SearchDocumentFormatId> [-Confirm
[<SwitchParameter>]] [-Server <ServerIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example removes the search document format with an identity equal to "sct".

```
Remove-SearchDocumentFormat -Identity sct
```

Detailed Description

After running the **Remove-SearchDocumentFormat** cmdlet, your must run the following cmdlet to restart the search service. There will be a brief search outage.

```
Restart-Service HostControllerService
```

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Search - diagnostics" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.Search.SearchDocumentFormatId	The *Identity* parameter uniquely identifies the format to be removed. You can use the **Get-SearchDocumentFormat** cmdlet to view the identities of the installed formats.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation

			prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-SearchDocumentFormat

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-SearchDocumentFormat** cmdlet to enable or disable the file format for Exchange Search.

Important:
When you disable a file format for content indexing by Exchange Search, contents of the file become unsearchable by Exchange Search clients such as Microsoft Office Outlook Web App, Microsoft Outlook in online mode, and In-Place eDiscovery.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-SearchDocumentFormat -Enabled <$true | $false> -Identity
<SearchDocumentFormatId> [-Confirm [<SwitchParameter>]] [-Server
<ServerIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This command disables the Zip file format for indexing by Exchange Search.

```
Set-SearchDocumentFormat ZIP -Enabled $false
```

Detailed Description

In Microsoft Exchange Server 2013, Exchange Search includes built-in support for indexing many file formats. If you disable indexing for a supported file format, items containing an attachment of that file type aren't considered unsearchable. When you perform an In-Place eDiscovery search, and you select the option to include unsearchable items, only items that are actually unsearchable are returned. Items that weren't searched because the associated file format is set as unsearchable aren't returned.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Search - diagnostics" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Enabled	Required	System.Boolean	The *Enabled* parameter specifies whether the file format is enabled. Set the parameter to `$false` to disable the format for content indexing.
Identity	Required	Microsoft.Exchange.Management.Search.SearchDocumentFormatId	The *Identity* parameter specifies the identity of the file format.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name of the server against which the command is executed.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-StoreMailbox

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-StoreMailbox** cmdlet to purge the mailbox and all of its message content from the mailbox database. This results in permanent data loss for the mailbox being purged. You can only run this cmdlet against disconnected or soft-deleted mailboxes. Running this command against an active mailbox fails, and you receive an error.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-StoreMailbox -Database <DatabaseIdParameter> -Identity
<StoreMailboxIdParameter> -MailboxState <Disabled | SoftDeleted> [-Confirm
[<SwitchParameter>]] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example purges the soft-deleted mailbox for Ayla Kol from mailbox database MBD01.

```
Remove-StoreMailbox -Database MBD01 -Identity Ayla -
MailboxState SoftDeleted
```

EXAMPLE 2

This example permanently purges the disconnected mailbox with the GUID 2ab32ce3-fae1-4402-9489-c67e3ae173d3 from mailbox database MBD01.

```
Remove-StoreMailbox -Database MBD01 -Identity "2ab32ce3-
fae1-4402-9489-c67e3ae173d3" -MailboxState Disabled
```

EXAMPLE 3

This example permanently purges all soft-deleted mailboxes from mailbox database MBD01.

```
Get-MailboxStatistics -Database MBD01 | where
{$_.DisconnectReason -eq "SoftDeleted"} | foreach {Remove-
StoreMailbox -Database $_.database -Identity $_.mailboxguid
-MailboxState SoftDeleted}
```

Detailed Description

When mailboxes are moved from a Microsoft Exchange Server 2013 database to any other

database, Exchange doesn't fully delete the mailbox from the source database immediately upon completion of the move. Instead, the mailbox in the source mailbox database is switched to a *soft-deleted* state, which allows mailbox data to be accessed during a mailbox restore operation by using the new **MailboxRestoreRequest** cmdlet set. The soft-deleted mailboxes are retained in the source database until the deleted mailbox retention period expires.

To view soft-deleted mailboxes, run the Get-MailboxStatistics cmdlet against a database using the property **DisconnectReason** with a value of `SoftDeleted`.

A mailbox is marked as Disabled immediately after the **Disable-Mailbox** or **Remove-Mailbox** command completes. Exchange retains disabled mailboxes in the mailbox database based on the deleted mailbox retention settings configured for that mailbox database. After the specified period of time, the mailbox is permanently deleted.

To view disabled mailboxes, run the **Get-MailboxStatistics** cmdlet against a database using the property **DisconnectReason** with a value of `Disabled`.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remove store mailbox" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Database	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Database* parameter specifies the identity of the mailbox database on which the mailbox that you want to remove resides. This parameter accepts the following values: • Database name • GUID
Identity	Required	Microsoft.Exchange.Configuration.Tasks.StoreMailboxIdParameter	The *Identity* parameter specifies the identity of the mailbox that you want to remove. This parameter accepts the following

			values: • GUID • Distinguished name (DN) • User principal name (UPN) • LegacyExchangeDN • *Domain\Account Name* • SMTP address
MailboxState	Required	Microsoft.Exchange.Management.StoreTasks.MailboxStateParameter	The *MailboxState* parameter specifies the mailbox state on the source mailbox database. This parameter accepts the following values: • `Disabled` • `SoftDeleted`
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what

			changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-StoreMailboxState

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Update-StoreMailboxState** cmdlet to synchronize the mailbox state for a mailbox in the Exchange mailbox store with the state of the corresponding Active Directory user account.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-StoreMailboxState -Database <DatabaseIdParameter> -Identity
<StoreMailboxIdParameter> [-Confirm [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example updates the mailbox state for a mailbox located on the mailbox database MDB01 and whose GUID is 4a830e3f-fd07-4629-baa1-8bce16b86d88.

```
Update-StoreMailboxState -Database MDB01 -Identity
4a830e3f-fd07-4629-baa1-8bce16b86d88
```

EXAMPLE 2

This example updates the mailbox state for all mailboxes on the mailbox database MDB02.

```
Get-MailboxStatistics -Database MDB02 | ForEach { Update-
StoreMailboxState -Database $_.Database -Identity
$_.MailboxGuid -Confirm:$false }
```

EXAMPLE 3

This example updates the mailbox state for all disconnected mailboxes on the mailbox database MDB03.

```
Get-MailboxStatistics -Database MDB03 | Where
{ $_.DisconnectReason -ne $null } | ForEach { Update-
StoreMailboxState -Database $_.Database -Identity
$_.MailboxGuid -Confirm:$false }
```

Detailed Description

The **Update-StoreMailboxState** cmdlet forces the mailbox store state in the Exchange store to be synchronized with Active Directory. In some cases, it's possible that the store state for a mailbox to become out-of-sync with the state of the corresponding Active Directory user account. This can result from Active Directory replication latency. For example, if a mailbox-enabled user account is disabled in Active Directory but isn't marked as disabled in the Exchange mailbox store. In this case, running the **Update-StoreMailboxState** will synchronize the mailbox store state with the state of the Active Directory user account and mark the mailbox as disabled in the mailbox store. You can use this command to troubleshoot issues that may be a result when the store state for a mailbox is unexpected or if you suspect that the store state is different than the state for the corresponding Active Directory account.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Remove store mailbox" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description

Database	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Database* parameter specifies the identity of the mailbox database that contains the mailbox that you want to update the store state for. This parameter accepts the following values: • Database name • GUID
Identity	Required	Microsoft.Exchange.Configuration.Tasks.StoreMailboxIdParameter	The *Identity* parameter specifies the identity of the mailbox that you want to update the store state for. Use the mailbox GUID as the value for this parameter. Run the following command to obtain the mailbox GUID and other information for all mailboxes in your organization. `Get-MailboxDataba`
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the

			confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-StoreUsageStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox database cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-17*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-StoreUsageStatistics** cmdlet to aid in diagnosing performance issues with your servers or databases.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-StoreUsageStatistics -Identity <GeneralMailboxIdParameter> [-
CopyOnServer <ServerIdParameter>] <COMMON PARAMETERS>

Get-StoreUsageStatistics -Database <DatabaseIdParameter> [-CopyOnServer
<ServerIdParameter>] <COMMON PARAMETERS>

Get-StoreUsageStatistics -Server <ServerIdParameter> [-IncludePassive
<SwitchParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-Filter <String>]
```

Examples

Below are some examples of how to use the **Get-StoreUsageStatistics** cmdlet.

EXAMPLE 1

This example retrieves store usage statistics for all active databases on server EX1.

```
Get-StoreUsageStatistics -Server EX1 | ft -auto
```

EXAMPLE 2

This example retrieves store usage statistics for all active and passive databases on server EX1.

```
Get-StoreUsageStatistics -Server EX1 -IncludePassive | ft -
auto
```

EXAMPLE 3

This example retrieves store usage statistics for database DB1, and sorts the output by the 10 highest log file generators.

```
Get-StoreUsageStatistics -Database DB1 | Sort-Object
LogRecordBytes -desc | Select-Object -First 10 | ft
DigestCategory, *guid, LogRecordBytes, *time* -auto
```

Detailed Description

The Microsoft Exchange Information Store collects per-user information on latency, input/output (I/

O), page counts, processor usage, and *TimeInServer*. The *TimeInServer* metric represents the total time that synchronous and asynchronous requests spend in the Microsoft Exchange Information Store for a user's mailbox. You can retrieve this resource information in the Microsoft Exchange Information Store for the 25 highest usage accounts on a specified database. Usage of a mailbox is defined as the amount of server time spent in performing operations for that mailbox. The cmdlet reports the top 25 users for every one-minute period for the last 10 minutes (250 objects per ten-minute interval). The resource usage is an indicator of the load that different users are placing on the server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox databases" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Database	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Database* parameter specifies the name of the mailbox database. When you specify a value for the *Database* parameter, the Exchange Management Shell returns usage statistics for the top 25 mailboxes on the database specified. You can use the following value: • *Database*
Identity	Required	Microsoft.Exchange.Configuration.Tasks.GeneralMailboxIdParameter	The *Identity* parameter specifies a mailbox. When you specify a value for the *Identity* parameter, the command looks up the mailbox specified in the

			Identity parameter, connects to the server where the mailbox resides, and returns the statistics for the mailbox. You can use one of the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias **Note:** Results are returned for the mailbox only if it's one of the top 25 users of store resources.
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server from which you want to obtain mailbox statistics. You can use one of the following values: • Fully qualified domain name (FQDN) • NetBIOS name When you specify a value for the *Server* parameter, the command returns

			usage statistics for the top 25 mailboxes on all the active databases on the specified server. If you don't specify this parameter, the command returns logon statistics for the local server.
CopyOnServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *CopyOnServer* parameter is used to retrieve statistics from a specific database copy from the server specified with the *Server* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients.
IncludePassive	Optional	System.Management.Automation.SwitchParameter	Without the *IncludePassive* parameter, the cmdlet retrieves statistics from active database copies only. Using the *IncludePassive* parameter, you can have

			the cmdlet return statistics from all active and passive database copies.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Mailbox server cmdlets

Exchange Server 2013 > Exchange Management Shell > Exchange 2013 cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-08*

Test-AssistantHealth

Get-MailboxServer

Set-MailboxServer

Test-MRSHealth

Test-AssistantHealth

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox server cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-AssistantHealth** cmdlet to verify that the Microsoft Exchange Mailbox Assistants service (MSExchangeMailboxAssistants) is healthy, to recover from health issues, and to report the

status of the diagnosis or recovery action.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-AssistantHealth [-IncludeCrashDump <SwitchParameter>] [-
MaxProcessingTimeInMinutes <UInt32>] [-MonitoringContext
<SwitchParameter>] [-ResolveProblems <SwitchParameter>] [-ServerName
<ServerIdParameter>] [-WatermarkBehindWarningThreholdInMinutes <UInt32>]
[-Confirm [<SwitchParameter>]] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example detects and repairs the mailbox assistant's health on MBXSVR01, includes the error information, and formats the output to a list.

```
Test-AssistantHealth -ServerName MBXSVR01 -IncludeCrashDump
-ResolveProblems | Format-List
```

EXAMPLE 2

This example detects the mailbox assistant's health on the local Mailbox server. The *MaxProcessingTimeInMinutes* parameter specifies 30 minutes as the maximum amount of time the service is allowed to process an event without responding, and formats the output to a list.

```
Test-AssistantHealth -MaxProcessingTimeInMinutes 30 |
Format-List
```

Detailed Description

The Mailbox Assistants service runs on all servers that have the Mailbox server role installed. This service is responsible for scheduling and dispatching several assistants that ensure mailboxes function correctly.

By default, when you run this cmdlet, it returns the *RunspaceId*, events, and performance counters in a table format.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Assistants" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
IncludeCrashDump	Optional	System.Management.Automation.SwitchParameter	The *IncludeCrashDump* parameter specifies that the command should take an error report prior to taking any recovery actions. This parameter should only be used if running from a local computer. If you use the parameter while connected remotely, the command fails. The default value for this parameter is `$false`. You don't have to specify a value with this parameter.
MaxProcessingTimeInMinutes	Optional	System.UInt32	The *MaxProcessingTimeInMinutes* parameter specifies the maximum amount of

			time the MSExchangeMailboxAssistants service is allowed to process an event without responding. You can specify a value from 1 through 3600 minutes. The default value is 15 minutes.
MonitoringContext	Optional	System.Management.Automation.SwitchParameter	The *MonitoringContext* switch includes the associated monitoring events and performance counters in the results. You don't need to specify a value with this switch. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
ResolveProblems	Optional	System.Management.Automation.SwitchParameter	This *ResolveProblems* parameter specifies that if the command detects an issue, it attempts to fix it. This command attempts to fix the following issues: • Starts the Mailbox

			Assistants service if it isn't running. • Restarts the Mailbox Assistants service if it detects that the service is hung or deadlocked for more than 15 minutes. You don't have to specify a value with this parameter.
ServerName	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *ServerName* parameter specifies the identity of the Mailbox server on which the mailbox assistant that's being tested resides. If this parameter isn't specified, the command runs on the local server. If the local server isn't a Mailbox server, the command fails.
WatermarkBehindWarningThreholdInMinutes	Optional	System.UInt32	The *WatermarkBehindWarningThreholdInMinutes* parameter specifies the threshold for watermark age. Event watermarks indicate the last time that events were successfully processed by an assistant.

			An event watermark that hasn't been updated in a while may indicate a problem. For each Mailbox Assistant, the **Test-AssistantHealth** cmdlet compares the current time with the time stamp of the last event watermark to determine the watermark age. If that age exceeds the value set by the *WatermarkBehindWarningThreholdInMinutes* parameter, a warning is generated. You can specify a value from 1 through 10080 minutes. The default value is 60 minutes.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxServer

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox server cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MailboxServer** cmdlet to return a Mailbox server object and all its attributes. If no parameter is specified, a complete list of the Mailbox servers in the entire organization is returned.

```
Get-MailboxServer [-Identity <MailboxServerIdParameter>] [-
DomainController <Fqdn>] [-Status <SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves all the Mailbox servers in the organization.

```
Get-MailboxServer
```

EXAMPLE 2

This example retrieves the specific server instance Server1.

```
Get-MailboxServer -Identity Server1
```

Detailed Description

To view all the Mailbox server attributes that this cmdlet returns, you must pipe the command to the **Format-List** cmdlet.

The **ExchangeVersion** attribute returned is the minimum version of Microsoft Exchange that you can use to manage the returned object. This attribute isn't the same as the version of Microsoft Exchange that's displayed in the Exchange Administration Center when you select **Server Configuration**.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox server configuration" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxServerIdParameter	The *Identity* parameter specifies the Mailbox server. You can use the following values: • Name • GUID • Distinguished name (DN)
Status	Optional	System.Management.Automation.SwitchParameter	The *Status* parameter specifies whether you want to get additional

			status information, such as locale. You don't need to specify a value with this parameter. If you specify this parameter, you should format the output in such a way that you can view the additional attributes, for example, pipe the output to the **Format-List** cmdlet.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxServer

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox server cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-MailboxServer** cmdlet to modify the configuration settings and attributes of an Exchange 2013 Mailbox server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxServer -Identity <MailboxServerIdParameter> [-
AutoDagServerConfigured <$true | $false>] [-AutoDatabaseMountDial
<Lossless | GoodAvailability | BestAvailability>] [-
CalendarRepairIntervalEndWindow <Int32>] [-
CalendarRepairLogDirectorySizeLimit <Unlimited>] [-
CalendarRepairLogEnabled <$true | $false>] [-CalendarRepairLogFileAgeLimit
<EnhancedTimeSpan>] [-CalendarRepairLogPath <LocalLongFullPath>] [-
CalendarRepairLogSubjectLoggingEnabled <$true | $false>] [-
CalendarRepairMissingItemFixDisabled <$true | $false>] [-
CalendarRepairMode <ValidateOnly | RepairAndValidate>] [-
CalendarRepairWorkCycle <EnhancedTimeSpan>] [-
CalendarRepairWorkCycleCheckpoint <EnhancedTimeSpan>] [-Confirm
[<SwitchParameter>]] [-DarTaskStoreTimeBasedAssistantWorkCycle
<EnhancedTimeSpan>] [-DarTaskStoreTimeBasedAssistantWorkCycleCheckpoint
<EnhancedTimeSpan>] [-DatabaseCopyActivationDisabledAndMoveNow <$true |
$false>] [-DatabaseCopyAutoActivationPolicy <Unrestricted | IntrasiteOnly
| Blocked>] [-DirectoryProcessorWorkCycle <EnhancedTimeSpan>] [-
DirectoryProcessorWorkCycleCheckpoint <EnhancedTimeSpan>] [-
DomainController <Fqdn>] [-FaultZone <String>] [-
FolderLogForManagedFoldersEnabled <$true | $false>] [-
ForceGroupMetricsGeneration <$true | $false>] [-GroupMailboxWorkCycle
<EnhancedTimeSpan>] [-GroupMailboxWorkCycleCheckpoint <EnhancedTimeSpan>]
[-InferenceDataCollectionWorkCycle <EnhancedTimeSpan>] [-
InferenceDataCollectionWorkCycleCheckpoint <EnhancedTimeSpan>] [-
InferenceTrainingWorkCycle <EnhancedTimeSpan>] [-
InferenceTrainingWorkCycleCheckpoint <EnhancedTimeSpan>] [-
IsExcludedFromProvisioning <$true | $false>] [-
JournalingLogForManagedFoldersEnabled <$true | $false>] [-
JunkEmailOptionsCommitterWorkCycle <EnhancedTimeSpan>] [-Locale
<MultiValuedProperty>] [-LogDirectorySizeLimitForManagedFolders
<Unlimited>] [-LogFileAgeLimitForManagedFolders <EnhancedTimeSpan>] [-
LogFileSizeLimitForManagedFolders <Unlimited>] [-LogPathForManagedFolders
<LocalLongFullPath>] [-MailboxAssociationReplicationWorkCycle
<EnhancedTimeSpan>] [-MailboxAssociationReplicationWorkCycleCheckpoint
<EnhancedTimeSpan>] [-MailboxProcessorWorkCycle <EnhancedTimeSpan>] [-
ManagedFolderAssistantSchedule <ScheduleInterval[]>] [-
ManagedFolderWorkCycle <EnhancedTimeSpan>] [-
ManagedFolderWorkCycleCheckpoint <EnhancedTimeSpan>] [-
MAPIEncryptionRequired <$true | $false>] [-MaximumActiveDatabases <Int32>]
[-MaximumPreferredActiveDatabases <Int32>] [-MaxTransportSyncDispatchers
<Int32>] [-MigrationLogFilePath <LocalLongFullPath>] [-
MigrationLogLoggingLevel <None | Error | Warning | Information | Verbose |
Instrumentation>] [-MigrationLogMaxAge <EnhancedTimeSpan>] [-
MigrationLogMaxDirectorySize <ByteQuantifiedSize>] [-
MigrationLogMaxFileSize <ByteQuantifiedSize>] [-OABGeneratorWorkCycle
<EnhancedTimeSpan>] [-OABGeneratorWorkCycleCheckpoint <EnhancedTimeSpan>]
[-PeopleCentricTriageWorkCycle <EnhancedTimeSpan>] [-
PeopleCentricTriageWorkCycleCheckpoint <EnhancedTimeSpan>] [-
PeopleRelevanceWorkCycle <EnhancedTimeSpan>] [-
PeopleRelevanceWorkCycleCheckpoint <EnhancedTimeSpan>] [-
ProbeTimeBasedAssistantWorkCycle <EnhancedTimeSpan>] [-
ProbeTimeBasedAssistantWorkCycleCheckpoint <EnhancedTimeSpan>] [-
PublicFolderWorkCycle <EnhancedTimeSpan>] [-
PublicFolderWorkCycleCheckpoint <EnhancedTimeSpan>] [-
RetentionLogForManagedFoldersEnabled <$true | $false>] [-
SearchIndexRepairTimeBasedAssistantWorkCycle <EnhancedTimeSpan>] [-
SearchIndexRepairTimeBasedAssistantWorkCycleCheckpoint <EnhancedTimeSpan>]
[-SharePointSignalStoreWorkCycle <EnhancedTimeSpan>] [-
SharePointSignalStoreWorkCycleCheckpoint <EnhancedTimeSpan>] [-
SharingPolicySchedule <ScheduleInterval[]>] [-SharingPolicyWorkCycle
<EnhancedTimeSpan>] [-SharingPolicyWorkCycleCheckpoint <EnhancedTimeSpan>]
[-SharingSyncWorkCycle <EnhancedTimeSpan>] [-
SharingSyncWorkCycleCheckpoint <EnhancedTimeSpan>] [-SiteMailboxWorkCycle
<EnhancedTimeSpan>] [-SiteMailboxWorkCycleCheckpoint <EnhancedTimeSpan>]
[-StoreDsMaintenanceWorkCycle <EnhancedTimeSpan>] [-
StoreDsMaintenanceWorkCycleCheckpoint <EnhancedTimeSpan>] [-
StoreIntegrityCheckWorkCycle <EnhancedTimeSpan>] [-
StoreIntegrityCheckWorkCycleCheckpoint <EnhancedTimeSpan>] [-
StoreMaintenanceWorkCycle <EnhancedTimeSpan>] [-
StoreMaintenanceWorkCycleCheckpoint <EnhancedTimeSpan>] [-
StoreScheduledIntegrityCheckWorkCycle <EnhancedTimeSpan>] [-
StoreScheduledIntegrityCheckWorkCycleCheckpoint <EnhancedTimeSpan>] [-
StoreUrgentMaintenanceWorkCycle <EnhancedTimeSpan>] [-
StoreUrgentMaintenanceWorkCycleCheckpoint <EnhancedTimeSpan>] [-
SubjectLogForManagedFoldersEnabled <$true | $false>] [-
SubmissionServerOverrideList <MultiValuedProperty>] [-TopNWorkCycle
<EnhancedTimeSpan>] [-TopNWorkCycleCheckpoint <EnhancedTimeSpan>] [-
```

```
TransportSyncDispatchEnabled <$true | $false>] [-TransportSyncLogEnabled
<$true | $false>] [-TransportSyncLogFilePath <LocalLongFullPath>] [-
TransportSyncLogLoggingLevel <None | Error | Information | Verbose |
RawData | Debugging>] [-TransportSyncLogMaxAge <EnhancedTimeSpan>] [-
TransportSyncLogMaxDirectorySize <ByteQuantifiedSize>] [-
TransportSyncLogMaxFileSize <ByteQuantifiedSize>] [-
TransportSyncMailboxHealthLogEnabled <$true | $false>] [-
TransportSyncMailboxHealthLogFilePath <LocalLongFullPath>] [-
TransportSyncMailboxHealthLogMaxAge <EnhancedTimeSpan>] [-
TransportSyncMailboxHealthLogMaxDirectorySize <ByteQuantifiedSize>] [-
TransportSyncMailboxHealthLogMaxFileSize <ByteQuantifiedSize>] [-
UMReportingWorkCycle <EnhancedTimeSpan>] [-UMReportingWorkCycleCheckpoint
<EnhancedTimeSpan>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example throttles the Calendar Repair Assistant to detect and repair calendar inconsistencies for the Mailbox server MBX02 in a 7-day period. During that 7-day period, all mailboxes will be scanned, and at the end of the period, the process will start over.

```
Set-MailboxServer -Identity MBX02 -CalendarRepairWorkCycle
7.00:00:00 -CalendarRepairWorkCycleCheckpoint 7.00:00:00
```

EXAMPLE 2

This example throttles the Managed Folder Assistant, which applies message retention settings to all mailboxes for the Mailbox server MBX02 in a 10-day period. During that 10-day period, all mailboxes will be scanned, and at the end of the period, the process will start over.

```
Set-MailboxServer -Identity MBX02 -ManagedFolderWorkCycle
10.00:00:00 -ManagedFolderWorkCycleCheckpoint 10.00:00:00
```

EXAMPLE 3

This example throttles the Sharing Policy and Sharing Sync Assistants to apply sharing policies, sync shared calendars, and free/busy information for the mailboxes on server MBX02 in a 7-day period. During that 7-day period, all mailboxes will be scanned, and at the end of the period, the process will start over.

```
Set-MailboxServer -Identity MBX02 -SharingPolicyWorkCycle
7.00:00:00 -SharingPolicyWorkCycleCheckpoint 7.00:00:00 -
SharingSyncWorkCycle 7.00:00:00 -
SharingSyncWorkCycleCheckpoint 7.00:00:00
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Server Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxServerIdParameter	The *Identity* parameter specifies the Mailbox server. You can use the following values: • GUID • Distinguished name (DN) • Name of a Mailbox server
AutoDagServerConfigured	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
AutoDatabaseMountDial	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AutoDatabaseMountDial	The *AutoDatabaseMountDial* parameter specifies the automatic database mount behavior for a continuous replication environment after a database failover. You can use the following values: • `BestAvailability` If you specify this value, the database automatically mounts immediately after a failover if the copy queue length is less than or equal to 12. The copy queue length is the

			number of logs recognized by the passive copy that needs to be replicated. If the copy queue length is more than 12, the database doesn't automatically mount. When the copy queue length is less than or equal to 12, Exchange attempts to replicate the remaining logs to the passive copy and mounts the database. • `GoodAvailability` If you specify this value, the database automatically mounts immediately after a failover if the copy queue length is less than or equal to six. The copy queue length is the number of logs recognized by the passive copy that needs to be replicated. If the copy queue length is more than six, the database doesn't automatically mount. When the copy queue length is less than or equal to six, Exchange attempts to replicate the remaining logs to the passive copy and mounts the database. • `Lossless` If you specify this value, the database doesn't automatically mount until all logs that were generated on the

			active copy have been copied to the passive copy. This setting also causes Active Manager's best copy selection algorithm to sort potential candidates for activation based on the database copy's activation preference value and not its copy queue length. The default value is `GoodAvailability`. If you specify either `BestAvailability` or `GoodAvailability`, and all of the logs from the active copy haven't been replicated to the passive copy, you may lose some mailbox data. However, the transport dumpster feature, (which is enabled by default) helps protect against data loss by resubmitting messages that are in the transport dumpster queue.
CalendarRepairIntervalEndWindow	Optional	System.Int32	The *CalendarRepairIntervalEndWindow* parameter specifies the number of days into the future to repair calendars. For

			example, if this parameter is set to 90, the Calendar Repair Assistant repairs calendars on this Mailbox server 90 days from now. The default value is 30 days.
CalendarRepairLogDirectorySizeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CalendarRepairLogDirectorySizeLimit* parameter specifies the size limit for all log files for the Calendar Repair Assistant. After the limit is reached, the oldest files are deleted. The maximum size of the calendar repair log directory is calculated as the total size of all log files that have the same name prefix. Other files that don't follow the name prefix convention aren't counted in the total directory size calculation. Renaming old log files or copying other files into the calendar repair log directory could cause the directory to exceed its specified maximum size.

			Calendar repair log files for the Mailbox server role begin with the name prefix CRA. The default value is `unlimited`. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes.
CalendarRepairLogEnabled	Optional	System.Boolean	The *CalendarRepairLogEnabled* parameter specifies whether the Calendar Repair Attendant logs items that it repairs. The repair log doesn't contain failed repair attempts. The default value is `$true`.
CalendarRepairLogFileAgeLimit	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *CalendarRepairLogFileAgeLimit* parameter specifies how long to retain calendar repair logs. Log files that exceed the

			maximum retention period are deleted. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify a 30-day interval, enter 30.00:00:00. The default value is 00.00:00:00, which specifies that there's no limit on file retention (and not that files are overwritten immediately).
CalendarRepairLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *CalendarRepairLogPath* parameter specifies the path to the directory that stores the calendar repair log files. The default value is <*Exchange installation path*>v15\Logging\Calendar Repair Assistant.
CalendarRepairLogSubjectLoggingEnabled	Optional	System.Boolean	The *CalendarRepairLogSubjectLoggingEnabled* parameter specifies that the subject of the repaired

			calendar item is logged in the calendar repair log. The default value is `$true`.
CalendarRepairMissingItemFixDisabled	Optional	System.Boolean	The *CalendarRepairMissingItemFixDisabled* parameter specifies that the Calendar Repair Assistant won't fix missing attendee calendar items for mailboxes homed on this Mailbox server. The default value is `$false`.
CalendarRepairMode	Optional	Microsoft.Exchange.Data.CalendarRepairType	The *CalendarRepairMode* parameter specifies the mode that the Calendar Repair Assistant will run in.
CalendarRepairWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *CalendarRepairWorkCycle* parameter specifies the time span in which all mailboxes on the specified server will be scanned by the Calendar Repair Assistant. Calendars that have inconsistencies will be flagged and repaired according to the interval specified by the *CalendarRepairWorkCycleCheckpoint* parameter.

			To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 7 days for this parameter, use 07.00:00:00. The Calendar Repair Assistant will process all mailboxes on this server every 7 days.
CalendarRepairWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *CalendarRepairWorkCycleCheckpoint* parameter specifies the time span at which all mailboxes will be identified as needing work completed on them. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 1 day for this parameter, use 1.00:00:00. The Calendar Repair Assistant will process all mailboxes on this server every day.
Confirm	Optional	System.Management.	The *Confirm* switch

		Automation.SwitchParameter	causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DarTaskStoreTimeBasedAssistantWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
DarTaskStoreTimeBasedAssistantWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
DatabaseCopyActivationDisabledAndMoveNow	Optional	System.Boolean	The *DatabaseCopyActivationDisabledAndMoveNow* parameter specifies whether to prevent databases from being mounted on this server if there are other healthy copies of the databases on other servers. It will also immediately move any mounted databases on the server to other servers if copies exist and are healthy. Setting this parameter won't cause databases to move to a

			server that has the *DatabaseCopyAutoActivationPolicy* parameter set to `Blocked`.
DatabaseCopyAutoActivationPolicy	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DatabaseCopyAutoActivationPolicyType	The *DatabaseCopyAutoActivationPolicy* parameter specifies the type of automatic activation available for mailbox database copies on the specified Mailbox server(s). Values for this parameter are: • `Blocked` Databases can't be automatically activated on the specified Mailbox server(s). In addition, this stops server locator requests to the specified server, which prevents access to manually activated databases on the server if all DAG members are configured with a value of blocked. • `IntrasiteOnly` The database copy is allowed to be activated on Mailbox servers in the same Active Directory site. This prevents cross-site failover and activation. • `Unrestricted` There are no special restrictions on activating

			mailbox database copies on the specified Mailbox server(s).
DirectoryProcessorWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
DirectoryProcessorWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
FaultZone	Optional	System.String	This parameter is reserved for internal Microsoft use.
FolderLogForManagedFoldersEnabled	Optional	System.Boolean	The *FolderLogForManagedFoldersEnabled* parameter specifies whether the folder log for managed folders is enabled for messages that were moved to managed folders. The two possible values for this parameter are `$true` or `$false`. If you specify `$true`, information about folders that have managed folder mailbox policies applied to them is

			logged.
ForceGroupMetricsGeneration	Optional	System.Boolean	The *ForceGroupMetricsGeneration* parameter specifies that group metrics information must be generated on the Mailbox server regardless of whether that server generates an offline address book (OAB). By default, group metrics are generated only on servers that generate OABs. Group metrics information is used by MailTips to inform senders about how many recipients their messages will be sent to. You need to use this parameter if your organization doesn't generate OABs and you want the group metrics data to be available.
GroupMailboxWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
GroupMailboxWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
InferenceDataCollectionWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.

InferenceDataCollectionWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
InferenceTrainingWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
InferenceTrainingWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
IsExcludedFromProvisioning	Optional	System.Boolean	The *IsExcludedFromProvisioning* parameter specifies that the Mailbox server isn't considered by the OAB provisioning load balancer. If the *IsExcludedFromProvisioning* parameter is set to `$true`, the server won't be used for provisioning a new OAB or for moving existing OABs.
JournalingLogForManagedFoldersEnabled	Optional	System.Boolean	The *JournalingLogForManagedFoldersEnabled* parameter specifies whether the log for managed folders is enabled for journaling. The two possible values for this parameter are `$true` or `$false`. If you specify `$true`, information

			about messages that were journaled is logged. The logs are located at the location you specify with the *LogPathForManagedFolders* parameter.
JunkEmailOptionsCommitterWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
Locale	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Locale* parameter specifies the locale. A locale is a collection of language-related user preferences such as writing system, calendar, and date format. The following are examples: • en-US (English - United States) • de-AT (German - Austria) • es-CL (Spanish - Chile) For more information, see CultureInfo Class.
LogDirectorySizeLimitForManagedFolders	Optional	Microsoft.Exchange.Data.Unlimited	The *LogDirectorySizeLimitForManagedFolders* parameter specifies the size limit for all managed folder log files from a single message database.

			After the limit is reached for a set of managed folder log files from a message database, the oldest files are deleted to make space for new files. The size of the managed folder log files is calculated as the total size of all log files that have the same name prefix. For example, for a file with the name Managed_Folder_Assistant[Mailbox Database 01] 20061018-1.log, the prefix is Managed_Folder_Assistant[Mailbox Database 01]. If you rename log files or copy other files into the managed folder log directory, these files aren't counted in the log files size calculation. The managed folder log files for each message database have a unique name prefix. Therefore, this limit applies to the log files for each message database, and not to all the log files in the

			directory. If you have more than one message database, the maximum size of the managed folder log directory isn't the size specified in the *LogDirectorySizeLimitForManagedFolders* parameter because the managed folder log files generated by different databases have different name prefixes. The maximum size of the managed folder log directory is *X* times the specified value if you have *X* message databases. You must specify either an integer or unlimited. The default value is `unlimited`. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes.

LogFileAgeLimitForManagedFolders	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *LogFileAgeLimitForManagedFolders* parameter specifies how long to retain managed folder logs. Log files that exceed the maximum retention period are deleted. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify a 30-day interval, enter 30.00:00:00. The default value is 00.00:00:00, which specifies that there's no limit on file retention (and not that files are overwritten immediately).
LogFileSizeLimitForManagedFolders	Optional	Microsoft.Exchange.Data.Unlimited	The *LogFileSizeLimitForManagedFolders* parameter specifies the maximum size for each managed folder log file. When the log file size limit is reached, a new log file is created. The default value is 10 megabytes (MB).

			When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes.
LogPathForManagedFolders	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *LogPathForManagedFolders* parameter specifies the path to the directory that stores the managed folder log files. The default value is <*Exchange installation path*>v15\Logging\ Managed Folder Assistant.
MailboxAssociationReplicationWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
MailboxAssociationReplicationWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
MailboxProcessorWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *MailboxProcessorWorkCycle* parameter specifies how often to scan for locked mailboxes. The default value is 1 day.

ManagedFolderAssistantSchedule	Optional	Microsoft.Exchange.Common.ScheduleInterval[]	The *ManagedFolderAssistantSchedule* parameter specifies the intervals each week during which the Managed Folder Assistant applies messaging records management (MRM) settings to managed folders. The format is *StartDay.Time-EndDay.Time*. You can use the following values for the start and end days: • Full name of the day • Abbreviated name of the day • Integer from 0 through 6, where 0 = Sunday The start time and end time must be at least 15 minutes apart. Minutes are rounded down to 0, 15, 30, or 45. If you specify more than one interval, there must be at least 15 minutes between each interval. The following are examples: • "Sun.11:30 PM-Mon.1:30 AM"

			• 6.22:00-6.22:15 (The assistant will run from Saturday at 10:00 PM until Saturday at 10:15 PM.) • "Monday.4:30 AM-Monday.5:30 AM","Wednesday.4:30 AM-Wednesday.5:30 AM" (The assistant will run on Monday and Wednesday mornings from 4:30 until 5:30.) • "Sun.1:15 AM-Monday.23:00" If the Managed Folder Assistant doesn't finish processing the mailboxes on the server during the time that you've scheduled, it automatically resumes processing where it left off the next time it runs.
ManagedFolderWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ManagedFolderWorkCycle* parameter specifies the time span in which all mailboxes on the specified server will be processed by the Managed Folder Assistant. The Managed Folder Assistant applies

			retention policies according to the *ManagedFolderWorkCycleCheckpoint* interval. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 7 days for this parameter, use 07.00:00:00. The Managed Folder Assistant will process all mailboxes on this server every 7 days.
ManagedFolderWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ManagedFolderWorkCycleCheckpoint* parameter specifies the time span at which to refresh the list of mailboxes so that new mailboxes that have been created or moved will be part of the work queue. Also, as mailboxes are prioritized, existing mailboxes that haven't been successfully processed for a long time will be placed higher in the queue and will have a

			greater chance of being processed again in the same work cycle. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 24 hours for this parameter, use 24:00:00.
MAPIEncryptionRequired	Optional	System.Boolean	The *MAPIEncryptionRequired* parameter specifies whether Exchange blocks MAPI clients that don't use encrypted remote procedure calls (RPCs). The two possible values for this parameter are `$true` or `$false`.
MaximumActiveDatabases	Optional	System.Int32	The *MaximumActiveDatabases* parameter specifies the number of databases that can be mounted on this Mailbox server. This parameter accepts numeric values. When the maximum number is reached, the

			database copies on the server won't be activated if a failover or switchover occurs. If the copies are already active on a server, the Information Store on the server won't allow databases to be mounted.
MaximumPreferredActiveDatabases	Optional	System.Int32	The *MaximumPreferredActiveDatabases* parameter specifies a preferred maximum number of databases that a server should have. This value is different from the actual maximum, which is configured using the *MaximumActiveDatabases* parameter. The value of *MaximumPreferredActiveDatabases* is only honored during best copy and server selection, database and server switchovers, and when rebalancing the DAG.
MaxTransportSyncDispatchers	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
MigrationLogFilePath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.

MigrationLogLoggingLevel	Optional	Microsoft.Exchange.Data.MigrationEventType	This parameter is reserved for internal Microsoft use.
MigrationLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
MigrationLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
MigrationLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
OABGeneratorWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *OABGeneratorWorkCycle* parameter specifies the time span in which the OAB generation on the specified server will be processed. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 7 days for this parameter, use 07.00:00:00.
OABGeneratorWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *OABGeneratorWorkCycleCheckpoint* parameter specifies the time span at which to run OAB generation. To specify a value, enter it

			as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 24 hours for this parameter, use 24:00:00.
PeopleCentricTriageWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
PeopleCentricTriageWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
PeopleRelevanceWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
PeopleRelevanceWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
ProbeTimeBasedAssistantWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
ProbeTimeBasedAssistantWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
PublicFolderWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *PublicFolderWorkCycle* parameter is used by the public folder assistant to determine how often the mailboxes in a database are processed by the assistant.

PublicFolderWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *PublicFolderWorkCycleCheckpoint* determines how often the mailbox list for a database is evaluated. The processing speed is also calculated.
RetentionLogForManagedFoldersEnabled	Optional	System.Boolean	The *RetentionLogForManagedFoldersEnabled* parameter specifies whether the Managed Folder Assistant logs information about messages that have reached their retention limits. The two possible values for this parameter are `$true` or `$false`. If you specify `$true`, information about messages that have been processed because they have reached their retention limits is logged.
SearchIndexRepairTimeBasedAssistantWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
SearchIndexRepairTimeBasedAssistantWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
SharePointSignalStoreWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.

SharePointSignalStoreWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
SharingPolicySchedule	Optional	Microsoft.Exchange.Common.ScheduleInterval[]	The *SharingPolicySchedule* parameter specifies the intervals each week during which the sharing policy runs. The Sharing Policy Assistant checks permissions on shared calendar items and contact folders in users' mailboxes against the assigned sharing policy. The assistant lowers or removes permissions according to the policy. The format is *StartDay.Time-EndDay.Time*. You can use the following values for the start and end days: • Full name of the day • Abbreviated name of the day • Integer from 0 through 6, where 0 = Sunday The start time and end time must be at least 15 minutes apart. Minutes are rounded down to 0, 15, 30, or 45. If you

			specify more than one interval, there must be at least 15 minutes between each interval. The following are examples: • "Sun.11:30 PM-Mon.1:30 AM" • 6.22:00-6.22:15 (The assistant will run from Saturday at 10:00 PM until Saturday 10:15 PM.)
SharingPolicyWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *SharingPolicyWorkCycle* parameter specifies the time span in which all mailboxes on the specified server will be scanned by the Sharing Policy Assistant. The Sharing Policy Assistant scans all mailboxes and enables or disables sharing polices according to the interval specified by the *SharingPolicyWorkCycle*. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.

			For example, if you specify 7 days for this parameter, use 07.00:00:00. The Sharing Policy Assistant will process all mailboxes on this server every 7 days.
SharingPolicyWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *SharingPolicyWorkCycleCheckpoint* parameter specifies the time span at which to refresh the list of mailboxes so that new mailboxes that have been created or moved will be part of the work queue. Also, as mailboxes are prioritized, existing mailboxes that haven't been successfully processed for a long time will be placed higher in the queue and will have a greater chance of being processed again in the same work cycle. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify

			1 day for this parameter, use 1.00:00:00. The Sharing Policy Assistant will process all mailboxes on this server every day.
SharingSyncWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *SharingSyncWorkCycle* parameter specifies the time span in which all mailboxes on the specified server will be synced to the cloud-based service by the Sharing Sync Assistant. Mailboxes that require syncing will be synced according to the interval specified by the *SharingSyncWorkCycleCheckpoint* parameter. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 7 days for this parameter, use 07.00:00:00. The Sharing Sync Assistant will process all mailboxes on this server every 7 days.

SharingSyncWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *SharingSyncWorkCycleCheckpoint* parameter specifies the time span at which to refresh the list of mailboxes so that new mailboxes that have been created or moved will be part of the work queue. Also, as mailboxes are prioritized, existing mailboxes that haven't been successfully processed for a long time will be placed higher in the queue and will have a greater chance of being processed again in the same work cycle. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 1 day for this parameter, use 1.00:00:00. The Sharing Sync Assistant will process all mailboxes on this server every day.
SiteMailboxWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *SiteMailboxWorkCycle*

			parameter specifies the time span in which the site mailbox information on the specified server will be processed. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 7 days for this parameter, use 07.00:00:00.
SiteMailboxWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *SiteMailboxWorkCycleCheckpoint* parameter specifies the time span at which to refresh the site mailbox workcycle. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 24 hours for this parameter, use 24:00:00.
StoreDsMaintenanceWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
StoreDsMaintenance	Optional	Microsoft.Exchange.Da	This parameter is reserved

WorkCycleCheckpoint		ta.EnhancedTimeSpan	for internal Microsoft use.
StoreIntegrityCheckWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
StoreIntegrityCheckWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
StoreMaintenanceWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
StoreMaintenanceWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
StoreScheduledIntegrityCheckWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
StoreScheduledIntegrityCheckWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
StoreUrgentMaintenanceWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
StoreUrgentMaintenanceWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
SubjectLogForManagedFoldersEnabled	Optional	System.Boolean	The *SubjectLogForManagedFoldersEnabled* parameter specifies whether the subject of messages is displayed in managed folder logs. The two possible values for this parameter are `$true` or

			`$false`. If you specify `$false`, the subject of messages is blank in the managed folder logs. The default value is `$false`.
SubmissionServerOverrideList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
TopNWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *TopNWorkCycle* parameter specifies the time span in which all mailboxes that have Unified Messaging on the specified server will be scanned by the TopN Words Assistant. The TopN Words Assistant scans voice mail for the most frequently used words to aid in transcription. The most common words are then indexed according to the interval specified by the *TopNWorkCycleCheckpoint* parameter. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.

			For example, if you specify 7 days for this parameter, use 07.00:00:00. The TopN Words Assistant will process all mailboxes on which Unified Messaging is enabled on this server every 7 days.
TopNWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *TopNWorkCycleCheckpoint* parameter specifies the time span at which to refresh the list of mailboxes so that new mailboxes that have been created or moved will be part of the work queue. Also, as mailboxes are prioritized, existing mailboxes that haven't been successfully processed for a long time will be placed higher in the queue and will have a greater chance of being processed again in the same work cycle. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.

			For example, if you specify 1 day for this parameter, use 1.00:00:00. The TopN Words Assistant will process all mailboxes on this server every day.
TransportSyncDispatchEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncLogFilePath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
TransportSyncLogLoggingLevel	Optional	Microsoft.Exchange.Data.SyncLoggingLevel	This parameter is reserved for internal Microsoft use.
TransportSyncLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
TransportSyncLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
TransportSyncMailboxHealthLogEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
TransportSyncMailboxHealthLogFilePath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	This parameter is reserved for internal Microsoft use.
TransportSyncMailboxHealthLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	This parameter is reserved for internal Microsoft use.
TransportSyncMailbox	Optional	Microsoft.Exchange.Da	This parameter is reserved

HealthLogMaxDirectorySize		ta.ByteQuantifiedSize	for internal Microsoft use.
TransportSyncMailboxHealthLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is reserved for internal Microsoft use.
UMReportingWorkCycle	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *UMReportingWorkCycle* parameter specifies the time span in which the arbitration mailbox named SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9} on the specified server will be scanned by the Unified Messaging Reporting Assistant. The Unified Messaging Reporting Assistant updates the Call Statistics reports by reading Unified Messaging call data records for an organization on a regular basis. By default, it's scheduled to run once every 24 hours. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds.

			For example, if you specify 7 days for this parameter, use 07.00:00:00. The Unified Messaging Reporting Assistant will process all mailboxes that have Unified Messaging enabled on this server every 7 days. **Note:** Changing the default work cycle for this assistant might impact the performance of the Mailbox server for your organization.
UMReportingWorkCycleCheckpoint	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *UMReportingWorkCycleCheckpoint* parameter specifies the time span at which the arbitration mailbox named SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9} will be marked by processing. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, if you specify 1 day for this parameter, use 1.00:00:00.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-MRSHealth

Exchange Management Shell > Exchange 2013 cmdlets > Mailbox server cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-MRSHealth** cmdlet to test the health of an instance of the Microsoft Exchange Mailbox Replication service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-MRSHealth [-Identity <ServerIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-MaxQueueScanAgeSeconds
```

```
<Int32>] [-MonitoringContext <$true | $false>] [-MRSProxyCredentials
<PSCredential>] [-MRSProxyServer <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the health of the Mailbox Replication service on all Mailbox servers.

```
Get-MailboxServer | Test-MRSHealth
```

EXAMPLE 2

This example tests the health of the Mailbox Replication service on the Mailbox server named MBX01.

```
Test-MRSHealth MBX01
```

Detailed Description

The Microsoft Exchange Mailbox Replication service runs on Mailbox servers. This command ensures that the Mailbox Replication service is running and that it responds to a remote procedure call (RPC) ping check.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox move and migration permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the server on which to perform the health test. You can use any value that uniquely identifies the server. For example: • Name • Distinguished name (DN) • `ExchangeLegacyDN` • GUID If you don't specify the server, the command runs on the local server.
MaxQueueScanAgeSeconds	Optional	System.Int32	The *MaxQueueScanAgeSeconds* parameter specifies the threshold for the **last queue scan** property. If the time stamp on the **last queue scan** property is older than the value

			specified by this parameter, an error event is created that shows the Mailbox Replication service isn't scanning mailbox database queues. The default value is 1800 seconds (30 minutes).
MonitoringContext	Optional	System.Boolean	The *MonitoringContext* parameter includes or excludes the associated monitoring events and performance counters in the results. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you specify the value `$true`, the monitoring events and performance counters are included in the command results. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
MRSProxyCredentials	Optional	System.Management.	The *MRSProxyCredentials*

		Automation.PSCredential	parameter specifies the credentials that are required for the **MRSProxyPingCheck** test on the server that's specified by the *MRSProxyServer* parameter. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
MRSProxyServer	Optional	Microsoft.Exchange.Data.Fqdn	The *MRSProxyServer* parameter specifies the fully qualified domain name (FQDN) of the target server for the **MRSProxyPingCheck** test. The Microsoft Replication proxy service is part of the Mailbox Replication service, and is used for remote mailbox moves. However, the Mailbox Replication proxy service communicates only with the Mailbox Replication

			service on another server. You can test the Mailbox Replication proxy service in the following ways: • If you specify an *MRSProxyServer* value, and you specify the source server by using the *Identity* parameter, the test is performed between that server and the target server specified by the *MRSProxyServer* parameter. • If you specify an *MRSProxyServer* value, and you don't specify a source server by using the *Identity* parameter, the test is performed between the local server and the target server specified by the *MRSProxyServer* parameter. • If you don't specify an *MRSProxyServer* value or an *Identity* value, the test is performed between the Mailbox Replication service and the Mailbox Replication

			proxy service on the local server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Move and migration cmdlets

Exchange Server 2013 > Exchange Management Shell > Exchange 2013 cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-08*

Mailbox migration cmdlets

Complete-MigrationBatch

Get-MigrationBatch

New-MigrationBatch

Remove-MigrationBatch

Set-MigrationBatch

Start-MigrationBatch

Stop-MigrationBatch

Get-MigrationConfig

Set-MigrationConfig

Get-MigrationEndpoint

New-MigrationEndpoint

Remove-MigrationEndpoint

Set-MigrationEndpoint

Export-MigrationReport

Test-MigrationServerAvailability

Get-MigrationStatistics

Get-MigrationUser

Remove-MigrationUser

Get-MigrationUserStatistics

Mailbox move cmdlets

Get-MoveRequest

New-MoveRequest

Remove-MoveRequest

Resume-MoveRequest

Set-MoveRequest

Suspend-MoveRequest

Get-MoveRequestStatistics

Public folder migration cmdlets

Get-PublicFolderMigrationRequest

New-PublicFolderMigrationRequest

Remove-PublicFolderMigrationRequest

Resume-PublicFolderMigrationRequest

Set-PublicFolderMigrationRequest

Suspend-PublicFolderMigrationRequest

Get-PublicFolderMigrationRequestStatistics

Public folder move cmdlets

Get-PublicFolderMoveRequest

New-PublicFolderMoveRequest

Remove-PublicFolderMoveRequest

Resume-PublicFolderMoveRequest

Set-PublicFolderMoveRequest

Suspend-PublicFolderMoveRequest

Get-PublicFolderMoveRequestStatistics

Complete-MigrationBatch

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Complete-MigrationBatch** cmdlet to finalize a migration batch for a local move, cross-forest move, or remote move migration that has successfully finished initial synchronization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Complete-MigrationBatch [-Identity <MigrationBatchIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-NotificationEmails
<MultiValuedProperty>] [-Organization <OrganizationIdParameter>] [-
Partition <MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example completes the migration batch LocalMove1 and sends a notification email message to the specified users.

```
Complete-MigrationBatch -Identity LocalMove1 -
NotificationEmails admin@contoso.com,lucio@contoso.com
```

Detailed Description

After a migration batch for a local or cross-forest move has successfully run and has a status state of Synced, use the **Complete-MigrationBatch** cmdlet to finalize the migration batch. Finalization is the last phase performed during a local or cross-forest move. When you finalize a migration batch, the cmdlet does the following for each mailbox in the migration batch:

- Runs a final incremental synchronization.
- Configures the user's Microsoft Outlook profile to point to the new target domain.
- Converts the source mailbox to a mail-enabled user in the source domain.

When the finalization process is complete, you can remove the batch by using the **Remove-MigrationBatch** cmdlet.

If a migration batch has a status of Completed with Errors, you can rerun the **Complete-MigrationBatch** cmdlet. The cmdlet will attempt to finalize the failed users.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox move and migration permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Management.Migration.MigrationBatchIdParameter	The *Identity* parameter identifies the name of the migration batch. The value for this parameter is specified by the *Name* parameter for the **New-MigrationBatch** cmdlet. Use the **Get-MigrationBatch** cmdlet to determine the value of this parameter for the migration batch.
NotificationEmails	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *NotificationEmails* parameter specifies one or more email addresses that status

			reports are sent to after the migration batch is completed. Specify the value as a string array and separate multiple email addresses with commas. If you don't use this parameter, the final status report is sent to the administrator who runs the **Complete-MigrationBatch** cmdlet.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MigrationBatch

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MigrationBatch** cmdlet to retrieve status information about the current migration batch.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MigrationBatch [-Identity <MigrationBatchIdParameter>] <COMMON
PARAMETERS>


Get-MigrationBatch [-Endpoint <MigrationEndpointIdParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: [-Diagnostic <SwitchParameter>] [-DiagnosticArgument
<String>] [-DomainController <Fqdn>] [-IncludeReport <SwitchParameter>] [-
Organization <OrganizationIdParameter>] [-Partition <MailboxIdParameter>]
[-Status <Created | Syncing | Stopping | Stopped | Completed | Failed |
Removing | Synced | IncrementalSyncing | Completing | CompletedWithErrors
| SyncedWithErrors | Corrupted | Waiting | Starting>]
```

Examples

EXAMPLE 1

This example displays status information for the migration batch LocalMove2.

```
Get-MigrationBatch -Identity LocalMove2
```

EXAMPLE 2

This example displays information about all migration batches associated with the migration endpoint exsrv1.contoso.com.

```
Get-MigrationBatch -Endpoint exsrv1.contoso.com
```

Detailed Description

The **Get-MigrationBatch** cmdlet displays status information about the current migration batch. This information includes the following information:

- Status of the migration batch
- Total number of mailboxes being migrated
- Number of successfully completed migrations
- Migration errors
- Date and time when the migration was started

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Diagnostic	Optional	System.Management.Automation.SwitchParameter	The *Diagnostic* parameter returns additional information that you can use to troubleshoot migration errors or send to Microsoft Customer Service and Support.
DiagnosticArgument	Optional	System.String	This parameter is reserved for internal Microsoft use.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Endpoint	Optional	Microsoft.Exchange.Management.Migration.MigrationEndpointIdParameter	The *Endpoint* parameter returns a list of migration batches associated with the specified migration endpoint. If you use this parameter, you can't include the *Identity* parameter.
Identity	Optional	Microsoft.Exchange.Management.Migration.MigrationBatchIdParameter	The *Identity* parameter identifies the name of the current migration batch. The value for this parameter is specified by the *Name* parameter of the **New-MigrationBatch** cmdlet. If you use this parameter, you can't include the *Endpoint* parameter.
IncludeReport	Optional	System.Management.Automation.SwitchPar	The *IncludeReport* parameter returns

		ameter	additional information about the specified migration batch. This information is displayed in the **Report** field.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
Status	Optional	Microsoft.Exchange.Data.Storage.Management.MigrationBatchStatus	The *Status* parameter returns a list of migration batches that have the specified status state. Use one of the following values: • `Completed` • `CompletedWithErrors` • `Completing` • `Corrupted` • `Created` • `Failed` • `IncrementalSyncing` • `Removing` • `Starting` • `Stopped` • `Syncing` • `Stopping` • `Synced` • `SyncedwithErrors` • `Waiting`

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MigrationBatch

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MigrationBatch** cmdlet to submit a new migration request for a batch of users. The cmdlet is used to move mailboxes in an Exchange on-premises organization, migrate on-premises mailboxes to Exchange Online (also called *onboarding*), or migrate Exchange Online mailboxes back to an on-premises Exchange organization (also called *offboarding*) in an Exchange hybrid deployment. You can also use this cmdlet to perform another type of onboarding migration, which is called an IMAP migration. In this type of migration, mailbox data is migrated from on-premises mailboxes on an IMAP server to Exchange Online mailboxes.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MigrationBatch [-ArchiveOnly <SwitchParameter>] [-BadItemLimit
<Unlimited>] [-CSVData <Byte[]>] [-DisallowExistingUsers
<SwitchParameter>] [-ExcludeFolders <MultiValuedProperty>] [-
LargeItemLimit <Unlimited>] [-PrimaryOnly <SwitchParameter>] [-
SourceEndpoint <MigrationEndpointIdParameter>] [-TargetArchiveDatabases
<MultiValuedProperty>] [-TargetDatabases <MultiValuedProperty>] [-
TargetDeliveryDomain <String>] <COMMON PARAMETERS>


New-MigrationBatch -CSVData <Byte[]> -Local <SwitchParameter> [-
ArchiveOnly <SwitchParameter>] [-BadItemLimit <Unlimited>] [-
DisallowExistingUsers <SwitchParameter>] [-PrimaryOnly <SwitchParameter>]
[-TargetArchiveDatabases <MultiValuedProperty>] [-TargetDatabases
<MultiValuedProperty>] <COMMON PARAMETERS>


New-MigrationBatch -CSVData <Byte[]> -XO1 <SwitchParameter> <COMMON
PARAMETERS>


New-MigrationBatch -CSVData <Byte[]> [-ArchiveOnly <SwitchParameter>] [-
BadItemLimit <Unlimited>] [-DisallowExistingUsers <SwitchParameter>] [-
LargeItemLimit <Unlimited>] [-PrimaryOnly <SwitchParameter>] [-
TargetArchiveDatabases <MultiValuedProperty>] [-TargetDatabases
<MultiValuedProperty>] [-TargetDeliveryDomain <String>] [-TargetEndpoint
<MigrationEndpointIdParameter>] <COMMON PARAMETERS>


New-MigrationBatch -UserIds <MigrationUserIdParameter[]> [-DisableOnCopy
<SwitchParameter>] <COMMON PARAMETERS>


New-MigrationBatch -Users <MigrationUser[]> [-DisableOnCopy
<SwitchParameter>] <COMMON PARAMETERS>
```

```
COMMON PARAMETERS: -Name <String> [-AllowIncrementalSyncs <$true |
$false>] [-AutoComplete <SwitchParameter>] [-AutoRetryCount <Int32>] [-
AutoStart <SwitchParameter>] [-CompleteAfter <DateTime>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Locale <CultureInfo>] [-
NotificationEmails <MultiValuedProperty>] [-Organization
<OrganizationIdParameter>] [-Partition <MailboxIdParameter>] [-
ReportInterval <TimeSpan>] [-SkipSteps <SkippableMigrationSteps[]>] [-
StartAfter <DateTime>] [-TimeZone <ExTimeZoneValue>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a migration batch for a local move, where the mailboxes in the specified CSV file are moved to a different mailbox database. This CSV file contains a single column with the email address for the mailboxes that will be moved. The header for this column must be named **EmailAddress**. The migration batch in this example must be started manually by using the **Start-MigrationBatch** cmdlet or the Exchange admin center. Alternatively, you can use the *AutoStart* parameter to start the migration batch automatically.

```
New-MigrationBatch -Local -Name LocalMove1 -CSVData
([System.IO.File]::ReadAllBytes("C:\Users\Administrator
\Desktop\LocalMove1.csv")) -TargetDatabases MBXDB2

Start-MigrationBatch -Identity LocalMove1
```

EXAMPLE 2

This example creates a migration batch for a cross-forest enterprise move, where the mailboxes for the mail users specified in the CSV file are moved to a different forest. A new migration endpoint is created, which identifies the domain where the mailboxes are currently located. The endpoint is used to create the migration batch. Then the migration batch is started with the **Start-MigrationBatch** cmdlet. Note that cross-forest moves are initiated from the target forest, which is the forest that you want to move the mailboxes to.

```
$Credentials = Get-Credential

$MigrationEndpointSource = New-MigrationEndpoint -
ExchangeRemoteMove -Name Forest1Endpoint -Autodiscover -
EmailAddress administrator@forest1.contoso.com -Credentials
$Credentials

$CrossForestBatch = New-MigrationBatch -Name
CrossForestBatch1 -SourceEndpoint
```

```
$MigrationEndpointSource.Identity -TargetDeliveryDomain
forest2.contoso.com -TargetDatabases MBXDB1 -CSVData
([System.IO.File]::ReadAllBytes("C:\Users\Administrator
\Desktop\CrossForestBatch1.csv"))

Start-MigrationBatch -Identity $CrossForestBatch.Identity
```

EXAMPLE 3

This example creates a migration batch for an onboarding remote move migration from an on-premises Exchange organization to Exchange Online. The syntax is similar to that of a cross-forest move, but it's initiated from the Exchange Online organization. A new migration endpoint is created, which points to the on-premises organization as the source location of the mailboxes that will be migrated. This endpoint is used to create the migration batch. Then the migration batch is started with the **Start-MigrationBatch** cmdlet.

```
$Credentials = Get-Credential

$MigrationEndpointOnPrem = New-MigrationEndpoint -
ExchangeRemoteMove -Name OnpremEndpoint -Autodiscover -
EmailAddress administrator@onprem.contoso.com -Credentials
$Credentials

$OnboardingBatch = New-MigrationBatch -Name
RemoteOnBoarding1 -SourceEndpoint
$MigrationEndpointOnprem.Identity -TargetDeliveryDomain
cloud.contoso.com -CSVData
([System.IO.File]::ReadAllBytes("C:\Users\Administrator
\Desktop\RemoteOnBoarding1.csv"))

Start-MigrationBatch -Identity $OnboardingBatch.Identity
```

EXAMPLE 4

This example creates a migration batch for an offboarding remote move migration from Exchange Online to an on-premises Exchange organization. Like an onboarding remote move, it's initiated from the Exchange Online organization. First a Migration Endpoint is created that contains information about how to connect to the on-premises organization. The endpoint is used as the TargetEndpoint when creating the migration batch, which is then started with the **Start-MigrationBatch** cmdlet. The *TargetDatabases* parameter specifies multiple on-premises databases that the migration service can select as the target database to move the mailbox to.

```
$Credentials = Get-Credential

$MigrationEndpointOnPrem = New-MigrationEndpoint -
ExchangeRemoteMove -Name OnpremEndpoint -Autodiscover -
EmailAddress administrator@onprem.contoso.com -Credentials
$Credentials

$OffboardingBatch = New-MigrationBatch -Name
RemoteOffBoarding1 -TargetEndpoint
$MigrationEndpointOnprem.Identity -TargetDeliveryDomain
onprem.contoso.com -TargetDatabases
@(MBXDB01,MBXDB02,MBXDB03) -CSVData
([System.IO.File]::ReadAllBytes("C:\Users\Administrator
\Desktop\RemoteOffBoarding1.csv"))

Start-MigrationBatch -Identity $OffboardingBatch.Identity
```

EXAMPLE 5

This example creates a migration batch for the cutover Exchange migration CutoverBatch that's automatically started. The example uses the **Test-MigrationServerAvailability** cmdlet to obtain and test the connection settings to the on-premises Exchange server, and then uses those connection settings to create a migration endpoint. The endpoint is then used to create the migration batch. This example also includes the optional *TimeZone* parameter.

```
$credentials = Get-Credential

$TSMA = Test-MigrationServerAvailability -
ExchangeOutlookAnywhere -Autodiscover -EmailAddress
administrator@contoso.com -Credentials $credentials

$SourceEndpoint = New-MigrationEndpoint -
ExchangeOutlookAnywhere -Name SourceEndpoint -
ConnectionSettings $TSMA.ConnectionSettings

New-MigrationBatch -Name CutoverBatch -SourceEndpoint
$SourceEndpoint.Identity -TimeZone "Pacific Standard Time"
-AutoStart
```

EXAMPLE 6

This example creates and starts a migration batch for a staged Exchange migration. The example uses the **New-MigrationEndpoint** cmdlet to create a migration endpoint for the on-premises Exchange server, and then uses that endpoint to create the migration batch. The migration batch is started with the **Start-MigrationBatch** cmdlet.

```
$Credentials = Get-Credential

$MigrationEndpoint = New-MigrationEndpoint -
ExchangeOutlookAnywhere -Name ContosoEndpoint -Autodiscover
-EmailAddress administrator@contoso.com -Credentials
$Credentials

$StagedBatch1 = New-MigrationBatch -Name StagedBatch1 -
SourceEndpoint $MigrationEndpoint.Identity -CSVData
([System.IO.File]::ReadAllBytes("C:\Users\Administrator
\Desktop\StagedBatch1.csv"))

Start-MigrationBatch -Identity $StagedBatch1.Identity
```

EXAMPLE 7

This example creates a migration endpoint for the connection settings to the IMAP server. Then an IMAP migration batch is created that uses the CSV migration file IMAPmigration_1.csv and excludes the contents of the Deleted Items and Junk Email folders. This migration batch is pending until it's started with the **Start-MigrationBatch** cmdlet.

```
New-MigrationEndpoint -IMAP -Name IMAPEndpoint1 -
RemoteServer imap.contoso.com -Port 993

New-MigrationBatch -Name IMAPbatch1 -CSVData
([System.IO.File]::ReadAllBytes("C:\Users\Administrator
\Desktop\IMAPmigration_1.csv")) -SourceEndpoint
IMAPEndpoint1 -ExcludeFolders "Deleted Items","Junk Email"
```

Detailed Description

Use the **New-MigrationBatch** cmdlet to create a migration batch to migrate mailboxes and mailbox data in one of the following migration scenarios.

Moves in on-premises Exchange organizations

- **Local move:** A local move is where you move mailboxes from one mailbox database to another. A local move occurs within a single forest. For more information, see Example 1.

- **Cross-forest enterprise move:** In a cross-forest enterprise move, mailboxes are moved to a different forest. Cross-forest moves are initiated either from the target forest, which is the forest that you want to move the mailboxes to, or from the source forest, which is the forest that currently hosts the mailboxes. For more information, see Example 2.

Onboarding and offboarding in Exchange Online

- **Onboarding remote move migration:** In a hybrid deployment, you can move mailboxes from an on-premises Exchange organization to Exchange Online. This is also known as an *onboarding* remote move migration because you on-board mailboxes to Exchange Online. For more information, see Example 3.
- **Offboarding remote move migration:** You can also perform an *offboarding* remote move migration, where you migrate Exchange Online mailboxes to your on-premises Exchange organization. For more information, see Example 4.

Note:
Both onboarding and offboarding remote move migrations are initiated from your Exchange Online organization.

- **Cutover Exchange migration:** This is another type of onboarding migration and is used to migrate all mailboxes in an on-premises Exchange organization to Exchange Online. You can migrate a maximum of 1,000 Microsoft Exchange Server 2003, Exchange Server 2007, or Exchange Server 2010 mailboxes using a cutover migration. Mailboxes will be automatically provisioned in Exchange Online when you perform a cutover Exchange migration. For more information, see Example 5.
- **Staged Exchange migration:** You can also migrate a subset of mailboxes from an on-premises Exchange organization to Exchange Online. This is another type of onboarding migration. You can migrate only Exchange 2003 and Exchange 2007 mailboxes using a staged Exchange migration. Migrating Exchange 2010 and Exchange 2013 mailboxes isn't supported using a staged migration. Prior to running a staged migration, you have to use directory synchronization or some other method to provision mail users in your Exchange Online organization. For more information, see Example 6.
- **IMAP migration:** This onboarding migration type migrates mailbox data from an IMAP server (including Exchange) to Exchange Online. For an IMAP migration, you must first provision mailboxes in Exchange Online before you can migrate mailbox data. For more information, see Example 7.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description

Local	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Local* parameter specifies a local move, where mailboxes are moved to a different mailbox database within the same forest.
Name	Required	System.String	The *Name* parameter specifies an identifying name for the migration batch.
UserIds	Required	Microsoft.Exchange.Management.Migration.MigrationUserIdParameter[]	The *UserIds* parameter specifies an array of one or more user identities to be included in the migration batch.
Users	Required	Microsoft.Exchange.Migration.MigrationUser[]	The *Users* parameter specifies an array of one or more users to be included in the migration batch.
XO1	Required	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
AllowIncrementalSyncs	Optional	System.Boolean	The *AllowIncrementalSyncs* parameter specifies whether to allow

			incremental synchronization. During incremental synchronization, the source and target mailboxes are synchronized. This means that any new messages sent to the source mailbox are copied to the corresponding target mailbox. This occurs every 24 hours. You can use one of the following values for this parameter: • `$true` Enables incremental synchronization; this is the default value. • `$false` Prevents incremental synchronizations. If you use this value, the migration batch will go into **Stopped** state after initial synchronization is complete. To complete a migration batch for local moves, cross-forest moves, or remote move migrations, you have to enable incremental synchronization by using the **Set-MigrationBatch** cmdlet.

ArchiveOnly	Optional	System.Management.Automation.SwitchParameter	The *ArchiveOnly* parameter specifies whether to migrate only the archive mailboxes from the source to the target destination for the users specified in the migration batch. Primary mailboxes aren't migrated when this parameter is used. This parameter can only be used for local moves and remote move migrations. You can use the *TargetArchiveDatabases* parameter to specify the database to migrate the archive mailboxes to. You can also specify the target archive database in the CSV file. If you don't specify the target archive database, the cmdlet uses the automatic mailbox distribution logic to select the database.
AutoComplete	Optional	System.Management.Automation.SwitchParameter	The *AutoComplete* parameter specifies whether to force the finalization of the individual mailboxes in a

			migration batch when the initial synchronization for a mailbox is successfully completed. Alternatively, you have to run the **Complete-MigrationBatch** cmdlet to finalize a migration batch. This parameter can only be used for local moves and remote move migrations. This parameter will force the individual mailboxes to be finalized as soon as the mailbox has completed initial synchronization. **Important:** If you're migrating mailboxes from Exchange Online to an on-premises Exchange Server 2007 organization, you must include the *AutoComplete* parameter. This parameter is required because offboarding cloud-based mailboxes to an Exchange 2007 organization is performed offline, and mailboxes in the migration batch must be finalized after the initial synchronization is successfully completed.
AutoRetryCount	Optional	System.Int32	The *AutoRetryCount*

			parameter specifies the number of attempts to restart the migration batch to migrate mailboxes that encountered errors.
AutoStart	Optional	System.Management.Automation.SwitchParameter	The *AutoStart* parameter specifies whether to immediately start the processing of the new migration batch. If you don't use the *AutoStart* parameter, you have to manually start the migration batch by using the **Start-MigrationBatch** cmdlet.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the migration request encounters corruption in the mailbox. Use 0 to not skip bad items. The valid input range for this parameter is from 0 through 2147483647 and the value `unlimited`. The default value is 0. We recommend that you keep the default value 0 and

			only change the *BadItemLimit* parameter value if the move or migration fails.
CompleteAfter	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CSVData	Optional	System.Byte[]	The *CSVData* parameter specifies the CSV file that contains information about the user mailboxes to be moved or migrated. The required attributes in the header row of the CSV file vary depending on the type of migration. For more information, see CSV files for mailbox migration. Use the following format for the value of this parameter: `([System.IO.File]::Rea`

			`dAllBytes(<path of the CSV migration file>))`. The following is an example: `CSVData: ([System.IO.File]::ReadAllBytes("C:\Users\Administrator\Desktop\MigrationBatch_1.csv"))`
DisableOnCopy	Optional	System.Management.Automation.SwitchParameter	The *DisableOnCopy* parameter specifies that if copying items from one batch to another, the original job item will be disabled.
DisallowExistingUsers	Optional	System.Management.Automation.SwitchParameter	The *DisallowExistingUsers* parameter specifies whether to prevent the migration of mailboxes that are currently being migrated in a different migration batch. A validation warning is displayed for any pre-existing mailbox in the target destination.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			writes this configuration change to Active Directory.
ExcludeFolders	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in the cloud-based service. For an IMAP migration, the *ExcludeFolders* parameter specifies mailbox folders that you don't want to migrate from the on-premises messaging system to the cloud-based mailboxes. Use folder names relative to the IMAP root on the on-premises mail server. Specify the value as a string array and separate multiple folder names with commas.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of large items in each mailbox in the migration batch that will be skipped. When the number of large items exceeds this value, the migration for the mailbox fails. The default value is 0,

			which means that the migration fails if the mailbox contains any large items. For an on-premises Exchange organization, the size limit is defined by the target mailbox database. In Exchange Online, items up to 35 megabytes (MB) are migrated.
Locale	Optional	System.Globalization.CultureInfo	This parameter is available only in on-premises Exchange 2013. The *Locale* parameter specifies the language for the organization.
NotificationEmails	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *NotificationEmails* parameter specifies one or more email addresses that migration status reports are sent to. Specify the value as a string array, and separate multiple email addresses with commas. If you don't use this parameter, the status report isn't sent.
Organization	Optional	Microsoft.Exchange.Co	The *Organization*

		nfiguration.Tasks.OrganizationIdParameter	parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
PrimaryOnly	Optional	System.Management.Automation.SwitchParameter	The *PrimaryOnly* parameter specifies that only the primary mailboxes are migrated from the source destination to the target organization for the users specified in the migration batch. Archive mailboxes aren't migrated when this parameter is used. This parameter can only be used for local moves and remote move migrations.
ReportInterval	Optional	System.TimeSpan	This parameter is reserved for internal Microsoft use.
SkipSteps	Optional	Microsoft.Exchange.Data.Storage.Management.SkippableMigrationSteps[]	The *SkipSteps* parameter allows you to skip the step of setting the target email address on the source mailbox. This prevents mail from being forwarded from the original mailbox to the new mailbox that was

			migrated to the target destination. Use one of the following values with this parameter: • `SettingTargetAddress` • `None` This parameter is only enforced for migration batches for a staged Exchange migration.
SourceEndpoint	Optional	Microsoft.Exchange.Management.Migration.MigrationEndpointIdParameter	The *SourceEndpoint* parameter specifies the name of the migration endpoint for the current location of the mailboxes specified in the migration batch. The migration endpoint specified by this parameter contains the connection settings that will be used by the migration process to connect to the server where the mailboxes specified in the migration batch are located.
StartAfter	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.
TargetArchiveDatabases	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *TargetArchiveDatabases* parameter specifies the database where the

			archive mailboxes specified in the migration batch will be migrated to. You can also specify multiple databases for the value of this parameter. The migration service selects one database as the target database to move the archive mailbox to. For example: `-TargetArchiveDatabases @(MBXDB01,MBXDB02,MBXDB03)` This parameter can only be used for local moves and remote move migrations.
TargetDatabases	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *TargetDatabases* parameter specifies the identity of the database that you're moving mailboxes to. You can use the following values: • Database GUID • Database name If you don't specify the *TargetDatabases* parameter for a local move, the cmdlet uses the automatic mailbox distribution logic to select the database.

			You can also specify multiple databases for the value of this parameter. The migration service will select one as the target database to move the mailbox to. For example: -`TargetDatabases @(MBXDB01,MBXDB02,MBXDB03)` This parameter can only be used for local moves and remote move migrations.
TargetDeliveryDomain	Optional	System.String	The *TargetDeliveryDomain* parameter specifies the FQDN of the external email address created in the source forest for the mail-enabled user when the migration batch is complete. This parameter is required when creating remote move onboarding and remote offboarding migration batches.
TargetEndpoint	Optional	Microsoft.Exchange.Management.Migration.MigrationEndpointIdParameter	The *TargetEndpoint* parameter specifies the name of the migration endpoint for the destination where the mailboxes specified in the

			migration batch will be moved or migrated to. The migration endpoint specified by this parameter contains the connection settings that will be used by the migration process to connect to the server where the mailboxes will be moved.
TimeZone	Optional	Microsoft.Exchange.Data.Storage.Management.ExTimeZoneValue	The *TimeZone* parameter specifies the time zone of the administrator who submits the migration batch. Use a valid Windows time zone name. You can use Windows PowerShell to retrieve time zone names from the registry, for example: `Get-ChildItem "HKLM:\Software\Microsoft\Windows NT\CurrentVersion\Time zones" | Format-List pschildname`
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MigrationBatch

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MigrationBatch** cmdlet to delete a migration batch that either isn't running or has been completed. If necessary, you can run the **Get-MigrationBatch** cmdlet to determine the status of a migration batch before you remove it.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MigrationBatch [-Identity <MigrationBatchIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-Organization <OrganizationIdParameter>] [-Partition
<MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the migration batch Cutover1.

```
Remove-MigrationBatch -Identity Cutover1
```

EXAMPLE 2

This example removes the corrupted migration batch LocalMove1.

```
Remove-MigrationBatch -Identity LocalMove1 -Force
```

Detailed Description

The **Remove-MigrationBatch** cmdlet removes a migration batch. All subscriptions are deleted, and any object related to the migration batch is also deleted.

If you use the *Force* parameter with this cmdlet, the individual user requests and subscriptions that were part of the removed migration batch aren't removed. You have to remove the individual migration user requests with the `Remove-MigrationUser <Identity> -Force` command.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange

			2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to remove a corrupted migration batch. Corrupted migration batches have a status of **Corrupted**. If you try to remove a corrupted migration batch without using this switch, you receive an error saying the migrated batch can't be found. If you use this parameter to remove a corrupted migration batch, the individual user requests (also called *job items*) and subscriptions that were part of the removed migration batch aren't

			removed. You have to remove the individual migration user requests with the `Remove-MigrationUser <Identity> -Force` command.
Identity	Optional	Microsoft.Exchange.Management.Migration.MigrationBatchIdParameter	The *Identity* parameter identifies the migration batch that you want to remove. The value for this parameter is the name that was specified when the migration batch was created.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of

			those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MigrationBatch

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MigrationBatch** cmdlet to update a migration request for a batch of users. For more information, see New-MigrationBatch.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MigrationBatch -Identity <MigrationBatchIdParameter> [-
AllowIncrementalSyncs <$true | $false>] [-AutoRetryCount <Int32>] [-
BadItemLimit <Unlimited>] [-CompleteAfter <DateTime>] [-Confirm
[<SwitchParameter>]] [-CSVData <Byte[]>] [-DomainController <Fqdn>] [-
LargeItemLimit <Unlimited>] [-NotificationEmails <MultiValuedProperty>] [-
Organization <OrganizationIdParameter>] [-Partition <MailboxIdParameter>]
[-ReportInterval <TimeSpan>] [-StartAfter <DateTime>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example updates MigrationBatch01 with new *AutoRetryCount* and *AllowIncrementalSyncs*

parameter settings.

```
Set-MigrationBatch -Identity MigrationBatch01 -
AutoRetryCount 5 -AllowIncrementalSyncs $true
```

Detailed Description

The **Set-MigrationBatch** cmdlet configures your existing migration batches to migrate mailboxes and mailbox data in one of the following scenarios:

- Local move
- Cross-forest move
- Remote move
- Cutover Exchange migration
- Staged Exchange migration
- IMAP migration

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.Migration.MigrationBatchIdParameter	The *Identity* parameter specifies the name of the migration batch that you want to update configuration settings for.
AllowIncrementalSyncs	Optional	System.Boolean	The *AllowIncrementalSyncs* parameter specifies whether to enable or disable incremental syncs on mailbox moves and migrations.

AutoRetryCount	Optional	System.Int32	The *AutoRetryCount* parameter specifies the number of attempts to restart the migration batch to migrate mailboxes that encountered errors.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use 0 to not skip bad items. The valid input range for this parameter is from 0 through 2147483647. The default value is 0. We recommend that you keep the default value 0 and only change the *BadItemLimit* parameter value if the request fails.
CompleteAfter	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CSVData	Optional	System.Byte[]	The *CSVData* parameter specifies the CSV file that contains information about the user mailboxes to be moved or migrated. The required attributes in the header row of the CSV file vary depending on the type of migration. Use the following format for the value of this parameter: `([System.IO.File]::ReadAllBytes(<path of the CSV migration file>))`. The following is an example: `CSVData: ([System.IO.File]::ReadAllBytes("C:\Users\Administrator\Desktop\MigrationBatch_1.csv"))`
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-

			premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the size of large items to skip if the request encounters such items in the mailbox. Use 0 to not skip large items. The valid input range for this parameter is from 0 through 2147483647. The default value is 50. We recommend that you keep the default value 0 and only change the *LargeItemLimit* parameter value if the request fails.
NotificationEmails	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *NotificationEmails* parameter specifies one or more email addresses that

			migration status reports are sent to. Specify the value as a string array, and separate multiple email addresses with commas. If you don't use this parameter, the status report isn't sent.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
ReportInterval	Optional	System.TimeSpan	This parameter is reserved for internal Microsoft use.
StartAfter	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Start-MigrationBatch

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Start-MigrationBatch** cmdlet to start a move request or migration batch that was created with the **New-MigrationBatch** cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Start-MigrationBatch [-Identity <MigrationBatchIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-Partition <MailboxIdParameter>] [-Validate
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example starts the migration batch SEM1.

```
Start-MigrationBatch -Identity SEM1
```

Detailed Description

The **Start-MigrationBatch** cmdlet starts a pending migration batch that was created, but not started, with the **New-MigrationBatch** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox move and migration permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Management.Migration.MigrationBatchIdParameter	The *Identity* parameter identifies the migration batch that you want to start. Use the migration batch *Name* parameter as the value for this parameter. Use the **Get-MigrationBatch** cmdlet to identify the name of the migration batch.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
Validate	Optional	System.Management.Automation.SwitchParameter	The *Validate* parameter specifies whether to start the migration batch in the validation stage of the migration process. If you include this parameter, the migration performs a validation check of the mailboxes in the batch.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Stop-MigrationBatch

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Stop-MigrationBatch** cmdlet to stop the processing of a migration batch that's in progress.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Stop-MigrationBatch [-Identity <MigrationBatchIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-Partition <MailboxIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example stops the migration batch that's currently being processed by the migration service.

```
Stop-MigrationBatch
```

EXAMPLE 2

This example stops the migration batch MigrationBatch1.

```
Stop-MigrationBatch -Identity MigrationBatch1
```

Detailed Description

The **Stop-MigrationBatch** cmdlet stops the migration batch that's being processed in your on-premises Exchange organization or by the cloud-based migration service running in Microsoft Office 365. You can only stop migration batches that have mailboxes that are still in the process of being migrated or are waiting to be migrated. Stopping a migration won't affect mailboxes that have been migrated already. The migration of mailboxes that are being actively migrated is stopped immediately. If all migration requests in a migration batch are completed or failed, this cmdlet won't run.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox move and migration permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Management.Migration.MigrationBatchIdParameter	The *Identity* parameter identifies the name of the current migration batch. The value for this parameter is specified by the *Name* parameter of the **New-MigrationBatch** cmdlet.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MigrationConfig

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Get-MigrationConfig** cmdlet to retrieve migration configuration settings on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MigrationConfig [-Identity <MigrationConfigIdParameter>] [-
DomainController <Fqdn>] [-Partition <MailboxIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves the settings for the migration configuration.

```
Get-MigrationConfig
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox move and migration permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Identity	Optional	Microsoft.Exchange.Management.Migration.MigrationConfigIdParameter	This parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MigrationConfig

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-08*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-MigrationConfig** cmdlet to edit migration configurations for a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MigrationConfig [-Identity <MigrationConfigIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Features <None |
MultiBatch | Endpoints | UpgradeBlock | PAW>] [-MaxConcurrentMigrations
<Unlimited>] [-MaxNumberOfBatches <Int32>] [-Partition
<MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the migration system to have a maximum of 50 batches at any time.

```
Set-MigrationConfig -MaxNumberOfBatches 50
```

EXAMPLE 2

This example sets the migration system to only allow 100 concurrent migrations.

```
Set-MigrationConfig -MaxConcurrentMigrations 100
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox move and migration permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			writes this configuration change to Active Directory.
Features	Optional	Microsoft.Exchange.Data.Storage.Management.MigrationFeature	The *Features* parameter specifies the set of features to enable for the migration system. Use one of the following values: • `None` • `MultiBatch` • `Endpoints` • `UpgradeBlock`
Identity	Optional	Microsoft.Exchange.Management.Migration.MigrationConfigIdParameter	This parameter is reserved for internal Microsoft use.
MaxConcurrentMigrations	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxConcurrentMigrations* parameter specifies the maximum number of active migrations that your organization can run at any specific time.
MaxNumberOfBatches	Optional	System.Int32	The *MaxNumberOfBatches* parameter specifies the maximum number of batches that your organization can migrate at any time.
Partition	Optional	Microsoft.Exchange.Co	This parameter is reserved

		nfiguration.Tasks.MailboxIdParameter	for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MigrationEndpoint

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-17*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MigrationEndpoint** cmdlet to retrieve migration endpoint settings for source or

destination servers for cutover or staged Exchange migrations, IMAP migrations, and remote moves.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MigrationEndpoint [-Identity <MigrationEndpointIdParameter>] <COMMON
PARAMETERS>


Get-MigrationEndpoint -Type <None | IMAP | XO1 | ExchangeOutlookAnywhere |
BulkProvisioning | ExchangeRemoteMove | ExchangeLocalMove | PSTImport>
<COMMON PARAMETERS>


Get-MigrationEndpoint -ConnectionSettings <ExchangeConnectionSettings>
<COMMON PARAMETERS>


COMMON PARAMETERS: [-Diagnostic <SwitchParameter>] [-DiagnosticArgument
<String>] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-Partition <MailboxIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves the settings for the migration endpoint, OnboardingME01.

```
Get-MigrationEndpoint -Identity OnboardingME01
```

Detailed Description

The **Get-MigrationEndpoint** cmdlet retrieves settings for different types of migration:

- **Cross-forest move** Move mailboxes between two different on-premises Exchange forests. Cross-forest moves require the use of a RemoteMove endpoint.
- **Remote move** In a hybrid deployment, a remote move involves *onboarding* or *offboarding* migrations. Remote moves require the use of a RemoteMove endpoint. Onboarding moves mailboxes from an on-premises Exchange organization to Exchange Online in Office 365, and uses a RemoteMove endpoint as the source endpoint of the migration batch. Offboarding moves mailboxes from Exchange Online in Office 365 to an on-premises Exchange organization and uses a RemoteMove endpoint as the target endpoint of the migration batch.
- **Cutover Exchange migration** Migrate all mailboxes in an on-premises Exchange organization to Exchange Online in Office 365. Cutover Exchange migration requires the use of an Exchange endpoint.
- **Staged Exchange migration** Migrate a subset of mailboxes from an on-premises Exchange organization to Exchange Online in Office 365. Staged Exchange migration requires the use of an Exchange endpoint.
- **IMAP migration** Migrate mailbox data from an on-premises Exchange organization or other email system to Exchange Online in Office 365. For an IMAP migration, you must first create the cloud-based mailboxes before you migrate mailbox data. IMAP migrations require the use of an

IMAP endpoint.

- **Local** Move mailboxes between different servers or databases within a single on-premises Exchange forest. Local moves don't require the use of an endpoint.

For more information about the different move and migration scenarios, see:

- Mailbox moves in Exchange 2013
- Manage on-premises moves

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
ConnectionSettings	Required	Microsoft.Exchange.Data.Storage.Management.ExchangeConnectionSettings	The *ConnectionSettings* parameter specifies the configuration settings of source or target servers for which you want to find a matching endpoint.
Type	Required	Microsoft.Exchange.Data.Storage.Management.MigrationType	The *Type* parameter specifies what type of migration you want configuration settings for: • Remote moves and migrations • Cutover or staged Exchange migrations • IMAP migrations • PST Import
Diagnostic	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
DiagnosticArgument	Optional	System.String	This parameter is reserved

			for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Management.Migration.MigrationEndpointIdParameter	The *Identity* parameter specifies the name of the migration endpoint you want to retrieve settings for.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MigrationEndpoint

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-17*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MigrationEndpoint** cmdlet to configure the connection settings for cross-forests moves, remote move migrations, cutover or staged Exchange migrations, and IMAP migrations.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MigrationEndpoint -ExchangeRemoteMove <SwitchParameter> -RemoteServer
<Fqdn> [-Credentials <PSCredential>] <COMMON PARAMETERS>


New-MigrationEndpoint -Autodiscover <SwitchParameter> -Credentials
<PSCredential> -EmailAddress <SmtpAddress> -ExchangeOutlookAnywhere
<SwitchParameter> [-MailboxPermission <Admin | FullAccess>] [-
SourceMailboxLegacyDN <String>] [-TestMailbox <MailboxIdParameter>]
<COMMON PARAMETERS>


New-MigrationEndpoint -Credentials <PSCredential> -ExchangeOutlookAnywhere
<SwitchParameter> [-Authentication <Basic | Digest | Ntlm | Fba |
WindowsIntegrated | LiveIdFba | LiveIdBasic | WSSecurity | Certificate |
NegoEx | OAuth | Adfs | Kerberos | Negotiate | LiveIdNegotiate |
Misconfigured>] [-EmailAddress <SmtpAddress>] [-ExchangeServer <String>]
[-MailboxPermission <Admin | FullAccess>] [-NspiServer <String>] [-
RpcProxyServer <Fqdn>] [-SourceMailboxLegacyDN <String>] [-TestMailbox
<MailboxIdParameter>] <COMMON PARAMETERS>


New-MigrationEndpoint -PSTImport <SwitchParameter> -RemoteServer <Fqdn> [-
Credentials <PSCredential>] <COMMON PARAMETERS>


New-MigrationEndpoint -Autodiscover <SwitchParameter> -Credentials
<PSCredential> -EmailAddress <SmtpAddress> -ExchangeRemoteMove
<SwitchParameter> <COMMON PARAMETERS>


New-MigrationEndpoint -IMAP <SwitchParameter> -RemoteServer <Fqdn> [-
Authentication <Basic | Digest | Ntlm | Fba | WindowsIntegrated |
LiveIdFba | LiveIdBasic | WSSecurity | Certificate | NegoEx | OAuth | Adfs
| Kerberos | Negotiate | LiveIdNegotiate | Misconfigured>] [-Port <Int32>]
[-Security <None | Ssl | Tls>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-MaxConcurrentIncrementalSyncs <Unlimited>] [-
MaxConcurrentMigrations <Unlimited>] [-Organization
<OrganizationIdParameter>] [-Partition <MailboxIdParameter>] [-
SkipVerification <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates an endpoint for remote moves by using the *Autodiscover* parameter to detect the settings.

```
New-MigrationEndpoint -Name Endpoint1 -ExchangeRemoteMove -
Autodiscover -EmailAddress tonysmith@contoso.com -
Credentials (Get-Credential contoso\tonysmith)
```

EXAMPLE 2

This example creates an endpoint for remote moves by specifying the settings manually.

```
New-MigrationEndpoint -Name Endpoint2 -ExchangeRemoteMove -
RemoteServer MRSServer.contoso.com -Credentials (Get-
Credential Contoso.com\Administrator)
```

EXAMPLE 3

This example creates an Outlook Anywhere migration endpoint by using the *Autodiscover* parameter to detect the connection settings to the on-premises organization. Outlook Anywhere endpoints are used for cutover and staged Exchange migrations. The **Get-Credential** cmdlet is used to obtain the credentials for an on-premises account that has the necessary administrative privileges in the domain and that can access the mailboxes that will be migrated. When prompted for the user name, you can use either the email address or the domain\user name format for the administrator account. This account can be the same one that is specified by the *EmailAddress* parameter.

```
$Credentials = Get-Credential

New-MigrationEndpoint -ExchangeOutlookAnywhere -Name EXCH-
AutoDiscover -Autodiscover -EmailAddress
administrator@contoso.com -Credentials $Credentials
```

EXAMPLE 4

This example creates an Outlook Anywhere migration endpoint by specifying the connection settings manually. Outlook Anywhere endpoints are used for cutover and staged Exchange migrations. The value for the *ExchangeServer* parameter specifies the on-premises Exchange server that hosts the mailboxes that will be migrated. The value for the *RPCProxyServer* parameter specifies a Client Access server in the on-premises organization. The *EmailAddress* parameter can specify any mailbox in the on-premises domain.

```
New-MigrationEndpoint -ExchangeOutlookAnywhere -Name
EXCH_Manual -ExchangeServer EXCH-01-MBX.contoso.com -
```

```
RPCProxyServer EXCH-02-CAS.contoso.com -Credentials (Get-
Credential administrator@contoso.com) -EmailAddress
annb@contoso.com
```

Important:
It's recommended that you use a migration endpoint created with connection settings that are automatically discovered (see Example 3) because the Autodiscover service will be used to connect to each user mailbox in the migration batch. If you manually specify the connection settings for the endpoint and a user mailbox isn't located on the server specified by the *ExchangeServer* parameter, the migration for that user will fail. This is important if you have multiple on-premises Exchange servers. Otherwise, you may need to create different migration endpoints that correspond to each on-premises server.

EXAMPLE 5

This example creates an IMAP migration endpoint. The value for the *RemoteServer* parameter specifies the FQDN of the IMAP server that hosts the mailboxes that will be migrated. The endpoint is configured to use port 993 for SSL encryption.

```
New-MigrationEndpoint -IMAP -Name IMAPEndpoint -
RemoteServer imap.contoso.com -Port 993 -Security Ssl
```

EXAMPLE 6

This example creates an IMAP migration endpoint that supports 50 concurrent migrations and 10 concurrent incremental synchronizations. The endpoint is configured to use port 143 for TLS encryption.

```
New-MigrationEndpoint -IMAP -Name IMAP_TLS_Endpoint -
RemoteServer imap.contoso.com -Port 143 -Security Tls -
MaxConcurrentMigrations
50 -MaxConcurrentIncrementalSyncs 10
```

Detailed Description

The **New-MigrationEndpoint** cmdlet configures the connection settings for different types of migrations:

- **Cross-forest move:** Move mailboxes between two different on-premises Exchange forests. Cross-forest moves require the use of a Remote Move endpoint.
- **Remote move migration:** In a hybrid deployment, a remote move migration involves *onboarding* or *offboarding* migrations. Remote move migrations also require the use of an Exchange remote move endpoint. Onboarding moves mailboxes from an on-premises Exchange organization to Exchange Online in Office 365, and uses a remote move endpoint as the source endpoint of the migration batch. Offboarding moves mailboxes from Exchange Online in Office 365 to an on-

premises Exchange organization and uses a remote move endpoint as the target endpoint of the migration batch.

- **Cutover Exchange migration:** Migrate all mailboxes in an on-premises Exchange organization to Exchange Online in Office 365. A cutover Exchange migration requires the use of an Outlook Anywhere migration endpoint.
- **Staged Exchange migration:** Migrate a subset of mailboxes from an on-premises Exchange organization to Exchange Online in Office 365. A staged Exchange migration requires the use of an Outlook Anywhere migration endpoint.
- **IMAP migration:** Migrate mailbox data from an on-premises Exchange organization or other email system to Exchange Online in Office 365. For an IMAP migration, you must first create the cloud-based mailboxes before you migrate mailbox data. IMAP migrations require the use of an IMAP endpoint.

Note:
Moving mailboxes between different servers or databases within a single on-premises Exchange forest (called a *local move*) doesn't require a migration endpoint.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Autodiscover	Required	System.Management.Automation.SwitchParameter	For an Exchange migration, the *Autodiscover* parameter specifies whether to get other connection settings for the on-premises server from the Autodiscover service.
EmailAddress	Required	Microsoft.Exchange.Data.SmtpAddress	The *EmailAddress* parameter specifies the email address used by the Autodiscover service or in some cases used to validate the endpoint

			when you specify the connection settings manually.
ExchangeOutlookAnywhere	Required	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *ExchangeOutlookAnywhere* parameter specifies the type of endpoint for staged and cutover migrations.
ExchangeRemoteMove	Required	System.Management.Automation.SwitchParameter	The *ExchangeRemoteMove* parameter specifies the type of endpoint for cross-forest moves and remote move migrations in a hybrid deployment.
IMAP	Required	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *IMAP* parameter specifies the type of endpoint for IMAP migrations.
Name	Required	System.String	The *Name* parameter specifies the name you give to the new migration endpoint. You can use the

			Name parameter when you run the **New-MigrationBatch** cmdlet.
PSTImport	Required	System.Management.Automation.SwitchParameter	The *PSTImport* parameter specifies the type of endpoint for PST Import migrations.
RemoteServer	Required	Microsoft.Exchange.Data.Fqdn	The *RemoteServer* parameter specifies the FQDN of the remote server, which depends on the protocol type for moves: • For cross-forest moves and remote move migrations, this parameter refers to the Client Access server in the on-premises organization. • For IMAP migrations, this parameter refers to the IMAP server.
Authentication	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthenticationMethod	This parameter is available only in the cloud-based service. The *Authentication* parameter specifies the authentication method used by the on-premises mail server. If you don't

			include this parameter, `Basic` authentication is used.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Credentials	Optional	System.Management.Automation.PSCredential	The *Credentials* parameter specifies the credentials to connect to the source or target endpoint for all Exchange migration types. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in the cloud-based service. The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExchangeServer	Optional	System.String	This parameter is available only in the cloud-based service. The *ExchangeServer* parameter specifies the FQDN of the on-premises Exchange server that hosts the mailboxes that will be migrated. This parameter is used when you create an Outlook Anywhere migration endpoint for cutover and staged Exchange migrations. This parameter is required only when you don't use the *Autodiscover* parameter.
MailboxPermission	Optional	Microsoft.Exchange.Data.Storage.Management.MigrationMailboxPermission	This parameter is available only in the cloud-based service. The *MailboxPermission* parameter specifies what permissions to use to access the source mailbox

			during Outlook Anywhere onboarding (staged Exchange migration and cutover Exchange migration). The migration administrator account specified for the endpoint must have one of the following permissions: • `Admin`: The account is a domain administrator who can access any mailbox they want to migrate. • `FullAccess`: The account is assigned either the Full Access permission to the mailboxes they want to migrate or the Receive As permission to the mailbox database that hosts the mailboxes that will be migrated. If this parameter isn't specified, the cmdlet tries to access source mailboxes using the domain administrator permission and if that fails, it then tries to access the source mailboxes using the Full Access or Receive As permissions. This parameter can't be

			used for creating non-Outlook Anywhere migration endpoints.
MaxConcurrentIncrementalSyncs	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxConcurrentIncrementalSyncs* parameter specifies the maximum number of incremental syncs allowed per endpoint. The default value is 20.
MaxConcurrentMigrations	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxConcurrentMigrations* parameter specifies the maximum number of mailboxes that are migrated during initial sync. This parameter is applicable for all migration types. The default value is 100.
NspiServer	Optional	System.String	This parameter is available only in the cloud-based service. The *NspiServer* parameter specifies the remote Name Service Provider Interface (NSPI) server location for cutover and staged migrations. You must provide the FQDN of

			the server.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
Port	Optional	System.Int32	This parameter is available only in the cloud-based service. For an IMAP migration, the *Port* parameter specifies the TCP port number used by the migration process to connect to the remote server. This parameter is required when you want to migrate data from an on-premises IMAP server to cloud-based mailboxes.
RpcProxyServer	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in the cloud-based service. The *RpcProxyServer* parameter specifies the FQDN of the Client Access server for the on-premises Exchange organization. This parameter is used when

			you create an Outlook Anywhere migration endpoint for cutover and staged Exchange migrations. Typically, this FQDN will be the same as your Outlook Web App URL; for example, mail.contoso.com. This is also the URL for the proxy server that Outlook uses to connect to an Exchange server. This parameter is required only when you don't use the *Autodiscover* parameter.
Security	Optional	Microsoft.Exchange.Data.IMAPSecurityMechanism	This parameter is available only in the cloud-based service. The *Security* parameter specifies the encryption method used by the IMAP server for an IMAP migration. Options are None, SSL (the default), or TLS. This parameter is required when you want to migrate data from an on-premises IMAP server to cloud-based mailboxes.

SkipVerification	Optional	System.Management.Automation.SwitchParameter	The *SkipVerification* switch specifies whether to skip verifying that the remote server is reachable when creating a migration endpoint. The default value is `$false`.
SourceMailboxLegacyDN	Optional	System.String	This parameter is available only in the cloud-based service. The *SourceMailboxLegacyDN* parameter specifies the LegacyExchangeDN of an on-premises mailbox used to test the ability of the migration service to create a connection using this endpoint. The cmdlet tries to access this mailbox using the credentials for the administrator account specified in the command.
TestMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in the cloud-based service. The *TestMailbox* parameter specifies an Exchange Online mailbox used as the target by the migration service to verify

			the connection using this endpoint. If this parameter isn't specified, the migration service uses the migration arbitration mailbox in the Exchange Online organization to verify the connection. This parameter is only used to create Outlook Anywhere migration endpoints.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MigrationEndpoint

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MigrationEndpoint** cmdlet to remove existing migration endpoints for source or destination servers for cutover or staged Exchange migrations, IMAP migrations, and remote moves.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MigrationEndpoint -Identity <MigrationEndpointIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-Partition <MailboxIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the migration endpoint CrossForestME01.

```
Remove-MigrationEndpoint -Identity CrossForestME01
```

Detailed Description

Use the **Remove-MigrationEndpoint** cmdlet to remove an existing migration endpoint.

For more information about migration endpoints, see Set-MigrationEndpoint and New-MigrationEndpoint.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Management.Migration.MigrationEndpointIdParameter	The *Identity* parameter specifies the name of the migration endpoint you want to remove.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Co	This parameter is

		nfiguration.Tasks.MailboxIdParameter	reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MigrationEndpoint

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MigrationEndpoint** cmdlet to edit settings for cutover or staged Exchange migrations, IMAP migrations, and remote moves.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MigrationEndpoint -Identity <MigrationEndpointIdParameter> [-
Authentication <Basic | Digest | Ntlm | Fba | WindowsIntegrated |
LiveIdFba | LiveIdBasic | WSSecurity | Certificate | NegoEx | OAuth | Adfs
| Kerberos | Negotiate | LiveIdNegotiate | Misconfigured>] [-Confirm
[<SwitchParameter>]] [-Credentials <PSCredential>] [-DomainController
<Fqdn>] [-EmailAddress <SmtpAddress>] [-ExchangeServer <String>] [-
MailboxPermission <Admin | FullAccess>] [-MaxConcurrentIncrementalSyncs
<Unlimited>] [-MaxConcurrentMigrations <Unlimited>] [-NspiServer <String>]
[-Organization <OrganizationIdParameter>] [-Partition
<MailboxIdParameter>] [-Port <Int32>] [-RemoteServer <Fqdn>] [-
RpcProxyServer <Fqdn>] [-Security <None | Ssl | Tls>] [-SkipVerification
<SwitchParameter>] [-SourceMailboxLegacyDN <String>] [-TestMailbox
<MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the *MaxConcurrentIncrementalSyncs* setting to 50 on the CutoverExchangeEndpoint01 migration endpoint using the NSPI server Server01.

```
Set-MigrationEndpoint -Identity CutoverExchangeEndpoint01 -
MaxConcurrentIncrementalSyncs 50 -NspiServer
Server01.contoso.com
```

EXAMPLE 2

This example changes the *MaxConcurrentMigrations* setting to 10 on the Onboardingmigrationendpoint01 migration endpoint using the remote server, Server01.

```
Set-MigrationEndpoint -Identity
Onboardingmigrationendpoint01 -MaxConcurrentMigrations 10 -
RemoteServer Server01.contoso.com
```

Detailed Description

Use the **Set-MigrationEndpoint** cmdlet to configure settings for different types of migration:

- **Cross-forest move** Move mailboxes between two different on-premises Exchange forests. Cross-forest moves require the use of a RemoteMove endpoint.
- **Remote move** In a hybrid deployment, a remote move involves *onboarding* or *offboarding* migrations. Remote moves require the use of a RemoteMove endpoint. Onboarding moves mailboxes from an on-premises Exchange organization to Exchange Online in Office 365, and uses a RemoteMove endpoint as the source endpoint of the migration batch. Offboarding moves mailboxes from Exchange Online in Office 365 to an on-premises Exchange organization and

uses a RemoteMove endpoint as the target endpoint of the migration batch.

- **Cutover Exchange migration** Migrate all mailboxes in an on-premises Exchange organization to Exchange Online in Office 365. Cutover Exchange migration requires the use of an Exchange endpoint.
- **Staged Exchange migration** Migrate a subset of mailboxes from an on-premises Exchange organization to Exchange Online in Office 365. Staged Exchange migration requires the use of an Exchange endpoint.
- **IMAP migration** Migrate mailbox data from an on-premises Exchange organization or other email system to Exchange Online in Office 365. For an IMAP migration, you must first create the cloud-based mailboxes before you migrate mailbox data. IMAP migrations require the use of an IMAP endpoint.
- **Local** Move mailboxes between different servers or databases within a single on-premises Exchange forest. Local moves don't require the use of an endpoint.

For more information about the different move and migration scenarios, see:

- Mailbox moves in Exchange 2013
- Manage on-premises moves

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.Migration.MigrationEndpointIdParameter	The *Identity* parameter specifies the name of the migration endpoint you want to configure.
Authentication	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthenticationMethod	This parameter is available only in the cloud-based service. The *Authentication* parameter specifies the authentication method used by the on-premises mail server.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Credentials	Optional	System.Management.Automation.PSCredential	The *Credentials* parameter specifies the credentials to use for connecting to the remote endpoint. Credentials should be used when creating either a staged or cutover Exchange endpoint or a RemoteMove endpoint. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in the cloud-based service. The *EmailAddress* parameter specifies the email address of an administrator account that can access the remote server.
ExchangeServer	Optional	System.String	This parameter is available only in the cloud-based service. The *ExchangeServer* parameter specifies the on-premises source Exchange server for cutover and staged migrations. This parameter is applicable only to staged and cutover Exchange endpoints which don't use Autodiscovery.
MailboxPermission	Optional	Microsoft.Exchange.Data.Storage.Management.MigrationMailboxPermission	This parameter is available only in the cloud-based service. The *MailboxPermission* parameter specifies what

			permissions should be used to access the source mailbox during OutlookAnywhere onboarding (Staged Exchange Migration and Cutover Exchange Migration). This parameter is not for non-OutlookAnywhere migrations. The account specified must have the following permissions: 1. FullAccess permission. The account has Full-Access permission to the mailboxes they want to migrate. 2. Admin permission. The account is a domain administrator who can access any mailbox they want to migrate.
MaxConcurrentIncrementalSyncs	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxConcurrentIncrementalSyncs* parameter specifies the maximum number of incremental syncs allowed for this endpoint at a specified time. This value must be less or equal to

			MaxConcurrentMigrations parameter.
MaxConcurrentMigrations	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxConcurrentMigrations* parameter specifies the maximum number of mailboxes that will be migrated for this endpoint at a specified time. This parameter is applicable for all migration types.
NspiServer	Optional	System.String	This parameter is available only in the cloud-based service. The *NspiServer* parameter specifies the FQDN of the remote Name Service Provider Interface (NSPI) server. This parameter is only applicable to staged and cutover Exchange endpoints that don't use Autodiscovery.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.

Port	Optional	System.Int32	This parameter is available only in the cloud-based service. For an IMAP migration, the *Port* parameter specifies the TCP port number used by the migration process to connect to the remote server.
RemoteServer	Optional	Microsoft.Exchange.Data.Fqdn	The *RemoteServer* parameter specifies the remote server depending on the protocol type for moves. • For Microsoft Exchange Server 2013 and Exchange Server 2010 moves, this parameter refers to the FQDN of a Client Access server or array, or group of Client Access servers behind a supported network load balancer. • For IMAP moves, this parameter refers to the FQDN of the IMAP server.
RpcProxyServer	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in the cloud-based service. For a staged Exchange migration, the

			RpcProxyServer parameter specifies the FQDN of the RPC proxy server for the on-premises Exchange server. This parameter is only applicable to staged and cutover Exchange endpoints that don't use Autodiscovery
Security	Optional	Microsoft.Exchange.Data.IMAPSecurityMechanism	This parameter is reserved for internal Microsoft use.
SkipVerification	Optional	System.Management.Automation.SwitchParameter	The *SkipVerification* switch specifies whether to skip verifying that the remote server is reachable when creating a migration endpoint. The default value is `$false`.
SourceMailboxLegacyDN	Optional	System.String	This parameter is available only in the cloud-based service. The *SourceMailboxLegacyDN* parameter specifies a mailbox on the target server. Use the LegacyExchangeDN for the on-premises test mailbox as the value for this parameter. The cmdlet tries

			to access this mailbox using the credentials for the administrator account on the target server.
TestMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in the cloud-based service. The *TestMailbox* parameter specifies a mailbox on the target server. Use the primary SMTP address as the value for this parameter. The cmdlet tries to access this mailbox using the credentials for the administrator account on the target server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-MigrationReport

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

The **Export-MigrationReport** is used by the Exchange migration process to enable an administrator to download a CSV file that contains migration errors for a selected migration batch. This cmdlet isn't run by an administrator in Windows PowerShell.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Export-MigrationReport <COMMON PARAMETERS>

Export-MigrationReport -CsvStream <Stream> -Identity
<MigrationReportIdParameter> <COMMON PARAMETERS>

Export-MigrationReport -Identity <MigrationReportIdParameter> -RowCount
<Int32> -StartingRowIndex <Int32> <COMMON PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>] [-Partition
<MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

There are no examples for using this cmdlet because the values used for the required parameters are generated by and available only to the migration process.

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
CsvStream	Required	System.IO.Stream	This parameter is reserved for internal Microsoft use.
Identity	Required	Microsoft.Exchange.Management.Migration.MigrationReportIdParameter	This parameter is reserved for internal Microsoft use.
RowCount	Required	System.Int32	This parameter is reserved for internal Microsoft use.
StartingRowIndex	Required	System.Int32	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Dat	This parameter is available only in on-

		a.Fqdn	premises Exchange 2013. This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-MigrationServerAvailability

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-17*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Test-MigrationServerAvailability** cmdlet to test the availability of the target server in preparation to perform cross-forest mailbox moves, migration of on-premises mailboxes to Exchange Online, or to migrate on-premises mailbox data from an IMAP server to Exchange Online mailboxes. For all migration types, the cmdlet attempts to verify the connection settings used to connect to the target server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-MigrationServerAvailability <COMMON PARAMETERS>


Test-MigrationServerAvailability -Imap <SwitchParameter> -Port <Int32> -
RemoteServer <Fqdn> [-Authentication <Basic | Digest | Ntlm | Fba |
WindowsIntegrated | LiveIdFba | LiveIdBasic | WSSecurity | Certificate |
NegoEx | OAuth | Adfs | Kerberos | Negotiate | LiveIdNegotiate |
Misconfigured>] [-Security <None | Ssl | Tls>] <COMMON PARAMETERS>


Test-MigrationServerAvailability -ExchangeRemoteMove <SwitchParameter> -
RemoteServer <Fqdn> [-Credentials <PSCredential>] <COMMON PARAMETERS>


Test-MigrationServerAvailability -Credentials <PSCredential> -PSTImport
<SwitchParameter> -RemoteServer <Fqdn> [-FilePath <String>] <COMMON
PARAMETERS>


Test-MigrationServerAvailability -Credentials <PSCredential> -
ExchangeOutlookAnywhere <SwitchParameter> -ExchangeServer <String> -
RPCProxyServer <Fqdn> [-Authentication <Basic | Digest | Ntlm | Fba |
WindowsIntegrated | LiveIdFba | LiveIdBasic | WSSecurity | Certificate |
NegoEx | OAuth | Adfs | Kerberos | Negotiate | LiveIdNegotiate |
Misconfigured>] [-EmailAddress <SmtpAddress>] [-MailboxPermission <Admin |
FullAccess>] [-SourceMailboxLegacyDN <String>] [-TestMailbox
<MailboxIdParameter>] <COMMON PARAMETERS>


Test-MigrationServerAvailability -Autodiscover <SwitchParameter> -
Credentials <PSCredential> -EmailAddress <SmtpAddress> -ExchangeRemoteMove
<SwitchParameter> <COMMON PARAMETERS>


Test-MigrationServerAvailability -Autodiscover <SwitchParameter> -
Credentials <PSCredential> -EmailAddress <SmtpAddress> -
ExchangeOutlookAnywhere <SwitchParameter> [-MailboxPermission <Admin |
FullAccess>] [-SourceMailboxLegacyDN <String>] [-TestMailbox
<MailboxIdParameter>] <COMMON PARAMETERS>
```

```
Test-MigrationServerAvailability -Endpoint <MigrationEndpointIdParameter>
<COMMON PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-Organization
<OrganizationIdParameter>] [-Partition <MailboxIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

For IMAP migrations, this example verifies the connection to the IMAP mail server imap.contoso.com.

```
Test-MigrationServerAvailability -Imap -RemoteServer
imap.contoso.com -Port 143
```

EXAMPLE 2

This example uses the *Autodiscover* and *ExchangeOutlookAnywhere* parameters to verify the connection to an on-premises Exchange server in preparation for migrating on-premises mailboxes to Exchange Online. You can use a similar example to test the connection settings for a staged Exchange migration or a cutover Exchange migration.

```
$Credentials = Get-Credential

Test-MigrationServerAvailability -ExchangeOutlookAnywhere -
Autodiscover -EmailAddress administrator@contoso.com -
Credentials $Credentials
```

EXAMPLE 3

This example verifies the connection to a server running Microsoft Exchange Server 2003 named exch2k3.contoso.com and uses NTLM for the authentication method.

```
$Credentials = Get-Credential

Test-MigrationServerAvailability -ExchangeOutlookAnywhere -
ExchangeServer exch2k3.contoso.com -Credentials
$Credentials -RPCProxyServer mail.contoso.com -
Authentication NTLM
```

EXAMPLE 4

This example verifies the connection settings to a remote server, and then uses those settings to

create a migration endpoint.

```
$Credentials = Get-Credential

$TSMA = Test-MigrationServerAvailability -
ExchangeRemoteMove -Autodiscover -EmailAddress
administrator@contoso.com -Credentials $Credentials

New-MigrationEndpoint -ExchangeRemoteMove -Name
ContosoEndpoint -ConnectionSettings
$TSMA.ConnectionSettings
```

Detailed Description

The **Test-MigrationServerAvailability** cmdlet verifies that you can communicate with the on-premises mail server that houses the mailbox data that you want to migrate to cloud-based mailboxes. When you run this cmdlet, you must specify the migration type. You can specify whether to communicate with an IMAP server or with an Exchange server.

For an IMAP migration, this cmdlet uses the server's fully qualified domain name (FQDN) and a port number to verify the connection. If the verification is successful, use the same connection settings when you create a migration request with the **New-MigrationBatch** cmdlet.

For an Exchange migration, this cmdlet uses one of the following settings to communicate with the on-premises server:

- For Exchange 2003, it uses the server's FQDN and credentials for an administrator account that can access the server.
- For Exchange Server 2007 and later versions, you can connect using the Autodiscover service and the email address of an administrator account that can access the server.

If the verification is successful, you can use the same settings to create a migration endpoint. For more information, see:

- New-MigrationEndpoint
- New-MigrationBatch

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description

Autodiscover	Required	System.Management.Automation.SwitchParameter	The *Autodiscover* parameter specifies that the cmdlet should use the Autodiscover service to obtain the connection settings for the target server.
Credentials	Required	System.Management.Automation.PSCredential	The *Credentials* parameter specifies the logon credentials for an account that can access mailboxes on the target server. Specify the *username* in the *domain\username* format or the user principal name (UPN) (user@example.com) format. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
EmailAddress	Required	Microsoft.Exchange.Data.SmtpAddress	The *EmailAddress* parameter specifies the email address of an administrator account that can access the remote server. This parameter is

			required when you use the *Autodiscover* parameter.
Endpoint	Required	Microsoft.Exchange.Management.Migration.MigrationEndpointIdParameter	The *Endpoint* parameter specifies the name of the migration endpoint to connect to. A migration endpoint contains the connection settings and other migration configuration settings. If you include this parameter, the **Test-MigrationServerAvailability** cmdlet attempts to verify the ability to connect to the remote server using the settings in the migration endpoint.
ExchangeOutlookAnywhere	Required	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *ExchangeOutlookAnywhere* parameter specifies a migration type for migrating on-premises mailboxes to Exchange Online. Use this parameter if you plan to migrate mailboxes to Exchange Online using a

			staged Exchange migration or a cutover Exchange migration.
ExchangeRemoteMove	Required	System.Management.Automation.SwitchParameter	The *ExchangeRemoteMove* parameter specifies a type of migration where mailboxes are moved with full fidelity between two on-premises forests or between an on-premises forest and Exchange Online. Use this parameter if you plan to perform a cross-forest move or migrate mailboxes between an on-premises Exchange organization and Exchange Online in a hybrid deployment.
ExchangeServer	Required	System.String	This parameter is available only in the cloud-based service. The *ExchangeServer* parameter specifies the FQDN of the on-premises Exchange server. Use this parameter when you plan to perform a staged Exchange migration or a cutover Exchange

			migration. This parameter is required if you don't use the *Autodiscover* parameter.
Imap	Required	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *Imap* parameter specifies an IMAP migration as the migration type. This parameter is required when you want to migrate data from an IMAP mail server to Exchange Online mailboxes.
Port	Required	System.Int32	This parameter is available only in the cloud-based service. The *Port* parameter specifies the TCP port number used by the IMAP migration process to connect to the target server. This parameter is required only for IMAP migrations. The standard is to use port 143 for unencrypted connections, port 143 for Transport Layer Security

			(TLS), and port 993 for Secure Sockets Layer (SSL).
PSTImport	Required	System.Management.Automation.SwitchParameter	The *PSTImport* parameter specifies that the migration endpoint being tested is for PST Imports.
RemoteServer	Required	Microsoft.Exchange.Data.Fqdn	The *RemoteServer* parameter specifies the FQDN of the on-premises mail server. This parameter is required when you want to perform one of the following migration types: • Cross-forest move • Remote move (hybrid deployments) • IMAP migration
RPCProxyServer	Required	Microsoft.Exchange.Data.Fqdn	This parameter is available only in the cloud-based service. The *RPCProxyServer* parameter specifies the FQDN of the RPC proxy server for the on-premises Exchange server. This parameter is required when you don't use the *Autodiscover* parameter. Use this parameter if you

			plan to perform a staged Exchange migration or a cutover Exchange migration to migrate mailboxes to Exchange Online.
Authentication	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthenticationMethod	This parameter is available only in the cloud-based service. The *Authentication* parameter specifies the authentication method used by the on-premises mail server. Use `Basic` or `NTLM`. If you don't include this parameter, `Basic` authentication is used. The parameter is only used for cutover Exchange migrations and staged Exchange migrations.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

FilePath	Optional	System.String	The *FilePath* parameter specifies the path containing the PST files when testing a PST Import migration endpoint.
MailboxPermission	Optional	Microsoft.Exchange.Data.Storage.Management.MigrationMailboxPermission	This parameter is available only in the cloud-based service. The *MailboxPermission* parameter specifies what permissions are assigned to the migration administrator account defined by the *Credentials* parameter. You make the permissions assignment to test the connectivity to a user mailbox on the source mail server when you're testing the connection settings in preparation for a staged or cutover Exchange migration or for creating an Exchange Outlook Anywhere migration endpoint. Specify one of the following values for the account defined by the *Credentials* parameter: • **FullAccess** The

			account has been assigned the Full-Access permission to the mailboxes that will be migrated. • **Admin** The account is a member of the Domain Admins group in the organization that hosts the mailboxes that will be migrated. This parameter isn't used for testing the connection to the remote server for a remote move migration or an IMAP migration.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
Security	Optional	Microsoft.Exchange.Data.IMAPSecurityMechanism	This parameter is available only in the cloud-based service. The *Security* parameter specifies the encryption method used by the remote mail server. The options are `None`, `Tls`, or `Ssl`. Use this parameter

			only when testing the connection to an IMAP server or in preparation for creating a migration endpoint for an IMAP migration.
SourceMailboxLegacyDN	Optional	System.String	This parameter is available only in the cloud-based service. The *SourceMailboxLegacyDN* parameter specifies a mailbox on the target server. Use the *LegacyExchangeDN* for the on-premises test mailbox as the value for this parameter. The cmdlet will attempt to access this mailbox using the credentials for the administrator account on the target server.
TestMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in the cloud-based service. The *TestMailbox* parameter specifies a mailbox on the target server. Use the primary SMTP address as the

			value for this parameter. The cmdlet will attempt to access this mailbox using the credentials for the administrator account on the target server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MigrationStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MigrationStatistics** cmdlet to view detailed information about migration requests.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MigrationStatistics [-Identity <MigrationStatisticsIdParameter>] [-
Diagnostic <SwitchParameter>] [-DiagnosticArgument <String>] [-
DomainController <Fqdn>] [-Partition <MailboxIdParameter>]
```

Examples

EXAMPLE 1

This example returns the default statistics for the migration batch MigBatch01.

```
Get-MigrationStatistics -Identity MigBatch01 -Diagnostic
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox move and migration permissions" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Diagnostic	Optional	System.Management.Automation.SwitchParameter	The *Diagnostic* parameter returns additional information that you can use to troubleshoot migration errors or send to Microsoft Customer Service and Support.
DiagnosticArgument	Optional	System.String	This parameter is reserved for internal

			Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Management.Migration.MigrationStatisticsIdParameter	The *Identity* parameter specifies the identity of the migration batch.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MigrationUser

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-09*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MigrationUser** cmdlet to view information about move and migration users.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MigrationUser [-Identity <MigrationUserIdParameter>] <COMMON
PARAMETERS>


Get-MigrationUser -MailboxGuid <Guid> <COMMON PARAMETERS>


Get-MigrationUser [-BatchId <MigrationBatchIdParameter>] [-Status <Queued
| Syncing | Failed | Synced | IncrementalFailed | Completing | Completed |
CompletionFailed | Corrupted | Provisioning | ProvisionUpdating |
CompletionSynced | Validating | IncrementalSyncing | IncrementalSynced |
CompletedWithWarnings | Stopped | IncrementalStopped | Starting | Stopping
| Removing>] [-StatusSummary <Active | Failed | Synced | Completed |
Stopped>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-Partition <MailboxIdParameter>] [-ResultSize
<Unlimited>]
```

Examples

EXAMPLE 1

This example retrieves status information about the recently migrated user, Tony Smith.

```
Get-MigrationUser -Identity TonySmith@contoso.com
```

EXAMPLE 2

This example retrieves more detailed information about any ongoing migration for the user with the specified mailbox GUID.

```
Get-MigrationUser -MailboxGuid b6a6795c-a010-4f67-aaaa-
da372d56fcb9 | Get-MigrationUserStatistics
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox move and migration permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
MailboxGuid	Required	System.Guid	The *MailboxGuid* parameter specifies the GUID of a mailbox for which you want to view the migration information.
BatchId	Optional	Microsoft.Exchange.Management.Migration.MigrationBatchIdParameter	The *BatchId* parameter specifies the name of the migration batch for which you want to return users.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. This parameter is reserved for internal Microsoft use.
Identity	Optional	Microsoft.Exchange.Management.Migration.MigrationUserIdParameter	The *Identity* parameter specifies the particular user that you want to retrieve information about. The *Identity* parameter is represented as an email address.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.

ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all users that match the query, use unlimited for the value of this parameter. The default value is 1000.
Status	Optional	Microsoft.Exchange.Data.Storage.Management.MigrationUserStatus	The *Status* parameter returns information about migration users that have the specified status state. Use one of the following values: • `Completed` • `CompletedWithWarnings` • `Completing` • `CompletionFailed` • `CompletionSynced` • `Corrupted` • `Failed` • `IncrementalFailed` • `IncrementalStopped` • `IncrementalSynced` • `IncrementalSyncing` • `Provisioning` • `ProvisionUpdating` • `Queued` • `Removing` • `Starting` • `Stopped` • `Stopping` • `Synced` • `Syncing` • `Validating`
StatusSummary	Optional	Microsoft.Exchange.Data.Storage.Management.MigrationUserStatusSummary	The *StatusSummary* parameter returns abbreviated information about migration users that have the specified status value. Use one of

			the following values: • Active • Completed • Failed • Stopped • Synced

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MigrationUser

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MigrationUser** cmdlet to remove a migration user from a batch.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MigrationUser -Identity <MigrationUserIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-Organization <OrganizationIdParameter>] [-Partition
<MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the migration user Tony Smith from a migration batch.

```
Remove-MigrationUser -Identity TonySmith
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Move and Migration Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.Migration.MigrationUserIdParameter	The *Identity* parameter specifies the user that you want to remove from the migration batch.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies that some specific checks and removal steps should be skipped and that the migration user should be forcibly removed. This parameter is used to work around issues where the migration user needs to be removed to fix issues when the user or data is corrupted, or to prevent such issues from occurring
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MigrationUserStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MigrationUserStatistics** cmdlet to view detailed information about the migration requested for a specific user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MigrationUserStatistics -Identity <MigrationUserIdParameter> [-
Diagnostic <SwitchParameter>] [-DiagnosticArgument <String>] [-
DomainController <Fqdn>] [-IncludeReport <SwitchParameter>] [-
LimitSkippedItemsTo <Int32>] [-Organization <OrganizationIdParameter>] [-
Partition <MailboxIdParameter>]
```

Examples

EXAMPLE 1

This example uses the *IncludeReport* parameter to display detailed information about the migration status for the user.

```
Get-MigrationUserStatistics -Identity
davidp@corp.contoso.com -IncludeReport | FL
Status,Error,Report
```

EXAMPLE 2

This example displays the number of mailbox items that failed to migrate, which are called skipped items, and information about each skipped item.

```
Get-MigrationUserStatistics -Identity
davidp@corp.contoso.com | FL SkippedItemCount,SkippedItems
```

This example displays results information in the *SkippedItems* property for a maximum of 20 skipped items.

```
Get-MigrationUserStatistics -Identity
davidp@corp.contoso.com -LimitSkippedItemsTo 20 | FL
SkippedItemCount,SkippedItems
```

EXAMPLE 3

This example displays detailed information about users in the migration batch named *StagedBatch1*.

```
Get-MigrationUser -BatchId StagedBatch1 | Get-
MigrationUserStatistics
```

EXAMPLE 4

This example displays detailed information about users from all current migration batches.

```
Get-MigrationUser | Get-MigrationUserStatistics
```

EXAMPLE 5

This example uses the *Diagnostic* parameter to display detailed troubleshooting information about the migration for the user.

```
Get-MigrationUserStatistics -Identity
davidp@corp.contoso.com -Diagnostic | FL
Status,Error,DiagnosticInfo
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox move and migration permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.Migration.MigrationUserIdParameter	The *Identity* parameter specifies the user that you want to retrieve information about. Use an email address as the value for the *Identity* parameter.
Diagnostic	Optional	System.Management.Automation.SwitchParameter	The *Diagnostic* switch specifies whether to retrieve extremely detailed information about the user migration. The information provided is too much for a typical person to comprehend. Microsoft Support can use this information to troubleshoot user migration problems. You typically use this

			parameter at the request of support personnel.
DiagnosticArgument	Optional	System.String	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
IncludeReport	Optional	System.Management.Automation.SwitchParameter	The *IncludeReport* switch specifies whether to return additional details, which can be used for troubleshooting.
LimitSkippedItemsTo	Optional	System.Int32	The *LimitSkippedItemsTo* parameter specifies the maximum number of skipped items to display information about in the *SkippedItems* property

			in command output. For example, if this parameter is set to 5, the cmdlet returns information for up to five skipped items for the specified user, even if there are more than five skipped items.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
Partition	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-09*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Get-MoveRequest** cmdlet to view the detailed status of an ongoing asynchronous mailbox move that was initiated by using the New-MoveRequest cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MoveRequest [-AccountPartition <AccountPartitionIdParameter>] [-
Identity <MoveRequestIdParameter>] [-IncludeSoftDeletedObjects
<SwitchParameter>] <COMMON PARAMETERS>


Get-MoveRequest [-AccountPartition <AccountPartitionIdParameter>] [-
BatchName <String>] [-Flags <None | CrossOrg | IntraOrg | Push | Pull |
Offline | Protected | RemoteLegacy | HighPriority | Suspend |
SuspendWhenReadyToComplete | MoveOnlyPrimaryMailbox |
MoveOnlyArchiveMailbox | TargetIsAggregatedMailbox | Join | Split>] [-
HighPriority <$true | $false>] [-IncludeSoftDeletedObjects
<SwitchParameter>] [-MoveStatus <None | Queued | InProgress |
AutoSuspended | CompletionInProgress | Synced | Completed |
CompletedWithWarning | Suspended | Failed>] [-Offline <$true | $false>] [-
Protect <$true | $false>] [-RemoteHostName <Fqdn>] [-SourceDatabase
<DatabaseIdParameter>] [-Suspend <$true | $false>] [-
SuspendWhenReadyToComplete <$true | $false>] [-TargetDatabase
<DatabaseIdParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Credential <PSCredential>] [-DomainController <Fqdn>]
[-Organization <OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-ResultSize <Unlimited>] [-SortBy
<String>]
```

Examples

EXAMPLE 1

This example retrieves the status of the ongoing mailbox move for Tony Smith's mailbox (tony@contoso.com).

```
Get-MoveRequest -Identity 'tony@contoso.com'
```

EXAMPLE 2

This example retrieves the status of ongoing mailbox moves to the target database DB05.

```
Get-MoveRequest -MoveStatus InProgress -TargetDatabase DB05
```

EXAMPLE 3

This example retrieves the status of move requests in the FromDB01ToDB02 batch that completed, but had warnings.

```
Get-MoveRequest -BatchName "FromDB01ToDB02" -MoveStatus
CompletedWithWarning
```

Detailed Description

The search criteria for the **Get-MoveRequest** cmdlet is a Boolean **And** statement. If you use multiple parameters, it narrows your search and reduces your search results.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox moves" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
BatchName	Optional	System.String	The *BatchName* parameter specifies the name that was given to a batch move request. You can't use this parameter with the *Identity* parameter.
Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This

			credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential. If you don't specify the *Credential* parameter, the credential of the current user is used.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Flags	Optional	Microsoft.Exchange.Data.Directory.Recipient.RequestFlags	The *Flags* parameter specifies the move type to retrieve information for. The following values may be used: • `CrossOrg` • `HighPriority` • `IntraOrg` • `Join` • `MoveOnlyArchiveMailbox` • `MoveOnlyPrimaryMailbox` • `None` • `Offline` • `Protected` • `Pull` • `Push` • `RemoteLegacy` • `Split` • `Suspend`

			• `SuspendWhenReadyToComplete` • `TargetIsAggregatedMailbox`
HighPriority	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *HighPriority* parameter specifies that the cmdlet returns requests that were created with the *HighPriority* flag. The *HighPriority* flag indicates that the request should be processed before other lower priority requests in the queue. You can't use this parameter with the *Identity* parameter.
Identity	Optional	Microsoft.Exchange.MailboxReplicationService.MoveRequestIdParameter	The *Identity* parameter specifies the identity of the mailbox or mail user. You can use the following values: 1. GUID distingushed name 2. Domain\User 3. User principal name (UPN) 4. Legacy Exchange SMTP addressAlias This parameter can't be used with the following parameters:

			• *BatchName* • *HighPriority* • *MoveStatus* • *Offline* • *Protect* • *RemoteHostName* • *SourceDatabase* • *Suspend* • *SuspendWhenReadyToComplete* • *TargetDatabase*
IncludeSoftDeletedObjects	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IncludeSoftDeletedObjects* parameter specifies whether to return mailboxes that have been soft deleted. This parameter accepts `$true` or `$false`.
MoveStatus	Optional	Microsoft.Exchange.Data.Directory.Recipient.RequestStatus	The *MoveStatus* parameter returns move requests in the specified status. You can use the following values: • `AutoSuspended` • `Completed` • `CompletedWithWarning` • `CompletionInProgress` • `Failed` • `InProgress` • `None` • `Queued` `Suspended`

			You can't use this parameter with the *Identity* parameter.
Offline	Optional	System.Boolean	The *Offline* parameter specifies whether to return mailboxes that are being moved in offline mode. This parameter accepts `$true` or `$false`. You can't use this parameter with the *Identity* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies an organizational unit (OU) and is used to limit the results. If you use this parameter, you only get move requests that pertain to mailboxes in the container that you specify. You can use either the OU or the domain name. If you use the OU, you must specify the canonical name of the OU.
Protect	Optional	System.Boolean	This parameter is available

			only in on-premises Exchange 2013. The *Protect* parameter returns mailboxes being moved in protected mode. This parameter accepts `$true` or `$false`. You can't use this parameter with the *Identity* parameter.
RemoteHostName	Optional	Microsoft.Exchange.Data.Fqdn	The *RemoteHostName* parameter specifies the FQDN of the cross-forest organization from which you're moving the mailbox. You can't use this parameter with the *Identity* parameter.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. You can sort by only one attribute at a time. The

			results are sorted in ascending order.
SourceDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *SourceDatabase* parameter specifies that all mailboxes being moved from the specified source database are returned. You can use the following values: • GUID of the database • Database name You can't use this parameter with the *Identity* parameter.
Suspend	Optional	System.Boolean	The *Suspend* parameter specifies whether to return mailboxes with moves that have been suspended. This parameter accepts `$true` or `$false`. You can't use this parameter with the *Identity* parameter.
SuspendWhenReadyToComplete	Optional	System.Boolean	The *SuspendWhenReadytoComplete* parameter specifies whether to return mailboxes that have been

			moved with the **New-MoveRequest** command and its *SuspendWhenReadyToComplete* switch. This parameter accepts `$true` or `$false`. You can't use this parameter with the *Identity* parameter.
TargetDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *TargetDatabase* parameter specifies whether to return all mailboxes that are being moved to the specified target database. You can use the following values: • GUID of the database • Database name You can't use this parameter with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-23*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MoveRequest** cmdlet to begin the process of an asynchronous mailbox or personal archive move. You can also check mailbox readiness to be moved by using the *WhatIf* parameter.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MoveRequest [-ArchiveOnly <SwitchParameter>] [-ArchiveTargetDatabase
<DatabaseIdParameter>] [-DoNotPreserveMailboxSignature <SwitchParameter>]
[-ForcePull <SwitchParameter>] [-ForcePush <SwitchParameter>] [-
PrimaryOnly <SwitchParameter>] [-TargetDatabase <DatabaseIdParameter>]
<COMMON PARAMETERS>


New-MoveRequest -Remote <SwitchParameter> -RemoteHostName <Fqdn> [-
ArchiveDomain <String>] [-ArchiveOnly <SwitchParameter>] [-
ArchiveTargetDatabase <DatabaseIdParameter>] [-
IgnoreTenantMigrationPolicies <SwitchParameter>] [-PrimaryOnly
<SwitchParameter>] [-RemoteCredential <PSCredential>] [-
RemoteGlobalCatalog <Fqdn>] [-RemoteOrganizationName <String>] [-
TargetDatabase <DatabaseIdParameter>] [-TargetDeliveryDomain <Fqdn>]
<COMMON PARAMETERS>


New-MoveRequest -RemoteCredential <PSCredential> -RemoteGlobalCatalog
<Fqdn> -RemoteLegacy <SwitchParameter> [-IgnoreTenantMigrationPolicies
<SwitchParameter>] [-RemoteTargetDatabase <String>] [-TargetDatabase
<DatabaseIdParameter>] [-TargetDeliveryDomain <Fqdn>] <COMMON PARAMETERS>


New-MoveRequest -Outbound <SwitchParameter> -RemoteHostName <Fqdn> [-
ArchiveDomain <String>] [-ArchiveOnly <SwitchParameter>] [-
IgnoreTenantMigrationPolicies <SwitchParameter>] [-PrimaryOnly
<SwitchParameter>] [-RemoteArchiveTargetDatabase <String>] [-
RemoteCredential <PSCredential>] [-RemoteGlobalCatalog <Fqdn>] [-
RemoteOrganizationName <String>] [-RemoteTargetDatabase <String>] [-
TargetDeliveryDomain <Fqdn>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Identity <MailboxOrMailUserIdParameter> [-
AcceptLargeDataLoss <SwitchParameter>] [-AllowLargeItems
<SwitchParameter>] [-BadItemLimit <Unlimited>] [-BatchName <String>] [-
CheckInitialProvisioningSetting <SwitchParameter>] [-CompleteAfter
<DateTime>] [-CompletedRequestAgeLimit <Unlimited>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ForceOffline
<SwitchParameter>] [-IgnoreRuleLimitErrors <SwitchParameter>] [-
IncrementalSyncInterval <TimeSpan>] [-InternalFlags <InternalMrsFlag[]>]
[-LargeItemLimit <Unlimited>] [-PreventCompletion <SwitchParameter>] [-
Priority <Lowest | Lower | Low | Normal | High | Higher | Highest |
Emergency>] [-Protect <SwitchParameter>] [-SkipMoving
<SkippableMoveComponent[]>] [-StartAfter <DateTime>] [-Suspend
```

```
<SwitchParameter>] [-SuspendComment <String>] [-SuspendWhenReadyToComplete
<SwitchParameter>] [-WhatIf [<SwitchParameter>]] [-WorkloadType <None |
Local | Onboarding | Offboarding | TenantUpgrade | LoadBalancing |
Emergency | RemotePstIngestion | SyncAggregation>]
```

Examples

EXAMPLE 1

This example tests a mailbox's readiness to move to the new database DB01 within the same forest and for completeness of the command by using the *WhatIf* switch. When you use the *WhatIf* switch, the system performs checks on the mailbox, and if the mailbox isn't ready, you receive an error.

```
New-MoveRequest -Identity 'tony@alpineskihouse.com' -
TargetDatabase "DB01" -WhatIf
```

EXAMPLE 2

This example moves Tony Smith's mailbox to the new database DB01.

```
New-MoveRequest -Identity 'tony@alpineskihouse.com' -
TargetDatabase "DB01"
```

EXAMPLE 3

This example creates a batch move request for all mailboxes on the database DB01 and moves them to the database DB02 with the *BatchName* parameter value DB01toDB02.

```
Get-Mailbox -Database DB01 | New-MoveRequest -
TargetDatabase DB02 -BatchName "DB01toDB02"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox moves" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.Mail	The *Identity* parameter specifies the identity of the mailbox or mail user.

		boxOrMailUserIdParameter	You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SMTP address • Alias
Outbound	Required	System.Management.Automation.SwitchParameter	The *Outbound* switch specifies that this mailbox move is a cross-forest move and is being initiated from the source forest. You don't have to specify a value with this parameter. You can't use this parameter in conjunction with the *Remote* switch.
Remote	Required	System.Management.Automation.SwitchParameter	The *Remote* switch specifies that the move you're initiating is outside of your organization, and that this move is being initiated from the target forest. You don't have to specify a value with this parameter.

			You can't use this parameter in conjunction with the *Outbound* switch.
RemoteCredential	Required	System.Management.Automation.PSCredential	The *RemoteCredential* parameter specifies an administrator who has permission to perform the mailbox move, for example, Administrator@humongousinsurance.com. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
RemoteGlobalCatalog	Required	Microsoft.Exchange.Data.Fqdn	The *RemoteGlobalCatalog* parameter specifies the fully qualified domain name (FQDN) of the global catalog server for the remote forest.
RemoteHostName	Required	Microsoft.Exchange.Data.Fqdn	The *RemoteHostName* parameter specifies the FQDN of the cross-forest organization from which you're moving the mailbox.

RemoteLegacy	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemoteLegacy* switch specifies that this mailbox move is from a remote forest that doesn't have Exchange Server 2013 installed. You don't have to specify a value with this parameter.
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51 or higher. Items are considered corrupted if the item can't be read from the source database or can't be written to the target database. Corrupted items won't be available in the destination mailbox or .pst file.
AllowLargeItems	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *AllowLargeItems* switch specifies that items

			larger than the target mailbox limits are copied without failure. You can't use the *AllowLargeItems* switch and the *LargeItemLimit* parameter together in the same command.
ArchiveDomain	Optional	System.String	The *ArchiveDomain* parameter specifies the FQDN of the external domain to which you're moving the archive. This parameter is used for moving the archive to a cloud-based service.
ArchiveOnly	Optional	System.Management.Automation.SwitchParameter	The *ArchiveOnly* parameter specifies that you're moving only the personal archive associated with the mailbox. You can't use this parameter in conjunction with the *PrimaryOnly* parameter.
ArchiveTargetDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArchiveTargetDatabase*

			parameter specifies the Exchange target database to which you're moving the personal archive. If this parameter isn't specified, the archive is moved to the same database as the primary mailbox. You can use the following values for this parameter: • GUID of the database • Database name
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use 0 to not skip bad items. The valid input range for this parameter is from 0 through 2147483647. The default value is 0. We recommend that you keep the default value 0 and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the *BadItemLimit* parameter to more than 50, the command fails, and you

			receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process is complete.
BatchName	Optional	System.String	The *BatchName* parameter specifies a descriptive name for moving a batch of mailboxes. You can then use the name in the *BatchName* parameter as a search string when you use the Get-MoveRequest cmdlet.
CheckInitialProvisioningSetting	Optional	System.Management.Automation.SwitchParameter	In Microsoft Exchange Server 2013, there is a setting, **IsExcludedFromProvisioning** on each mailbox database that allows it to be excluded from provisioning new mailboxes. The *CheckInitialProvisioningSetting* parameter specifies

			the **IsExcludedFromProvisioning** setting when choosing the target database if none is provided.
CompleteAfter	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the request will be kept after it has completed before being automatically removed. The default *CompletedRequestAgeLimit* parameter value is 30 days.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-

			premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
DoNotPreserveMailboxSignature	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *DoNotPreserveMailboxSignature* parameter specifies that the command doesn't preserve the mailbox mapping signature. We recommend that you use this parameter only if the move request fails because the Named Property identifiers are depleted. If you specify this parameter, the mailbox user is required to restart Microsoft Outlook when the move request is complete.
ForceOffline	Optional	System.Management.	The *ForceOffline*

		Automation.SwitchParameter	parameter forces a mailbox move to be performed in offline mode. Moving a mailbox in offline mode means the user will have no access to email during the mailbox move.
ForcePull	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ForcePull* parameter specifies that the type of move should be a Pull move. This parameter can be used for local moves only.
ForcePush	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ForcePush* parameter specifies that the type of move should be a Push move. This parameter can be used for local moves only.
IgnoreRuleLimitErrors	Optional	System.Management.Automation.SwitchParameter	The *IgnoreRuleLimitErrors* parameter specifies that the command doesn't move the user's rules to the target server running

			Exchange.
IgnoreTenantMigrationPolicies	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IncrementalSyncInterval	Optional	System.TimeSpan	The *IncrementalSyncInterval* parameter specifies the wait time between incremental syncs. This parameter is used together with the *StartAfter* and *CompleteAfter* parameters to create a move request that will do periodic incremental syncs after the initial sync is complete.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	This parameter is available only in on-premises Exchange 2013. The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of large items to skip if the request

			encounters such items in the mailbox. Use 0 to not skip any large items. If any number above 50 is specified, the *AcceptLargeDataLoss* parameter must also be specified. The default value is 0. We recommend that you use the default value of 0 and increase the *LargeItemLimit* only when large items are encountered.
PreventCompletion	Optional	System.Management.Automation.SwitchParameter	The *PreventCompletion* parameter specifies that this cmdlet initializes, but isn't completed. This parameter accepts `$true` or `$false`. The default value is `$false`. If you specify `$true`, you have to run the **Resume-MoveRequest** cmdlet to complete the move request.
PrimaryOnly	Optional	System.Management.Automation.SwitchParameter	The *PrimaryOnly* parameter specifies that the command should only move the primary mailbox; the personal archive isn't moved. You

			don't have to specify a value with this parameter. You can't use this parameter in conjunction with the *ArchiveOnly* parameter.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	This parameter is available only in on-premises Exchange 2013. The *Priority* parameter specifies the order in which this request should be processed in the request queue. Requests are processed in order, based on server health, status, priority, and last update time.
Protect	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
RemoteArchiveTargetDatabase	Optional	System.String	The *RemoteArchiveTargetDatabase* parameter specifies the name of the target database in the remote forest to which you're moving the personal archive. Use this parameter when moving users with archives from

			the local forest to a remote forest. For moves from a remote forest to the local forest, use the *ArchiveTargetDatabase* parameter. If you use this parameter, you must specify the *Remote* or *RemoteLegacy* parameter.
RemoteOrganizationName	Optional	System.String	This parameter is reserved for internal Microsoft use.
RemoteTargetDatabase	Optional	System.String	The *RemoteTargetDatabase* parameter specifies the name of the target database in the remote forest. Use this parameter when moving mailboxes from the local forest to a remote forest. For moves from a remote forest to the local forest, use the *TargetDatabase* parameter. If you use this parameter, you must specify the *Remote* or *RemoteLegacy* parameter.
SkipMoving	Optional	Microsoft.Exchange.Management.RecipientT	The *SkipMoving* parameter allows certain

		asks.SkippableMoveComponent[]	stages of a mailbox move to be skipped for debugging purposes. Don't use this parameter unless directed to do so by a support professional or specific documentation.
StartAfter	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.
Suspend	Optional	System.Management.Automation.SwitchParameter	The *Suspend* switch specifies whether to suspend the request. If you use this switch, the request is queued, but the request won't reach the status of **InProgress** until you resume the request with the relevant resume cmdlet. You don't have to specify a value with this switch.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description about why the request was suspended. You can only use this parameter if you specify the *Suspend* parameter.
SuspendWhenReadyToComplete	Optional	System.Management.Automation.SwitchParameter	The *SuspendWhenReadyToComplete* switch specifies

			whether to suspend the move request before it reaches the status of **CompletionInProgress**. After the move is suspended, it has a status of **AutoSuspended**. You can then manually complete the move by using the **Resume-MoveRequest** command. **Note:** You can only use the *SuspendWhenReadyToComplete* switch for online mailbox moves and when moving mailboxes from Exchange Server 2007 or Exchange 2010 mailbox databases.
TargetDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *TargetDatabase* parameter specifies the identity of the database that you're moving the mailbox to. If you don't specify the *TargetDatabase* parameter, the command uses the automatic mailbox distribution logic to determine the database to move to.

			You can use the following values: • GUID of the database • Database name
TargetDeliveryDomain	Optional	Microsoft.Exchange.Data.Fqdn	The *TargetDeliveryDomain* parameter specifies the FQDN of the external email address created in the source forest for the mail-enabled user when the move request is complete. This parameter is allowed only when performing remote moves with the *Remote* or *RemoteLegacy* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. The *WhatIf* switch can also be used to test a mailbox's readiness to be moved. By using the *WhatIf* switch, you can view any errors that will occur without adding the mailbox to the move request queue. You don't have to specify a value

			with the *WhatIf* switch.
WorkloadType	Optional	Microsoft.Exchange.MailboxReplicationService.RequestWorkloadType	This parameter is available only in on-premises Exchange 2013. This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MoveRequest** cmdlet to cancel a mailbox move initiated using the New-MoveRequest cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MoveRequest -Identity <MoveRequestIdParameter> <COMMON PARAMETERS>

Remove-MoveRequest -MailboxGuid <Guid> -MoveRequestQueue
<DatabaseIdParameter> <COMMON PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the mailbox move request for Ayla Kol's mailbox.

```
Remove-MoveRequest -Identity 'Ayla@humongousinsurance.com'
```

EXAMPLE 2

This example cancels a mailbox move for a mailbox by using the *MailboxGuid* parameter for a mailbox on MBXDB01.

Note:

The *MailboxGuid* and *MoveRequestQueue* parameters are for debugging purposes only.

```
Remove-MoveRequest -MoveRequestQueue MBXDB01 -MailboxGuid
25e0eaf2-6cc2-4353-b83e-5cb7b72d441f
```

Detailed Description

The *MoveRequestQueue* and *MailboxGuid* parameters are for debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox moves" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MoveRequestIdParameter	The *Identity* parameter specifies the identity of the mailbox or mail user. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account*

			• User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias You can't use this parameter in conjunction with the *MailboxGuid* or *MoveRequestQueue* parameters.
MailboxGuid	Required	System.Guid	This parameter is available only in on-premises Exchange 2013. The *MailboxGuid* parameter specifies the GUID of the mailbox for which you want to remove the move request. If you specify the *MailboxGuid* parameter, you must also specify the *MoveRequestQueue* parameter. You can't use this parameter in conjunction with the *Identity* parameter.
MoveRequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *MoveRequestQueue*

			parameter specifies the database on which the move request is queued. You can use the following values: • GUID of the database • Database name You can't use this parameter in conjunction with the *Identity* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Resume-MoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Resume-MoveRequest** cmdlet to resume a move request that has been suspended or has failed.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Resume-MoveRequest -Identity <MoveRequestIdParameter> [-
SuspendWhenReadyToComplete <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resumes the move request of Tony Smith's mailbox.

```
Resume-MoveRequest -Identity "Tony@contoso.com"
```

EXAMPLE 2

This example resumes any failed move requests.

```
Get-MoveRequest -MoveStatus Failed | Resume-MoveRequest
```

EXAMPLE 3

This example resumes any move requests that have the suspend comment "Resume after 10 P.M."

```
Get-MoveRequest -MoveStatus Suspended | Get-
MoveRequestStatistics |Where {$_.Message -like "*resume
after 10 P.M."} | Resume-MoveRequest
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox moves" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MoveRequestIdParameter	The *Identity* parameter specifies the identity of the mailbox or mail user. You can use the following values:

			• GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SMTP address • Alias
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$false`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
SuspendWhenReadyToComplete	Optional	System.Management.Automation.SwitchPar	The *SuspendWhenReadyToCo*

		ameter	*mplete* parameter specifies whether to return mailboxes that have been moved with the **New-MoveRequest** command and its *SuspendWhenReadyToComplete* switch. This parameter accepts `$true` or `$false`. You can't use this parameter in conjunction with the *Identity* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MoveRequest** cmdlet to change move request options after the move request has been created. You can use the **Set-MoveRequest** cmdlet to recover from failed move requests.

```
Set-MoveRequest -Identity <MoveRequestIdParameter> [-AcceptLargeDataLoss
<SwitchParameter>] [-ArchiveTargetDatabase <DatabaseIdParameter>] [-
BadItemLimit <Unlimited>] [-BatchName <String>] [-CompleteAfter
<DateTime>] [-CompletedRequestAgeLimit <Unlimited>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreRuleLimitErrors
<$true | $false>] [-IncrementalSyncInterval <TimeSpan>] [-InternalFlags
<InternalMrsFlag[]>] [-LargeItemLimit <Unlimited>] [-PreventCompletion
<$true | $false>] [-Priority <Lowest | Lower | Low | Normal | High |
Higher | Highest | Emergency>] [-Protect <$true | $false>] [-
RemoteCredential <PSCredential>] [-RemoteGlobalCatalog <Fqdn>] [-
RemoteHostName <Fqdn>] [-SkipMoving <SkippableMoveComponent[]>] [-
StartAfter <DateTime>] [-SuspendWhenReadyToComplete <$true | $false>] [-
TargetDatabase <DatabaseIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the move request for Ayla to accept up to five corrupt mailbox items.

```
Set-MoveRequest -Identity Ayla@humongousinsurance.com -
BadItemLimit 5
```

Detailed Description

You can pipeline the **Set-MoveRequest** cmdlet from the **Get-MoveRequestStatistics**, **Get-MoveRequest**, or **Get-Mailbox** cmdlets.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox

moves" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MoveRequestIdParameter	The *Identity* parameter specifies the identity of the mailbox or mail user. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SMTP address • Alias
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51 or higher. Items are considered corrupted if the item can't be read from the source database or can't be written to the target database. Corrupted items won't be available in the destination mailbox or .pst file.

ArchiveTargetDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArchiveTargetDatabase* parameter specifies the Exchange target database to which you're moving the personal archive. You can use this parameter to change the target database only if the move request has a **MoveStatus** value of Queued. You can use the following values to specify the target database: • GUID of the database • Database name
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use 0 to not skip bad items. The valid input range for this parameter is from 0 through 2147483647. The default value is 0. We recommend that you keep

			the default value 0 and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the *BadItemLimit* parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process is complete.
BatchName	Optional	System.String	The *BatchName* parameter specifies a different name for a batch.
CompleteAfter	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the request is kept after it has completed before being automatically removed.

			The default *CompletedRequestAgeLimit* parameter value is 30 days.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreRuleLimitErrors	Optional	System.Boolean	The *IgnoreRuleLimitErrors* parameter specifies that the command won't move the user's rules to the target server running Microsoft Exchange.

IncrementalSyncInterval	Optional	System.TimeSpan	The *IncrementalSyncInterval* parameter specifies the wait time between incremental syncs. This parameter is used together with the *StartAfter* and *CompleteAfter* parameters to create a move request that will do periodic incremental syncs after the initial sync is complete.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	This parameter is available only in on-premises Exchange 2013. The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of large items to skip if the request encounters such items in the mailbox. Use 0 to not skip any large items. If any number above 50 is specified, the

			AcceptLargeDataLoss parameter must also be specified. The default value is 0. We recommend that you use the default value of 0 and increase the *LargeItemLimit* parameter value only when large items are encountered.
PreventCompletion	Optional	System.Boolean	The *PreventCompletion* parameter specifies that this cmdlet initializes, but isn't completed. This parameter accepts `$true` or `$false`. The default value is `$false`. If you specify `$true`, you have to run the **Resume-MoveRequest** cmdlet to complete the move request.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	This parameter is available only in on-premises Exchange 2013. The *Priority* parameter specifies the order in which this request should be processed in the request queue. Requests are processed in order, based on server health,

			status, priority, and last update time.
Protect	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
RemoteCredential	Optional	System.Management.Automation.PSCredential	The *RemoteCredential* parameter specifies an administrator who has permission to perform the mailbox move, for example, Administrator@humongousinsurance.com. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
RemoteGlobalCatalog	Optional	Microsoft.Exchange.Data.Fqdn	The *RemoteGlobalCatalog* parameter specifies the FQDN of the global catalog server for the remote forest.
RemoteHostName	Optional	Microsoft.Exchange.Data.Fqdn	The *RemoteHostName* parameter specifies the FQDN of the cross-forest organization from which you're moving the mailbox.

SkipMoving	Optional	Microsoft.Exchange.Management.RecipientTasks.SkippableMoveComponent[]	The *SkipMoving* parameter allows certain stages of a mailbox move to be skipped for debugging purposes. Do not use this parameter unless directed to do so by a support professional or specific documentation.
StartAfter	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.
SuspendWhenReadyToComplete	Optional	System.Boolean	The *SuspendWhenReadyToComplete* parameter specifies whether to suspend the move request before it reaches the status of CompletionInProgress. The move request then has a status of AutoSuspended. You can manually complete the move by using the **Resume-MoveRequest** command.
TargetDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *TargetDatabase* parameter specifies the

			identity of the database that you're moving the mailbox to. You can use this parameter to change the target database only if the move request has a **MoveStatus** value of `Queued`. You can use the following values: • GUID of the database • Database name
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Suspend-MoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Suspend-MoveRequest** cmdlet to suspend a move request any time after the move request was created, but before it reaches the status of CompletionInProgress. You can resume the move request by using the Resume-MoveRequest cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Suspend-MoveRequest -Identity <MoveRequestIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-SuspendComment <String>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example suspends the move request for Ayla's mailbox.

```
Suspend-MoveRequest -Identity "Ayla@humongousinsurance.com"
```

EXAMPLE 2

This example suspends all move requests that are in progress by using the Get-MoveRequest cmdlet to retrieve all move requests with a *MoveStatus* value of `InProgress`, and then pipelining the output to the **Suspend-MoveRequest** cmdlet.

```
Get-MoveRequest -MoveStatus InProgress | Suspend-
MoveRequest
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox moves" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MoveRequestIdParameter	The *Identity* parameter specifies the identity of the mailbox or mail user. You can use one of the following values: • GUID • Distinguished Name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SMTP address • Alias
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description as to why the request was suspended.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn’t accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn’t return data.

Get-MoveRequestStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MoveRequestStatistics** cmdlet to view detailed information about move requests.

Note:

Some of the failure messages that are returned by this cmdlet are temporary and don't indicate that a request has actually failed. If the **Status** value is `Queued` or `InProgress`, then the request is proceeding normally.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MoveRequestStatistics -Identity <MoveRequestIdParameter> <COMMON
PARAMETERS>


Get-MoveRequestStatistics -MoveRequestQueue <DatabaseIdParameter> [-
MailboxGuid <Guid>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Diagnostic <SwitchParameter>] [-DomainController
<Fqdn>] [-IncludeReport <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns the default statistics for Tony Smith's mailbox, which include the status, mailbox size, archive mailbox size, and the percentage complete.

```
Get-MoveRequestStatistics -Identity Tony@contoso.com
```

EXAMPLE 2

This example returns the detailed statistics for Tony Smith's mailbox by pipelining the results to the **Format-List** command.

```
Get-MoveRequestStatistics -Identity "contoso\tony" |
Format-List
```

EXAMPLE 3

This example returns additional information about the mailbox move for Tony Smith's mailbox and exports the report to a .csv file.

```
Get-MoveRequestStatistics -Identity Tony@contoso.com -
IncludeReport | Export-CSV C:\MRStats.csv
```

EXAMPLE 4

This example returns default statistics for all mailboxes whose move requests are in progress or haven't been cleared for the database MBXDB02.

```
Get-MoveRequestStatistics -MoveRequestQueue "MBXDB02"
```

Detailed Description

The *MoveRequestQueue* parameter syntax set is for debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox moves" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.MoveRequestIdParameter	The *Identity* parameter specifies the identity of the mailbox or mail user. You can use one of the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias

			This parameter can't be used with the *MoveRequestQueue* or *MailboxGuid* parameters.
MoveRequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *MoveRequestQueue* parameter specifies the mailbox database on which the move request resides. You can use one of the following values: • GUID of the database • Database name This parameter can't be used with the *Identity* or *MRSInstance* parameters.
Diagnostic	Optional	System.Management.Automation.SwitchParameter	The *Diagnostic* switch specifies whether to retrieve extremely detailed information about the mailbox import request.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that retrieves data from Active Directory.
IncludeReport	Optional	System.Management.Automation.SwitchParameter	The *IncludeReport* switch specifies whether to return additional details, which can be used for troubleshooting.
MailboxGuid	Optional	System.Guid	This parameter is available only in on-premises Exchange 2013. The *MailboxGuid* parameter specifies the GUID of a mailbox for which you want to view the move request statistics. This parameter can't be used with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PublicFolderMigrationRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-PublicFolderMigrationRequest** cmdlet to view the detailed status of on-going public folder migration requests that were initiated by using the **New-PublicFolderMigrationRequest** cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PublicFolderMigrationRequest [-AccountPartition
<AccountPartitionIdParameter>] [-Identity
<PublicFolderMigrationRequestIdParameter>] [-Organization
<OrganizationIdParameter>] <COMMON PARAMETERS>


Get-PublicFolderMigrationRequest [-AccountPartition
<AccountPartitionIdParameter>] [-BatchName <String>] [-HighPriority <$true
| $false>] [-Name <String>] [-Organization <OrganizationIdParameter>] [-
RequestQueue <DatabaseIdParameter>] [-Status <None | Queued | InProgress |
AutoSuspended | CompletionInProgress | Synced | Completed |
CompletedWithWarning | Suspended | Failed>] [-Suspend <$true | $false>]
<COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns the status of in-progress migration requests.

```
Get-PublicFolderMigrationRequest
```

EXAMPLE 2

This example returns all migration requests that have the name PFMigrate10_11_12, and the request has been suspended.

```
Get-PublicFolderMigrationRequest -Identity
"PFMigrate10_11_12" | Format-List Suspended,AutoSuspended
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
BatchName	Optional	System.String	The *BatchName* parameter specifies the name that was given to a batch migration request. You can't use this parameter in conjunction with the *Identity* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
HighPriority	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013.

			The *HighPriority* parameter specifies that the command returns all migration requests with a status of *HighPriority*.
Identity	Optional	Microsoft.Exchange.MailboxReplicationService.PublicFolderMigrationRequestIdParameter	The *Identity* parameter specifies the identity of the public folder. You can use the following values: • GUID • Name You can't use this parameter in conjunction with the following parameters: • *BatchName* • *RequestQueue* • *Status* • *Suspend*
Name	Optional	System.String	The *Name* parameter specifies the name of the public folder migration request.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
RequestQueue	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *RequestQueue* parameter specifies the

			identity of the mailbox database on which the migration request has been run. You can't use this parameter in conjunction with the *Identity* parameter.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Status	Optional	Microsoft.Exchange.Data.Directory.Recipient.RequestStatus	The *Status* parameter returns migration requests in the specified status. You can use the following values: • `AutoSuspended` • `Completed` • `CompletedWithWarning` • `CompletionInProgress` • `Failed` • `InProgress` • `None` • `Queued` • `Suspended` You can't use this parameter in conjunction with the *Identity* parameter.

Suspend	Optional	System.Boolean	The *Suspend* parameter specifies whether to return public folder migration requests that have been suspended. This parameter accepts `$true` or `$false`. You can't use this parameter in conjunction with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-PublicFolderMigrationRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-02*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-PublicFolderMigrationRequest** cmdlet to begin the process of migrating public folders from Microsoft Exchange Server 2007 or Exchange Server 2010 to Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-PublicFolderMigrationRequest -SourceDatabase <DatabaseIdParameter>
<COMMON PARAMETERS>
```

```
New-PublicFolderMigrationRequest -OutlookAnywhereHostName <Fqdn> -
RemoteCredential <PSCredential> -RemoteMailboxLegacyDN <String> -
RemoteMailboxServerLegacyDN <String> [-AuthenticationMethod <Basic |
Digest | Ntlm | Fba | WindowsIntegrated | LiveIdFba | LiveIdBasic |
WSSecurity | Certificate | NegoEx | OAuth | Adfs | Kerberos | Negotiate |
LiveIdNegotiate | Misconfigured>] [-Organization
<OrganizationIdParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-AcceptLargeDataLoss <SwitchParameter>] [-BadItemLimit
<Unlimited>] [-BatchName <String>] [-CompletedRequestAgeLimit <Unlimited>]
[-Confirm [<SwitchParameter>]] [-CSVData <Byte[]>] [-CSVStream <Stream>]
[-DomainController <Fqdn>] [-InternalFlags <InternalMrsFlag[]>] [-
LargeItemLimit <Unlimited>] [-Name <String>] [-Priority <Lowest | Lower |
Low | Normal | High | Higher | Highest | Emergency>] [-SkipMerging
<SkippableMergeComponent[]>] [-Suspend <SwitchParameter>] [-SuspendComment
<String>] [-WhatIf [<SwitchParameter>]] [-WorkloadType <None | Local |
Onboarding | Offboarding | TenantUpgrade | LoadBalancing | Emergency |
RemotePstIngestion | SyncAggregation>]
```

Examples

EXAMPLE 1

This example creates a public folder migration request from the Exchange 2010 source public folder database PFDB01 and uses the CSVData.csv file that was created using the **Export-PublicFolderStatistics.ps1** script. For more information, see Migrate public folders to Exchange 2013 from previous versions.

```
New-PublicFolderMigrationRequest -SourceDatabase PFDB01 -
CSVData (Get-Content C:\PFMigration\CSVData.csv -Encoding
Byte)
```

Detailed Description

Migrating public folders is a multi-step process. For more information before you attempt a public folder migration, see Migrate public folders to Exchange 2013 from previous versions.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
OutlookAnywhereHostName	Required	Microsoft.Exchange.Data.Fqdn	This parameter is reserved for internal Microsoft use.

RemoteCredential	Required	System.Management.Automation.PSCredential	This parameter is available only in the cloud-based service. The *RemoteCredential* parameter specifies an administrator who has permission to perform the migration request, for example, Administrator@humongousinsurance.com. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential. You must use this parameter in conjunction with the *RemoteMailboxServerLegacyDN* parameter.
RemoteMailboxLegacyDN	Required	System.String	This parameter is available only in the cloud-based service. The *RemoteMailboxLegacyDN* parameter specifies the mailbox of the remote credentials specified in

			the *RemoteCredential* parameter. You must use this parameter in conjunction with the *RemoteMailboxServerLegacyDN* parameter.
RemoteMailboxServerLegacyDN	Required	System.String	This parameter is available only in the cloud-based service. The *RemoteMailboxServerLegacyDN* parameter specifies the server legacy distinguished name (DN) of the back-end server. To find the **LegacyExchangeServerDN** property, run the following command: `Get-ExchangeServer <Identity> | Format-List LegacyExchangeServerDN`.
SourceDatabase	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *SourceDatabase* parameter specifies the identity of the database on which the public folders that are being migrated resides. You can

			use the following values: • GUID of the database • Database name
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51 or higher. Items are considered corrupted if the item can't be read from the source database or can't be written to the target database. Corrupted items won't be available in the destination mailbox or .pst file.
AuthenticationMethod	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthenticationMethod	This parameter is reserved for internal Microsoft use.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use `0` to not skip bad items. The valid input range for this parameter is from `0` through `2147483647`. The

			default value is 0. We recommend that you keep the default value 0 and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the *BadItemLimit* parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process is complete.
BatchName	Optional	System.String	The *BatchName* parameter specifies a descriptive name for the public folder migration batch. You can use the *BatchName* parameter as a search string when you use the **Get-PublicFolderMigrationRequest** cmdlet.
CompletedRequestAg	Optional	Microsoft.Exchange.Da	The

eLimit		ta.Unlimited	*CompletedRequestAgeLimit* parameter specifies how long the request is kept after it has completed before being automatically removed. The default *CompletedRequestAgeLimit* parameter value is 30 days.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
CSVData	Optional	System.Byte[]	The *CSVData* parameter specifies the mapping file output generated by the **PublicFoldertoMailboxMapGenerator.ps1** script. Use this parameter for local migrations. This parameter can't be used in conjunction with the *CSVStream* parameter. You must use this parameter if you don't

			use *CSVStream* parameter.
CSVStream	Optional	System.IO.Stream	The *CSVStream* parameter specifies the mapping file output generated by the **PublicFoldertoMailboxMapGenerator.ps1** script. Use this parameter for remote migrations. This parameter can't be used in conjunction with the *CSVData* parameter. You must use this parameter if you don't use *CSVData* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for

			debugging purposes.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of large items to skip if the request encounters such items in the mailbox. Use 0 to not skip any large items. If any number above 50 is specified, the *AcceptLargeDataLoss* parameter must also be specified. The default value is 0. We recommend that you use the default value of 0 and increase the *LargeItemLimit* parameter value only when large items are encountered.
Name	Optional	System.String	The *Name* parameter specifies the name of the public folder migration request.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	This parameter is available only in on-premises Exchange 2013. The *Priority* parameter

			specifies the priority setting that you want to use for the migration to be completed. This prioritizes against any Microsoft Exchange Mailbox Replication service (MRS) process. This parameter accepts the following values: • `Lowest` • `Lower` • `Low` • `Normal` • `High` • `Higher` • `Highest` • `Emergency` If you don't specify a value, the default value is `Normal`, which means that the request is prioritized by the time and date it was accepted into the MRS queue.
SkipMerging	Optional	Microsoft.Exchange.Management.RecipientTasks.SkippableMergeComponent[]	The *SkipMerging* parameter specifies whether certain stages of a public folder migration are skipped for debugging purposes. Don't use this parameter unless directed to do so by a Microsoft Customer Service and Support or specific documentation.

Suspend	Optional	System.Management.Automation.SwitchParameter	The *Suspend* switch specifies whether to suspend the request. If you use this switch, the request is queued, but the request won't reach the status of **InProgress** until you resume the request with the relevant resume cmdlet. You don't have to specify a value with this switch.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description about why the request was suspended. You can only use this parameter if you specify the *Suspend* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

WorkloadType	Optional	Microsoft.Exchange.MailboxReplicationService.RequestWorkloadType	This parameter is available only in on-premises Exchange 2013. The *WorkloadType* parameter specifies the type of request and the purpose for being performed. This information is used by Exchange. This parameter accepts the following values: • `None` • `Local` • `Onboarding` • `Offboarding` • `TenantUpgrade` • `LoadBalancing` • `Emergency`

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-PublicFolderMigrationRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-PublicFolderMigrationRequest** cmdlet to cancel or complete a migration request that was initiated using the **New-PublicFolderMigrationRequest** cmdlet. You use must this cmdlet to remove the public folder migration request before you can create another public folder migration request.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-PublicFolderMigrationRequest -Identity
<PublicFolderMigrationRequestIdParameter> <COMMON PARAMETERS>


Remove-PublicFolderMigrationRequest -RequestGuid <Guid> -RequestQueue
<DatabaseIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example cancels any public folder migration request that's actively running.

```
Get-PublicFolderMigrationRequest | Remove-
PublicFolderMigrationRequest
```

EXAMPLE 2

This example cancels a migration request by using the *RequestGuid* parameter for a mailbox on MBXDB01.

Note:

The *RequestGuid* and *RequestQueue* parameters are for debugging purposes only.

```
Remove-PublicFolderMigrationRequest -RequestQueue MBXDB01 -
RequestGuid 25e0eaf2-6cc2-4353-b83e-5cb7b72d441f
```

Detailed Description

The *RequestQueue* and *RequestGuid* parameters are for debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.PublicFolderMigrationRequestIdParameter	The *Identity* parameter specifies the identity of the public folder migration request. You can't use this parameter in conjunction with the *RequestGuid* or *RequestQueue* parameters.
RequestGuid	Required	System.Guid	This parameter is available only in on-premises Exchange 2013. The *RequestGuid* parameter specifies the GUID of the migration request. If you specify the *RequestGuid* parameter, you must also specify the *RequestQueue* parameter. You can't use this parameter in conjunction with the *Identity* parameter.
RequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *RequestQueue* parameter specifies the database on which the migration request is

			queued. If you specify the *RequestQueue* parameter, you must also specify the *RequestGuid* parameter. You can use the following values: • GUID of the database • Database name You can't use this parameter in conjunction with the *Identity* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration

			change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Resume-PublicFolderMigrationRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Resume-PublicFolderMigrationRequest** cmdlet to resume a migration request that failed

or has been suspended or auto-suspended.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Resume-PublicFolderMigrationRequest -Identity
<PublicFolderMigrationRequestIdParameter> [-Confirm [<SwitchParameter>]]
[-DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resumes the migration request PublicFolderMigration.

```
Resume-PublicFolderMigrationRequest -Identity
"PublicFolderMigration"
```

EXAMPLE 2

This example resumes any failed migration requests.

```
Get-PublicFolderMigrationRequest -Status Failed | Resume-
PublicFolderMigrationRequest
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.PublicFolderMigrationRequestIdParameter	The *Identity* parameter specifies the identity of the migration request. This parameter accepts the following values: • GUID • Name
Confirm	Optional	System.Management.	The *Confirm* switch can be

		Automation.SwitchParameter	used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-PublicFolderMigrationRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-PublicFolderMigrationRequest** cmdlet to change migration request options after the request has been created. You can use the **Set-PublicFolderMigrationRequest** cmdlet to recover from failed migration requests.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-PublicFolderMigrationRequest [-AcceptLargeDataLoss <SwitchParameter>]
[-BadItemLimit <Unlimited>] [-BatchName <String>] [-
CompletedRequestAgeLimit <Unlimited>] [-InternalFlags <InternalMrsFlag[]>]
[-LargeItemLimit <Unlimited>] [-Priority <Lowest | Lower | Low | Normal |
High | Higher | Highest | Emergency>] [-SkipMerging
<SkippableMergeComponent[]>] <COMMON PARAMETERS>


Set-PublicFolderMigrationRequest -RemoteCredential <PSCredential> <COMMON
PARAMETERS>


Set-PublicFolderMigrationRequest -RehomeRequest <SwitchParameter> <COMMON
PARAMETERS>


COMMON PARAMETERS: -Identity <PublicFolderMigrationRequestIdParameter> [-
AuthenticationMethod <Basic | Digest | Ntlm | Fba | WindowsIntegrated |
LiveIdFba | LiveIdBasic | WSSecurity | Certificate | NegoEx | OAuth | Adfs
| Kerberos | Negotiate | LiveIdNegotiate | Misconfigured>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-OutlookAnywhereHostName
<Fqdn>] [-PreventCompletion <$true | $false>] [-RemoteMailboxLegacyDN
<String>] [-RemoteMailboxServerLegacyDN <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the setting of the PublicFolderMigration migration request to accept up to five corrupted public folder items.

```
Set-PublicFolderMigrationRequest -Identity
PublicFolderMigration -BadItemLimit 5
```

Detailed Description

You can pipeline the **Set-PublicFolderMigrationRequest** cmdlet from the **Get-PublicFolderMigrationRequestStatistics**, **Get-PublicFolderMigrationRequst**, or the **Get-PublicFolder** cmdlets.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.PublicFolderMigrationRequestIdParameter	The *Identity* parameter specifies the identity of the public folder migration request.
RehomeRequest	Required	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RehomeRequest* parameter specifies to the Microsoft Exchange Mailbox Replication service (MRS) that the

			request needs to be moved to the same database as the public folder being migrated. This parameter is used primarily for debugging purposes.
RemoteCredential	Required	System.Management.Automation.PSCredential	This parameter is available only in the cloud-based service. The *RemoteCredential* parameter specifies an administrator who has permission to perform the migration request, for example, Administrator@humongousinsurance.com. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51

			or higher. Items are considered corrupted if the item can't be read from the source database or can't be written to the target database. Corrupted items won't be available in the destination mailbox or .pst file.
AuthenticationMethod	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthenticationMethod	This parameter is reserved for internal Microsoft use.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use `0` to not skip bad items. The valid input range for this parameter is from `0` through `2147483647`. The default value is `0`. We recommend that you keep the default value `0` and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the

			BadItemLimit parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process is complete.
BatchName	Optional	System.String	The *BatchName* parameter specifies a descriptive name for the public folder batch migration. You can use the *BatchName* parameter as a search string when you use the **Get-PublicFolderMigrationRequest** cmdlet.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the request is kept after it has completed before being automatically removed. The default

			CompletedRequestAgeLimit parameter value is 30 days.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
LargeItemLimit	Optional	Microsoft.Exchange.Da	The *LargeItemLimit*

		ta.Unlimited	parameter specifies the number of large items to skip if the request encounters such items in the mailbox. Use 0 to not skip any large items. If any number above 50 is specified, the *AcceptLargeDataLoss* parameter must also be specified. The default value is 0. We recommend that you use the default value of 0 and increase the *LargeItemLimit* parameter value only when large items are encountered.
OutlookAnywhereHostName	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is reserved for internal Microsoft use.
PreventCompletion	Optional	System.Boolean	The *PreventCompletion* parameter specifies that this cmdlet initializes, but isn't completed. This parameter accepts `$true` or `$false`. The default value is `$false`. If you specify `$true`, you have to run the **Resume-PublicFolderMigrationRequest** cmdlet to complete the migration

			request.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	This parameter is available only in on-premises Exchange 2013. The *Priority* parameter specifies the order in which this request should be processed in the request queue. Requests are processed in order, based on server health, status, priority, and last update time. This parameter accepts the following values: • `Lowest` • `Lower` • `Low` • `Normal` • `High` • `Higher` • `Highest` • `Emergency` If you don't specify a value, the default value is `Normal`, which means that the request is prioritized by the time and date it was accepted into the queue.
RemoteMailboxLegacyDN	Optional	System.String	This parameter is available only in the cloud-based service. The *RemoteMailboxLegacyDN*

			parameter specifies the **ExchangeLegacyDN** of the remote mailbox.
RemoteMailboxServerLegacyDN	Optional	System.String	This parameter is available only in the cloud-based service. The *RemoteMailboxServerLegacyDN* parameter specifies the server legacy DN of the back-end server. To find the **LegacyExchangeServerDN** value, run the following command: `Get-ExchangeServer <Identity> | Format-List LegacyExchangeServerDN`.
SkipMerging	Optional	Microsoft.Exchange.Management.RecipientTasks.SkippableMergeComponent[]	The *SkipMerging* parameter specifies whether certain stages of a migration are to be skipped for debugging purposes. Don't use this parameter unless directed to do so by Microsoft Customer Service and Support or specific documentation.
WhatIf	Optional	System.Management.Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Suspend-PublicFolderMigrationRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Suspend-PublicFolderMigrationRequest** cmdlet to suspend a migration request any time after the request was created, but before it reaches the status of *CompletionInProgress*. You can resume the migration request by using the **Resume-PublicFolderMigrationRequest** cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Suspend-PublicFolderMigrationRequest -Identity
<PublicFolderMigrationRequestIdParameter> [-Confirm [<SwitchParameter>]]
[-DomainController <Fqdn>] [-SuspendComment <String>] [-WhatIf
```

```
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example suspends the public folder migration request PFMigReq1.

```
Suspend-PublicFolderMigrationRequest -Identity PFMigReq1
```

EXAMPLE 2

This example suspends all migration requests that are in progress by using the Get-PublicFolderMigrationRequest cmdlet to retrieve all migration requests with a *Status* value of `InProgress`, and then pipelining the output to the **Suspend-PublicFolderMigrationRequest** cmdlet.

```
Get-PublicFolderMigrationRequest -Status InProgress |
Suspend-PublicFolderMigrationRequest
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.PublicFolderMigrationRequestIdParameter	The *Identity* parameter specifies the identity of the migration request. This parameter accepts the following values: • GUID • Name
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that

			appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description as to why the request was suspended.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PublicFolderMigrationRequestStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-PublicFolderMigrationRequestStatistics** cmdlet to view detailed information about migration requests.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PublicFolderMigrationRequestStatistics -Identity
<PublicFolderMigrationRequestIdParameter> <COMMON PARAMETERS>


Get-PublicFolderMigrationRequestStatistics -RequestQueue
<DatabaseIdParameter> [-RequestGuid <Guid>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Diagnostic <SwitchParameter>] [-DomainController
<Fqdn>] [-IncludeReport <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns default statistics for all migration requests that are in progress or haven't been cleared for the database MBXDB02.

```
Get-PublicFolderMigrationRequestStatistics -RequestQueue
"MBXDB02"
```

EXAMPLE 2

This example returns additional information about the migration request and exports the report to a CSV file.

```
Get-PublicFolderMigrationRequestStatistics -Identity
"PFMigrate_MarketingReports" -IncludeReport | Export-CSV C:
\PFMigstats.csv
```

Detailed Description

The *RequestQueue* parameter is for debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.PublicFolderMigrationRequestIdParameter	The *Identity* parameter specifies the identity of the public folder migration request. You can use one of the following values: • GUID • Name This parameter can't be used in conjunction with

			the *RequestQueue* or *RequestGuid* parameter.
RequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *RequestQueue* parameter specifies the mailbox database on which the migration request resides. You can use one of the following values: • GUID of the database • Database name This parameter can't be used in conjunction with the *Identity* parameter.
Diagnostic	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

IncludeReport	Optional	System.Management.Automation.SwitchParameter	The *IncludeReport* switch specifies whether to return additional details, which can be used for troubleshooting.
RequestGuid	Optional	System.Guid	This parameter is available only in on-premises Exchange 2013. The *RequestGuid* parameter specifies the GUID of a migration request. This parameter can't be used in conjunction with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PublicFolderMoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-PublicFolderMoveRequest** cmdlet to view the detailed status of an ongoing public folder move that was initiated using the New-PublicFolderMoveRequest cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PublicFolderMoveRequest [-AccountPartition
<AccountPartitionIdParameter>] [-Identity
<PublicFolderMoveRequestIdParameter>] [-Organization
<OrganizationIdParameter>] <COMMON PARAMETERS>


Get-PublicFolderMoveRequest [-AccountPartition
<AccountPartitionIdParameter>] [-BatchName <String>] [-HighPriority <$true
| $false>] [-Name <String>] [-Organization <OrganizationIdParameter>] [-
RequestQueue <DatabaseIdParameter>] [-Status <None | Queued | InProgress |
AutoSuspended | CompletionInProgress | Synced | Completed |
CompletedWithWarning | Suspended | Failed>] [-Suspend <$true | $false>]
<COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns the status of an in-progress public folder move request with the identity \PublicFolderMove, which is the default name assigned to public folder moves.

```
Get-PublicFolderMoveRequest -Identity \PublicFolderMove
```

EXAMPLE 2

This example returns the status of in-progress and queued requests that are on the source database MBD01.

```
Get-PublicFolderMoveRequest -RequestQueue MBD01
```

Detailed Description

Public folder move requests are used to move public folders between public folder mailboxes. After the move request is complete, you need to update the hierarchy using the Update-PublicFolderMailbox cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
BatchName	Optional	System.String	The *BatchName* parameter specifies the name that was given to a batch public folder move request. You can't use this parameter in conjunction with the *Identity* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
HighPriority	Optional	System.Boolean	The *HighPriority* parameter specifies that the cmdlet returns requests that were created with the *HighPriority* flag. The *HighPriority* flag indicates that the request should be processed before other lower priority requests in the queue.

			You can't use this parameter in conjunction with the *Identity* parameter.
Identity	Optional	Microsoft.Exchange.MailboxReplicationService.PublicFolderMoveRequestIdParameter	The *Identity* parameter specifies the identity of the public folder move request. The default identity assigned to public folder move requests is `\PublicFolderMove`. This parameter can't be used in conjunction with the following parameters: • *BatchName* • *HighPriority* • *Name* • *RequestQueue* • *Suspend* • *Status*
Name	Optional	System.String	The *Name* parameter specifies the name of the public folder move request. If you didn't specify a name when creating the move request, `PublicFolderMove` will be the default name assigned to the request. You can't use this parameter in conjunction

			with the *Identity* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
RequestQueue	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies the target mailbox database on which the public folder of the request resides. You can use one of the following values: • GUID of the database • Database name You can't use this parameter in conjunction with the *Identity* parameter.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Status	Optional	Microsoft.Exchange.Data.Directory.Recipient.RequestStatus	The *Status* parameter specifies whether to return public folders that are in a specific status.

			This parameter accepts the following values: • `None` • `Queued` • `InProgress` • `AutoSuspended` • `CompletionInProgress` • `Completed` • `CompletedWithWarning` • `Suspended` • `Failed` You can't use this parameter in conjunction with the *Identity* parameter.
Suspend	Optional	System.Boolean	The *Suspend* parameter specifies whether to return public folders with moves that have been suspended. This parameter accepts `$true` or `$false`. You can't use this parameter in conjunction with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-PublicFolderMoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-PublicFolderMoveRequest** cmdlet to begin the process of moving public folder contents between public folder mailboxes. Moving public folders only moves the physical contents of the public folder; it doesn't change the logical hierarchy. When the move request is completed, you must run the **Remove-PublicFolderMoveRequest** cmdlet to remove the request or wait until the time specified in the *CompletedRequestAgeLimit* parameter has passed. The request must be removed before you can run another move request.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-PublicFolderMoveRequest -Folders <PublicFolderIdParameter[]> -
TargetMailbox <MailboxIdParameter> [-AcceptLargeDataLoss
<SwitchParameter>] [-AllowLargeItems <SwitchParameter>] [-BadItemLimit
<Unlimited>] [-CompletedRequestAgeLimit <Unlimited>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-InternalFlags
<InternalMrsFlag[]>] [-LargeItemLimit <Unlimited>] [-Name <String>] [-
Organization <OrganizationIdParameter>] [-Priority <Lowest | Lower | Low |
Normal | High | Higher | Highest | Emergency>] [-Suspend
<SwitchParameter>] [-SuspendComment <String>] [-SuspendWhenReadyToComplete
<SwitchParameter>] [-WhatIf [<SwitchParameter>]] [-WorkloadType <None |
Local | Onboarding | Offboarding | TenantUpgrade | LoadBalancing |
Emergency | RemotePstIngestion | SyncAggregation>]
```

Examples

EXAMPLE 1

This example begins the move request for the public folder \CustomerEngagements from public folder mailbox DeveloperReports to DeveloperReports01.

```
New-PublicFolderMoveRequest -Folders \DeveloperReports
\CustomerEngagements -TargetMailbox DeveloperReports01
```

EXAMPLE 2

This example begins the move request for public folders under the \Dev public folder branch to the target public folder mailbox DeveloperReports01.

Note:

You can also move a branch of public folders by using the **Move-PublicFolderBranch.ps1** script.

```
New-PublicFolderMoveRequest -Folders \Dev
\CustomerEngagements,\Dev\RequestsforChange,\Dev\Usability
-TargetMailbox DeveloperReports01
```

Detailed Description

The **New-PublicFolderMoveRequest** cmdlet moves public folders from a source public folder mailbox to a target public folder mailbox. To move the public folder mailbox to another mailbox database, use the New-MoveRequest cmdlet. To ensure that this folder is already in the target public folder mailbox, run the **Update-PublicFolderMailbox** cmdlet against the target public folder mailbox. You can only perform one move request at a time. You can also move public folders by using the **Move-PublicFolderBranch.ps1** script.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Folders	Required	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter[]	The *Folders* parameter specifies the public folders that you want to move. If the public folder has child public folders, child public folders won't be moved unless you explicitly state them in the command. You can move multiple public folders by separating them with a comma, for example, \Dev\CustomerEngagements, \Dev\RequestsforChange, \Dev\Usability.

TargetMailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *TargetMailbox* parameter specifies the target public folder mailbox that you want to move the public folders to. This parameter accepts the following: • Alias • Canonical DN • Display name • Distinguished name (DN) • *Domain\Account* • GUID • ImmutableId • SMTP address • User principal name (UPN)
AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* is set to 51 or higher. Items are considered corrupted if the item can't be read from the source mailbox or can't be written to the target mailbox. Corrupted items won't be available in the destination mailbox.
AllowLargeItems	Optional	System.Management.	The *AllowLargeItems*

		Automation.SwitchParameter	parameter specifies that you can move large items only when large items are encountered. Large items are email messages with a maximum of 1,023 attachments.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use 0 to not skip bad items. The valid input range for this parameter is from 0 through 2147483647. The default value is 0. We recommend that you keep the default value 0 and only change the *BadItemLimit* parameter value if the request fails. **Note:** If you set the *BadItemLimit* parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the

			command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process is complete.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the request will be kept after it has completed before being automatically removed. The default *CompletedRequestAgeLimit* parameter value is 30 days.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			writes this configuration change to Active Directory.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of large items to skip if the request encounters such items in the public folder. Use 0 to not skip any large items. If any number above 50 is specified, the *AcceptLargeDataLoss* parameter must also be specified. The default value is 0. We recommend that you use the default value of 0 and increase the *LargeItemLimit* only when large items are encountered.
Name	Optional	System.String	The *Name* parameter specifies the name of the public folder move request. If you don't

			specify a name, the default name is `PublicFolderMove`.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	The *Priority* parameter specifies the order in which this request should be processed in the request queue. Requests are processed in order, based on server health, status, priority, and last update time. This parameter accepts the following: • `Lowest` • `Lower` • `Low` • `Normal` • `High` • `Higher` • `Highest` • `Emergency`
Suspend	Optional	System.Management.Automation.SwitchParameter	The *Suspend* switch specifies whether to suspend the request. If you use this switch, the request is queued, but the request won't reach the status of **InProgress** until you resume the request with the relevant resume cmdlet. You don't have to

			specify a value with this switch.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description about why the request was suspended. You can only use this parameter if you specify the *Suspend* parameter.
SuspendWhenReadyToComplete	Optional	System.Management.Automation.SwitchParameter	The *SuspendWhenReadyToComplete* switch specifies whether to suspend the request before it reaches the status of **CompletionInProgress**. After the move is suspended, it has a status of **AutoSuspended**. You can then manually complete the move by using the **Resume-PublicFolderMoveRequest** command.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WorkloadType	Optional	Microsoft.Exchange.MailboxReplicationService.RequestWorkloadType	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-PublicFolderMoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-PublicFolderMoveRequest** cmdlet to cancel a mailbox move initiated using the New-MoveRequest cmdlet. After the move has been finalized, you can't undo the move request.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-PublicFolderMoveRequest -Identity
<PublicFolderMoveRequestIdParameter> <COMMON PARAMETERS>


Remove-PublicFolderMoveRequest -RequestGuid <Guid> -RequestQueue
<DatabaseIdParameter> <COMMON PARAMETERS>
```

```
COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the public folder move request \PublicFolderMove.

```
Remove-PublicFolderMoveRequest -Identity \PublicFolderMove
```

EXAMPLE 2

This example cancels a public folder move by using the *RequestGuid* parameter for a public folder move request on MBXDB01.

Note:

The *RequestGuid* and *RequestQueue* parameters are for debugging purposes only.

```
Remove-PublicFolderMoveRequest -RequestQueue MBXDB01 -
RequestGuid 25e0eaf2-6cc2-4353-b83e-5cb7b72d441f
```

Detailed Description

The *RequestQueue* and *RequestGuid* parameters are for debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.PublicFolderMoveRequestIdParameter	The *Identity* parameter specifies the identity of the public folder move request. The default identity is \PublicFolderMove. You can't use this

			parameter in conjunction with the *RequestGuid* or *RequestQueue* parameter.
RequestGuid	Required	System.Guid	The *RequestGuid* parameter specifies the GUID of the public folder move request. If you specify the *RequestGuid* parameter, you must also specify the *RequestQueue* parameter. You can't use this parameter in conjunction with the *Identity* parameter.
RequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies the target database on which the move request is queued. You can use the following values: • GUID of the database • Database name You can't use this parameter in conjunction with the *Identity* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when

			this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Resume-PublicFolderMoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Resume-PublicFolderMoveRequest** cmdlet to resume a public folder move request that has been suspended or has failed.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Resume-PublicFolderMoveRequest -Identity
<PublicFolderMoveRequestIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resumes the public folder move request \PublicFolderMove.

```
Resume-PublicFolderMoveRequest -Identity \PublicFolderMove
```

EXAMPLE 2

This example resumes failed public folder move requests.

```
Get-PublicFolderMoveRequest -MoveStatus Failed | Resume-
PublicFolderMoveRequest
```

EXAMPLE 3

This example resumes a move request that has the suspend comment "Resume after 10 P.M."

```
Get-PublicFolderMoveRequest -MoveStatus Suspended | Get-
PublicFolderMoveRequestStatistics |Where {$_.Message -like
"*resume after 10 P.M."} | Resume-PublicFolderMoveRequest
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.PublicFolderMoveRequestIdParameter	The *Identity* parameter specifies the identity of the public folder move request. The default identity is \PublicFolderMove.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this

			configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-PublicFolderMoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-PublicFolderMoveRequest** cmdlet to change a public folder move request after the

move request has been created. You can use the **Set-PublicFolderMoveRequest** cmdlet to recover from a failed move request.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-PublicFolderMoveRequest -Identity <PublicFolderMoveRequestIdParameter>
[-AcceptLargeDataLoss <SwitchParameter>] [-BadItemLimit <Unlimited>] [-
CompletedRequestAgeLimit <Unlimited>] [-InternalFlags <InternalMrsFlag[]>]
[-LargeItemLimit <Unlimited>] [-Priority <Lowest | Lower | Low | Normal |
High | Higher | Highest | Emergency>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-SuspendWhenReadyToComplete <$true | $false>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the public folder move request \PublicFolderMove to accept up to five corrupted public folder items.

```
Set-PublicFolderMoveRequest -Identity \PublicFolderMove -
BadItemLimit 5
```

Detailed Description

You can pipeline the **Set-PublicFolderMoveRequest** cmdlet from the **Get-PublicFolderMoveRequestStatistics** or **Get-PublicFolderMoveRequest** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.PublicFolderMoveRequestIdParameter	The *Identity* parameter specifies the identity of the public folder move request. The default identity of a public folder move request is \PublicFolderMove.

AcceptLargeDataLoss	Optional	System.Management.Automation.SwitchParameter	The *AcceptLargeDataLoss* parameter specifies that a large amount of data loss is acceptable if the *BadItemLimit* parameter is set to 51 or higher. Items are considered corrupted if the items can't be read from the source mailbox or can't be written to the target mailbox. Corrupted items won't be available in the destination mailbox.
BadItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *BadItemLimit* parameter specifies the number of bad items to skip if the request encounters corruption in the mailbox. Use `0` to not skip bad items. The valid input range for this parameter is from `0` through `2147483647`. The default value is `0`. We recommend that you keep the default value `0` and only change the *BadItemLimit* parameter value if the request fails.

			Note: If you set the *BadItemLimit* parameter to more than 50, the command fails, and you receive a warning stating: "Please confirm your intention to accept a large amount of data loss by specifying *AcceptLargeDataLoss*." If you receive this warning, you need to run the command again, this time using the *AcceptLargeDataLoss* parameter. No further warnings appear, and any corrupted items aren't available after the process is complete.
CompletedRequestAgeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *CompletedRequestAgeLimit* parameter specifies how long the request is kept after it has completed before being automatically removed. The default *CompletedRequestAgeLimit* parameter value is 30 days.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run.

			To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
InternalFlags	Optional	Microsoft.Exchange.Management.RecipientTasks.InternalMrsFlag[]	The *InternalFlags* parameter specifies the optional steps in the request. This parameter is used primarily for debugging purposes.
LargeItemLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *LargeItemLimit* parameter specifies the number of large items to skip if the request encounters such items in the public folder. Use 0 to not skip any large items. If any number above 50 is specified, the *AcceptLargeDataLoss*

			parameter must also be specified. The default value is 0. We recommend that you use the default value of 0 and increase the *LargeItemLimit* only when large items are encountered.
Priority	Optional	Microsoft.Exchange.MailboxReplicationService.RequestPriority	The *Priority* parameter specifies the order in which this request should be processed in the request queue. Requests are processed in order, based on server health, status, priority, and last update time.
SuspendWhenReadyToComplete	Optional	System.Boolean	The *SuspendWhenReadyToComplete* parameter specifies whether to suspend the request before it reaches the status of **CompletionInProgress**. After the move is suspended, it has a status of **AutoSuspended**. You can then manually

			complete the move by using the **Resume-PublicFolderMoveRequest** command.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Suspend-PublicFolderMoveRequest

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Suspend-PublicFolderMoveRequest** cmdlet to suspend a move request any time after the move request was created, but before it reaches the status of **CompletionInProgress**. You can resume the move request by using the Resume-PublicFolderMoveRequest cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Suspend-PublicFolderMoveRequest -Identity
<PublicFolderMoveRequestIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-SuspendComment <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example suspends the public folder move request \PublicFolderMove.

```
Suspend-PublicFolderMoveRequest -Identity \PublicFolderMove
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.PublicFolderMoveRequestIdParameter	The *Identity* parameter specifies the identity of the public folder move request. The default identity is \PublicFolderMove.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default

			when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
SuspendComment	Optional	System.String	The *SuspendComment* parameter specifies a description about why the request was suspended. You can only use this parameter if you specify the *Suspend* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PublicFolderMoveRequestStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Move and migration cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-PublicFolderMoveRequestStatistics** cmdlet to view detailed information about public folder move requests.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PublicFolderMoveRequestStatistics -Identity
<PublicFolderMoveRequestIdParameter> <COMMON PARAMETERS>


Get-PublicFolderMoveRequestStatistics -RequestQueue <DatabaseIdParameter>
[-RequestGuid <Guid>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Diagnostic <SwitchParameter>] [-DomainController
<Fqdn>] [-IncludeReport <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns the default statistics for the public folder move request \PublicFolderMove.

```
Get-PublicFolderMoveRequestStatistics -Identity
\PublicFolderMove
```

EXAMPLE 2

This example returns the detailed statistics for the move request \PublicFolderMove by pipelining the results to the **Format-List** command.

```
Get-PublicFolderMoveRequestStatistics -Identity
\PublicFolderMove | Format-List
```

Detailed Description

The *RequestQueue* parameter is for debugging purposes only.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MailboxReplicationService.PublicFolderMoveRequestIdParameter	The *Identity* parameter specifies the identity of the public folder move request. The default public folder move request identity is \PublicFolderMove. This parameter can't be used with the *RequestQueue* parameter.
RequestQueue	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *RequestQueue* parameter specifies the mailbox database on

			which the move request resides. You can use one of the following values: • GUID of the database • Database name This parameter is for debugging purposes only. This parameter can't be used with the *Identity* parameter.
Diagnostic	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
IncludeReport	Optional	System.Management.Automation.SwitchParameter	The *IncludeReport* switch specifies whether to return additional details, which can be used for troubleshooting.
RequestGuid	Optional	System.Guid	The *RequestGuid* parameter specifies the GUID of the public folder move request for which you want to view the request statistics.

			This parameter can't be used with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Organization cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-10*

Exchange assistance configuration cmdlets

Get-ExchangeAssistanceConfig

Set-ExchangeAssistanceConfig

Exchange server cmdlets

Get-ExchangeServer

Set-ExchangeServer

Get-ExchangeServerAccessLicenseUser

Get-ExchangeServerAccessLicense

Organization configuration cmdlets

Get-OrganizationConfig

Set-OrganizationConfig

Update Exchange Management Shell help

Update-ExchangeHelp

Get-ExchangeAssistanceConfig

Exchange Management Shell > Exchange 2013 cmdlets > Organization cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ExchangeAssistanceConfig** cmdlet to view the configuration information for the URLs that Microsoft Exchange Help uses to connect to the source of the documentation.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ExchangeAssistanceConfig [-Identity <OrganizationIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example shows the configuration information that the web management interface uses to locate the source of the documentation for Contoso.com.

```
Get-ExchangeAssistanceConfig -Identity Contoso.com
```

EXAMPLE 2

This example shows the configuration information for all organizations and formats the information into a table.

```
Get-ExchangeAssistanceConfig | Format-Table
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Exchange Help settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter specifies the identity of the organization.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ExchangeAssistanceConfig

Exchange Management Shell > Exchange 2013 cmdlets > Organization cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ExchangeAssistanceConfig** cmdlet to modify the Microsoft Exchange Help configurations for your organization.

```
Set-ExchangeAssistanceConfig [-Identity <OrganizationIdParameter>] [-
CommunityLinkDisplayEnabled <$true | $false>] [-CommunityURL <Uri>] [-
Confirm [<SwitchParameter>]] [-ControlPanelFeedbackEnabled <$true |
$false>] [-ControlPanelFeedbackURL <Uri>] [-ControlPanelHelpURL <Uri>] [-
DomainController <Fqdn>] [-ExchangeHelpAppOnline <$true | $false>] [-
ManagementConsoleFeedbackEnabled <$true | $false>] [-
ManagementConsoleFeedbackURL <Uri>] [-ManagementConsoleHelpURL <Uri>] [-
OWAFeedbackEnabled <$true | $false>] [-OWAFeedbackURL <Uri>] [-OWAHelpURL
<Uri>] [-OWALightFeedbackEnabled <$true | $false>] [-OWALightFeedbackURL
<Uri>] [-OWALightHelpURL <Uri>] [-PrivacyLinkDisplayEnabled <$true |
$false>] [-PrivacyStatementURL <Uri>] [-WhatIf [<SwitchParameter>]] [-
WindowsLiveAccountPageURL <Uri>] [-WindowsLiveAccountURLEnabled <$true |
$false>]
```

Examples

EXAMPLE 1

This example changes the Help location for the Exchange Administration Center.

```
Set-ExchangeAssistanceConfig -ExchangeHelpAppOnline $false
-ManagementConsoleHelpURL 'http://exhelponline'
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Help settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
CommunityLinkDisplayEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
CommunityURL	Optional	System.Uri	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ControlPanelFeedbackEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ControlPanelFeedbackURL	Optional	System.Uri	This parameter is reserved for internal Microsoft use.
ControlPanelHelpURL	Optional	System.Uri	The *ControlPanelHelpURL* parameter specifies the URL for where Help is being hosted for the web management interface.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration

			change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
ExchangeHelpAppOnline	Optional	System.Boolean	The *ExchangeHelpAppOnline* specifies whether to indicate that your organization is using the Exchange Help online that's hosted on Microsoft TechNet. The default value is `$true`. If set to `$false`, you need to change the following parameters' URLs to point to the location where your Help is hosted: • *ControlPanelHelpURL* • *ManagementConsoleHelpURL* • *OWAHelpURL* • *OWALightHelpURL*
Identity	Optional	Microsoft.Exchange.Co	This parameter is reserved

		nfiguration.Tasks.OrganizationIdParameter	for internal Microsoft use.
ManagementConsoleFeedbackEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
ManagementConsoleFeedbackURL	Optional	System.Uri	This parameter is reserved for internal Microsoft use.
ManagementConsoleHelpURL	Optional	System.Uri	The *ManagementConsoleHelpURL* parameter specifies the URL for where Help is being hosted.
OWAFeedbackEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
OWAFeedbackURL	Optional	System.Uri	This parameter is reserved for internal Microsoft use.
OWAHelpURL	Optional	System.Uri	The *OWAHelpURL* parameter specifies the URL for where Help is being hosted for the standard version of Microsoft Office Outlook Web App.
OWALightFeedbackEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
OWALightFeedbackURL	Optional	System.Uri	This parameter is reserved for internal Microsoft use.
OWALightHelpURL	Optional	System.Uri	The *OWALightHelpURL* parameter specifies the

			URL for where Help is being hosted for the light version of Outlook Web App.
PrivacyLinkDisplayEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
PrivacyStatementURL	Optional	System.Uri	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsLiveAccountPageURL	Optional	System.Uri	This parameter is reserved for internal Microsoft use.
WindowsLiveAccountURLEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-ExchangeHelp

Exchange Management Shell > Exchange 2013 cmdlets > Organization cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Update-ExchangeHelp** cmdlet to download and integrate the latest version of Help for all cmdlets on the local Microsoft Exchange 2013 server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-ExchangeHelp [-Force <SwitchParameter>]
```

Examples

EXAMPLE 1

To download and integrate the latest version of Help for all Exchange cmdlets on the local Exchange server, run the following command.

```
Update-ExchangeHelp
```

Detailed Description

If you have multiple Exchange servers in your organization, you need to run **Update-ExchangeHelp** on each server. To update Help on a specific Exchange server, connect to the server using remote Shell, and then run **Update-ExchangeHelp** in the remote Shell session. For more information, see Connect to Exchange using remote Shell.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange Help settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description

Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter. By default, the **Update-ExchangeHelp** cmdlet has a throttling period of one hour. If you run this cmdlet within one hour of the last time you ran it, it won't check for updates. You can use the *Force* switch to bypass the throttling restriction and force the cmdlet to check for updates.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ExchangeServer

Exchange Management Shell > Exchange 2013 cmdlets > Organization cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ExchangeServer** cmdlet to obtain the attributes of a specified server. If a server isn't specified, the cmdlet obtains the attributes of all the servers in the Exchange organization.

Note:

When you run the **Get-ExchangeServer** cmdlet with no parameters, it returns the attributes of all the servers in the Exchange organization. To return specific server properties (including domain controller information) where the **Get-ExchangeServer** cmdlet has to contact servers directly or perform a complex or slow calculation, make sure you use the *Status* parameter.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ExchangeServer -Domain <Fqdn> <COMMON PARAMETERS>

Get-ExchangeServer [-Identity <ServerIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-Status <SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves the attributes of all the servers in the Exchange organization.

```
Get-ExchangeServer | Format-List
```

Detailed Description

To view all the Exchange server attributes that this cmdlet returns, you must pipe the command to the **Format-List** cmdlet.

The **ExchangeVersion** attribute returned is the minimum version of Microsoft Exchange that you can use to manage the returned object. This attribute isn't the same as the version of Exchange displayed in the Exchange Administration Center when you select **Server Configuration**.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Domain	Required	Microsoft.Exchange.Data.Fqdn	The *Domain* parameter specifies the fully qualified domain name (FQDN) of the domain. If you use this parameter, you can't use the *Identity* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance

			of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the identity of the server. If you use this parameter, you can't use the *Domain* parameter.
Status	Optional	System.Management.Automation.SwitchParameter	The *Status* parameter specifies the status of the server.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ExchangeServer

Exchange Management Shell > Exchange 2013 cmdlets > Organization cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ExchangeServer** cmdlet to set Exchange attributes in Active Directory for a specified server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ExchangeServer -Identity <ServerIdParameter> [-Confirm
[<SwitchParameter>]] [-CustomerFeedbackEnabled <$true | $false>] [-
DomainController <Fqdn>] [-ErrorReportingEnabled <$true | $false>] [-
InternetWebProxy <Uri>] [-MailboxProvisioningAttributes
<MailboxProvisioningAttributes>] [-MailboxRelease <None | E14 | E15>] [-
MonitoringGroup <String>] [-ProductKey <ProductKey>] [-
StaticConfigDomainController <String>] [-StaticDomainControllers
<MultiValuedProperty>] [-StaticExcludedDomainControllers
<MultiValuedProperty>] [-StaticGlobalCatalogs <MultiValuedProperty>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables error reporting on the specified server.

```
Set-ExchangeServer -Identity TestServer.Contoso.com -
ErrorReportingEnabled: $false
```

EXAMPLE 2

This example enrolls an Exchange server into the Customer Experience Improvement Program. In this example, the server name is SERVER01.

```
Set-ExchangeServer -Identity 'SERVER01' -
CustomerFeedbackEnabled $true
```

EXAMPLE 3

This example removes an Exchange server from the Customer Experience Improvement Program. In this example, the server name is SERVER01.

```
Set-ExchangeServer -Identity 'SERVER01' -
CustomerFeedbackEnabled $false
```

Detailed Description

The **Set-ExchangeServer** cmdlet sets generic Exchange attributes in Active Directory for a specified computer. You can only use this task on one server at a time. If you want to bulk manage your servers running Microsoft Exchange, add this task to a script.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the GUID, distinguished name (DN), or name of the server.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CustomerFeedbackEnabled	Optional	System.Boolean	The *CustomerFeedbackEnabled* parameter specifies whether the Exchange server is enrolled in the Microsoft Customer Experience Improvement Program (CEIP). The CEIP collects anonymous information about how you use Exchange and

			problems that you might encounter. If you decide not to participate in the CEIP, the servers are opted-out automatically.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
ErrorReportingEnabled	Optional	System.Boolean	The *ErrorReportingEnabled* parameter specifies whether error reporting is enabled.
InternetWebProxy	Optional	System.Uri	The *InternetWebProxy* parameter specifies

			which web proxy servers, such as computers running Forefront Threat Management Gateway, Exchange should use to reach the Internet.
MailboxProvisioningAttributes	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MailboxProvisioningAttributes	This parameter is reserved for internal Microsoft use.
MailboxRelease	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MailboxRelease	This parameter is reserved for internal Microsoft use.
MonitoringGroup	Optional	System.String	The *MonitoringGroup* parameter specifies how to add your Exchange 2013 servers to monitoring groups. You can add your servers to an existing group or create a monitoring group based on location or deployment, or to partition monitoring responsibility among your servers.
ProductKey	Optional	Microsoft.Exchange.Management.SystemConfi	The *ProductKey* parameter specifies the

		gurationTasks.ProductKey	server product key.
StaticConfigDomainController	Optional	System.String	The *StaticConfigDomainController* parameter specifies whether to configure a domain controller to be used by the server via Directory Service Access (DSAccess).
StaticDomainControllers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *StaticDomainControllers* parameter specifies whether to configure a list of domain controllers to be used by the server via DSAccess.
StaticExcludedDomainControllers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *StaticExcludedDomainControllers* parameter specifies whether to exclude a list of domain controllers from being used by the server.
StaticGlobalCatalogs	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *StaticGlobalCatalogs* parameter specifies whether to configure a list of global catalogs to

			be used by the server via DSAccess.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ExchangeServerAccessLicense

Exchange Management Shell > Exchange 2013 cmdlets > Organization cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ExchangeServerAccessLicense** cmdlet to return a list of licenses in use in your

Microsoft Exchange Server 2013 organization. This refers to the specific legal name of the license, as defined in the Microsoft Product List and is representative of your licenses when you run this cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ExchangeServerAccessLicense
```

Examples

EXAMPLE 1

This example retrieves a list of Exchange 2013 licenses in your organization.

```
Get-ExchangeServerAccessLicense
```

Detailed Description

The **Get-ExchangeServerAccessLicense** cmdlet returns a list of licenses in use in your Exchange 2013 organization. The cmdlet returns a collection of these license names:

- Exchange 15 Standard CAL
- Exchange 15 Enterprise CAL
- Exchange 15 Server Standard Edition
- Exchange 15 Enterprise Edition

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell Infrastructure Permissions" section in the Exchange and Shell infrastructure permissions topic.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ExchangeServerAccessLicenseUser

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ExchangeServerAccessLicenseUser** cmdlet to return a list of unique users for the specified license name. Each object contains the fully qualified domain name (FQDN) or primary SMTP address of the mailbox and the license name to which it's associated.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ExchangeServerAccessLicenseUser -LicenseName <String>
```

Examples

EXAMPLE 1

This example returns the unique users for your license name, Exchange 2013 Server Standard Edition.

```
Get-ExchangeServerAccessLicenseUser -LicenseName "Exchange Server 2013 Standard Edition"
```

Detailed Description

The **Get-ExchangeServerAccessLicenseUser** cmdlet returns a collection of unique users for the specified license name. The list of unique users represents an estimate of your licenses when you run this cmdlet. Each object contains the FQDN or primary SMTP address of the mailbox and the license name to which it's associated.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
LicenseName	Required	System.String	The *LicenseName* parameter specifies a license type for the

			servers in your organization. License types include the following: • Exchange 2013 Standard client access license (CAL) • Exchange 2013 Enterprise CAL • Exchange 2013 Server Standard Edition • Exchange 2013 Server Enterprise Edition

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-OrganizationConfig

Exchange Management Shell > Exchange 2013 cmdlets > Organization cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-OrganizationConfig** cmdlet to get configuration data for an Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-OrganizationConfig [-Identity <OrganizationIdParameter>] <COMMON
```

```
PARAMETERS>

Get-OrganizationConfig [-AccountPartition <AccountPartitionIdParameter>]
<COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example gets the organization configuration information for the domain controller ContosoDC.

```
Get-OrganizationConfig -DomainController ContosoDC
```

EXAMPLE 2

This example gets the configuration information for the tenant organization.

```
Get-OrganizationConfig Export-Cli c:\myFile.xml
```

For more information, see **Troubleshoot a hybrid deployment**.

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter specifies the organization you want configuration data from.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-OrganizationConfig

Exchange Management Shell > Exchange 2013 cmdlets > Organization cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-24*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-OrganizationConfig** cmdlet to configure various settings of an Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-OrganizationConfig [-Identity <OrganizationIdParameter>] <COMMON
PARAMETERS>


Set-OrganizationConfig [-AdfsAuthenticationConfiguration <String>] <COMMON
PARAMETERS>


Set-OrganizationConfig [-AdfsAudienceUris <MultiValuedProperty>] [-
AdfsEncryptCertificateThumbprint <String>] [-AdfsIssuer <Uri>] [-
AdfsSignCertificateThumbprints <MultiValuedProperty>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
ActivityBasedAuthenticationTimeoutEnabled <$true | $false>] [-
ActivityBasedAuthenticationTimeoutInterval <EnhancedTimeSpan>] [-
ActivityBasedAuthenticationTimeoutWithSingleSignOnEnabled <$true |
$false>] [-AppsForOfficeEnabled <$true | $false>] [-
AVAuthenticationService <ProtocolConnectionSettings>] [-
ByteEncoderTypeFor7BitCharsets <Int32>] [-CalendarVersionStoreEnabled
<$true | $false>] [-Confirm [<SwitchParameter>]] [-CustomerFeedbackEnabled
<$true | $false>] [-DefaultPublicFolderAgeLimit <EnhancedTimeSpan>] [-
DefaultPublicFolderDeletedItemRetention <EnhancedTimeSpan>] [-
DefaultPublicFolderIssueWarningQuota <Unlimited>] [-
DefaultPublicFolderMaxItemSize <Unlimited>] [-
DefaultPublicFolderMovedItemRetention <EnhancedTimeSpan>] [-
DefaultPublicFolderProhibitPostQuota <Unlimited>] [-
DistributionGroupDefaultOU <OrganizationalUnitIdParameter>] [-
DistributionGroupNameBlockedWordsList <MultiValuedProperty>] [-
DistributionGroupNamingPolicy <DistributionGroupNamingPolicy>] [-
DomainController <Fqdn>] [-EwsAllowEntourage <$true | $false>] [-
EwsAllowList <MultiValuedProperty>] [-EwsAllowMacOutlook <$true | $false>]
[-EwsAllowOutlook <$true | $false>] [-EwsApplicationAccessPolicy
<EnforceAllowList | EnforceBlockList>] [-EwsBlockList
<MultiValuedProperty>] [-EwsEnabled <$true | $false>] [-
ExchangeNotificationEnabled <$true | $false>] [-
ExchangeNotificationRecipients <MultiValuedProperty>] [-
ForwardSyncLiveIdBusinessInstance <$true | $false>] [-
HierarchicalAddressBookRoot <UserContactGroupIdParameter>] [-Industry
<NotSpecified | Agriculture | Finance | BusinessServicesConsulting |
Communications | ComputerRelatedProductsServices | Construction |
Education | EngineeringArchitecture | Government | Healthcare |
Hospitality | Legal | Manufacturing | MediaMarketingAdvertising | Mining |
NonProfit | PersonalServices | PrintingPublishing | RealEstate | Retail |
Transportation | Utilities | Wholesale | Other>] [-
IsExcludedFromOffboardMigration <$true | $false>] [-
IsExcludedFromOnboardMigration <$true | $false>] [-
IsFfoMigrationInProgress <$true | $false>] [-
IsGuidPrefixedLegacyDnDisabled <$true | $false>] [-
IsMailboxForcedReplicationDisabled <$true | $false>] [-
IsProcessEhaMigratedMessagesEnabled <$true | $false>] [-
IsSyncPropertySetUpgradeAllowed <$true | $false>] [-MailTipsAllTipsEnabled
<$true | $false>] [-MailTipsExternalRecipientsTipsEnabled <$true |
$false>] [-MailTipsGroupMetricsEnabled <$true | $false>] [-
MailTipsLargeAudienceThreshold <UInt32>] [-
MailTipsMailboxSourcedTipsEnabled <$true | $false>] [-
ManagedFolderHomepage <String>] [-MapiHttpEnabled <$true | $false>] [-
MaxConcurrentMigrations <Unlimited>] [-
MicrosoftExchangeRecipientEmailAddresses <ProxyAddressCollection>] [-
MicrosoftExchangeRecipientEmailAddressPolicyEnabled <$true | $false>] [-
MicrosoftExchangeRecipientPrimarySmtpAddress <SmtpAddress>] [-
MicrosoftExchangeRecipientReplyRecipient <RecipientIdParameter>] [-
OrganizationSummary <MultiValuedProperty>] [-
PreferredInternetCodePageForShiftJis <Int32>] [-
PublicComputersDetectionEnabled <$true | $false>] [-
PublicFolderMigrationComplete <$true | $false>] [-PublicFoldersEnabled
<None | Local | Remote>] [-PublicFoldersLockedForMigration <$true |
```

```
$false>] [-ReadTrackingEnabled <$true | $false>] [-ReleaseTrack <None |
FirstRelease | Preview>] [-RemotePublicFolderMailboxes
<MultiValuedProperty>] [-RequiredCharsetCoverage <Int32>] [-
RmsoSubscriptionStatus <Unknown | Enabled | Suspended | LockedOut |
AdhocEnabled | Deleted>] [-SCLJunkThreshold <Int32>] [-SharePointUrl
<Uri>] [-SIPAccessService <ProtocolConnectionSettings>] [-
SIPSessionBorderController <ProtocolConnectionSettings>] [-
SiteMailboxCreationURL <Uri>] [-TenantRelocationsAllowed <$true | $false>]
[-UMAvailableLanguages <MultiValuedProperty>] [-WACDiscoveryEndpoint
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a distribution group naming policy using the following configuration:

- Distribution groups will be created in the Users\Groups container.
- The words curse, bad, and offensive will be blocked from being used in distribution group names.
- All distribution groups will be prefixed with "DL_" and suffixed with an underscore (_) and the user's department and country code.

```
Set-OrganizationConfig -DistributionGroupDefaultOU Users
\Groups -DistributionGroupNameBlockedWordsList
curse,bad,offensive -DistributionGroupNamingPolicy
"DL_<GroupName>_<Department><CountryCode>"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange server configuration settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
ActivityBasedAuthenticationTimeoutEnabled	Optional	System.Boolean	The *ActivityBasedAuthenticationTimeoutEnabled*

			parameter specifies whether the timed logoff feature is enabled. The default value is `$true`.
ActivityBasedAuthenticationTimeoutInterval	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *ActivityBasedAuthenticationTimeoutInterval* parameter specifies the time span for logoff. You enter this value as a time span: *hh*:*mm*:*ss* where *hh* = hours, *mm* = minutes and *ss* = seconds. Valid values for this parameter are from `00:05:00` to `08:00:00` (5 minutes to 8 hours). The default value is `06:00:00` (6 hours).
ActivityBasedAuthenticationTimeoutWithSingleSignOnEnabled	Optional	System.Boolean	The *ActivityBasedAuthenticationTimeoutWithSingleSignOnEnabled* parameter specifies whether to keep single sign-on enabled. The default value is `$true`.
AdfsAudienceUris	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *AdfsAudienceUris* parameter specifies one

			or more external URLs that are used for Active Directory Federation Services (AD FS) claims-based authentication. For example, the external Outlook Web App and external Exchange admin center (EAC) URLs. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`. For more information about configuring AD FS claims based authentication in

			Exchange, see Using AD FS claims-based authentication with Outlook Web App and EAC.
AdfsAuthenticationConfiguration	Optional	System.String	This parameter is reserved for internal Microsoft use.
AdfsEncryptCertificateThumbprint	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *AdfsEncryptCertificateThumbprint* parameter specifies the service communication SSL certificate that's used for AD FS claims-based authentication. This parameter uses a certificate thumbprint (GUID) value to identify the certificate. To get the thumbprint value for the service communication SSL certificate, open Windows PowerShell on the AD FS server and run the command `Get-ADFSCertificate -CertificateType "Service-Communications"`. For

			more information, see Get-ADFSCertificate.
AdfsIssuer	Optional	System.Uri	This parameter is available only in on-premises Exchange 2013. The *AdfsIssuer* parameter specifies URL of the AD FS server that's used for AD FS claims-based authentication. This is the URL where AD FS relying parties send users for authentication. To get this value, open Windows PowerShell on the AD FS server and run the command `Get-ADFSEndpoint -AddressPath /adfs/ls | Format-List FullUrl`.
AdfsSignCertificateThumbprints	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *AdfsSignCertificateThumbprints* parameter specifies one or more X.509 token-signing certificates that are used for AD FS claims-based authentication. This parameter uses certificate thumbprint values

			(GUIDs) to identify the certificates. To get the thumbprint values of the primary and secondary token-signing certificates, open Windows PowerShell on the AD FS server and run the command `Get-ADFSCertificate -CertificateType "Token-signing"`. For more information, see Get-ADFSCertificate. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.

AppsForOfficeEnabled	Optional	System.Boolean	The *AppsForOfficeEnabled* parameter specifies whether to enable apps for Microsoft Outlook features. By default, the parameter is set to `$true`. If the flag is set to `$false`, no new apps can be activated for any user in the organization.
AVAuthenticationService	Optional	Microsoft.Exchange.Data.ProtocolConnectionSettings	This parameter is reserved for internal Microsoft use.
ByteEncoderTypeFor7BitCharsets	Optional	System.Int32	The *ByteEncoderTypeFor7BitCharsets* parameter specifies the 7-bit transfer encoding method for MIME format for messages sent to this remote domain. The valid values for this parameter are: • 0 Always use default 7-bit transfer encoding for HTML and plain text. • 1 Always use QP (quoted-printable) encoding for HTML and plain text. • 2 Always use Base64 encoding for HTML and plain text. • 5 Use QP encoding for HTML and plain text

			unless line wrapping is enabled in plain text. If line wrapping is enabled, use 7-bit encoding for plain text. • 6 Use Base64 encoding for HTML and plain text, unless line wrapping is enabled in plain text. If line wrapping is enabled in plain text, use Base64 encoding for HTML, and use 7-bit encoding for plain text. • 13 Always use QP encoding for HTML. Always use 7-bit encoding for plain text. • 14 Always use Base64 encoding for HTML. Always use 7-bit encoding for plain text. If no value is specified, Exchange always uses QP encoding for HTML and plain text.
CalendarVersionStoreEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
CustomerFeedbackEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *CustomerFeedbackEnabled* parameter specifies whether the server running Microsoft Exchange is enrolled in the Microsoft Customer Experience Improvement Program.
DefaultPublicFolderAgeLimit	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *DefaultPublicFolderAgeLimit* parameter specifies the default age limit for public folders across the entire organization. A public folder is automatically deleted when this age limit is exceeded. This attribute applies to all public folders in the organization that don't have their own **AgeLimit** setting.
DefaultPublicFolderDeletedItemRetention	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *DefaultPublicFolderDeletedItemRetention* parameter specifies the default value

			of the length of time to retain deleted items for public folders across the entire organization. This attribute applies to all public folders in the organization that don't have their own **RetainDeletedItemsFor** attribute set.
DefaultPublicFolderIssueWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *DefaultPublicFolderIssueWarningQuota* parameter specifies the default value across the entire organization for the public folder size at which a warning message is sent to this folder's owners, warning that the public folder is almost full. This attribute applies to all public folders within the organization that don't have their own warning quota attribute set. The default value of this attribute is `unlimited`. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes)

			• `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 0 through 2199023254529 bytes(2 TB). If you enter a value of `unlimited`, no size limit is imposed on the public folder.
DefaultPublicFolderMaxItemSize	Optional	Microsoft.Exchange.Data.Unlimited	The *DefaultPublicFolderMaxItemSize* parameter specifies the default maximum size for posted items within public folders across the entire organization. Items larger than the value of the *DefaultPublicFolderMaxItemSize* parameter are rejected. This attribute applies to all public folders within the organization that don't have their own **MaxItemSize** attribute set. The default value of this attribute is `unlimited`. When you enter a value, qualify the value with one

			of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from 0 through 2199023254529 bytes (2 TB). If you enter a value of `unlimited`, no size limit is imposed on the public folder.
DefaultPublicFolderMovedItemRetention	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *DefaultPublicFolderMovedItemRetention* parameter specifies how long items that have been moved between mailboxes are kept in the source mailbox for recovery purposes before being removed by the Public Folder Assistant. When you move folder contents between mailboxes, a copy of the original data is left on the source mailbox, inaccessible for users but available for recovery by

			system administrators. If the move process fails and you want to roll it back, use the `Set-PublicFolder -OverrideContentMailbox` command to recover data. For more information, see Set-PublicFolder.
DefaultPublicFolderProhibitPostQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *DefaultPublicFolderProhibitPostQuota* parameter specifies the size of a public folder at which users are notified that the public folder is full. Users can't post to a folder whose size is larger than the *DefaultPublicFolderProhibitPostQuota* parameter value. The default value of this attribute is `unlimited`. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The valid input range for

			this parameter is from 0 through 2199023254529 bytes (2 TB). If you enter a value of `unlimited`, no size limit is imposed on the public folder.
DistributionGroupDefaultOU	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *DistributionGroupDefaultOU* parameter specifies the container where distribution groups are created by default.
DistributionGroupNameBlockedWordsList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *DistributionGroupNameBlockedWordsList* parameter specifies words that can't be included in the names of distribution groups. Separate multiple values with commas.
DistributionGroupNamingPolicy	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DistributionGroupNamingPolicy	The *DistributionGroupNamingPolicy* parameter specifies the template applied to the name of distribution groups that are created in the organization. You can enforce that a prefix or suffix be applied to all distribution groups. Prefixes and suffixes can be either a string or an

			attribute, and you can combine strings and attributes. When creating a naming policy, use the following syntax: "prefix<*GroupName*>suffix" **Note:** Don't set the <*GroupName*>. Users create the name when they create the distribution group. You can have multiple prefixes and suffixes. You can use the following attributes that will be gathered from the user who's creating the distribution group mailbox settings: • Department • Company • Office • StateOrProvince • CountryorRegion • CountryCode • Title • CustomAttribute1 to CustomAttribute15 To create a naming policy using an attribute, use the following syntax: `"<PrefixAttribute><GroupName><SuffixAttribut`

			e>". For example, to create a naming policy using the Department as a prefix and CustomAttribute1 as the suffix: "<Department><GroupName><CustomAttribute1>". To create a naming policy using strings, use the following syntax "string<GroupName>string". For example to create a naming policy using the string "DL_" as the prefix use the following syntax: "DL_<GroupName>".
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge

			Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
EwsAllowEntourage	Optional	System.Boolean	The *EwsAllowEntourage* parameter specifies whether to enable or disable Entourage 2008 to access Exchange Web Services (EWS) for the entire organization. The default value is `$true`.
EwsAllowList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *EwsAllowList* parameter specifies the applications (user agent strings) that can access EWS when the *EwsApplicationAccessPolicy* parameter is set to `EnforceAllowList`.
EwsAllowMacOutlook	Optional	System.Boolean	The *EwsAllowMacOutlook* parameter specifies whether to enable or disable Microsoft Outlook for Mac 2011 to access EWS for the entire organization.
EwsAllowOutlook	Optional	System.Boolean	The *EwsAllowOutlook* parameter enables or

			disables Microsoft Office Outlook 2007 to access EWS for the entire organization. Outlook 2007 uses EWS for free and busy information, out-of-office settings, and calendar sharing.
EwsApplicationAccessPolicy	Optional	Microsoft.Exchange.Data.Directory.EwsApplicationAccessPolicy	The *EwsApplicationAccessPolicy* parameter defines which applications other than Entourage, Mac Outlook, and Outlook can access EWS. If set to `EnforceAllowList`, only applications specified in the *EwsAllowList* parameter are allowed access to EWS. If set to `EnforceBlockList`, every application is allowed access to EWS except the ones specified in the *EwsBlockList* parameter.
EwsBlockList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *EwsBlockList* parameter specifies the applications that can't access EWS when the *EwsApplicationAccessPolicy* parameter is set to `EnforceBlockList`.

EwsEnabled	Optional	System.Boolean	The *EwsEnabled* parameter specifies whether to globally enable or disable EWS access for the entire organization, regardless of what application is making the request. When the *EwsEnabled* parameter is set to `$false`, EWS access is turned off, regardless of the values of the *EwsAllowEntourage*, *EwsAllowMacOutlook*, and *EwsAllowOutlook* parameters. For the *EwsAllowEntourage*, *EwsAllowMacOutlook*, *EwsAllowOutlook* parameters to be meaningful, the *EwsEnabled* parameter must be set to `$true`.
ExchangeNotificationEnabled	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *ExchangeNotificationEnabled* parameter enables or disables Exchange notifications sent to

			administrators regarding their organizations. Valid input for this parameter is `$true` or `$false`.
ExchangeNotificationRecipients	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in the cloud-based service. The *ExchangeNotificationRecipients* parameter specifies the recipients for Exchange notifications sent to administrators regarding their organizations. If the *ExchangeNotificationEnabled* parameter is set to `$false`, no notification messages are sent. Be sure to enclose values that contain spaces in quotation marks (") and separate multiple values with commas. If this parameter isn't set, Exchange notifications are sent to all administrators.
ForwardSyncLiveIdBusinessInstance	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
HierarchicalAddressBookRoot	Optional	Microsoft.Exchange.Configuration.Tasks.User	The *HierarchicalAddressBookR*

		ContactGroupIdParameter	*oot* parameter specifies the user, contact, or group to be used as the root organization for a hierarchical address book in the Exchange organization. Setting a value for this parameter enables the hierarchical address book to be automatically displayed in Outlook for the organization. The default value is `$null`. This parameter accepts the following values: • Distinguished name (DN) • Canonical name • GUID • Name • Display name • LegacyExchangeDN • Primary SMTP email address
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter specifies the identity of the tenant organization.
Industry	Optional	Microsoft.Exchange.Data.Directory.IndustryType	This parameter is available only in on-premises Exchange 2013. The *Industry* parameter

			specifies the industry that best represents your organization.
IsExcludedFromOffboardMigration	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *IsExcludedFromOffboardMigration* parameter specifies that no new moves from the cloud to your on-premises organization are permitted. When this flag is set, no offboarding move requests are allowed.
IsExcludedFromOnboardMigration	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *IsExcludedFromOnboardMigration* parameter specifies that no new moves from your on-premises organization to the cloud are permitted. When this flag is set, no onboarding move requests are allowed.
IsFfoMigrationInProgr	Optional	System.Boolean	This parameter is reserved

ess			for internal Microsoft use.
IsGuidPrefixedLegacyDnDisabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
IsMailboxForcedReplicationDisabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
IsProcessEhaMigratedMessagesEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
IsSyncPropertySetUpgradeAllowed	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
MailTipsAllTipsEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *MailTipsAllTipsEnabled* parameter specifies whether MailTips are enabled. The default value is `$true`.
MailTipsExternalRecipientsTipsEnabled	Optional	System.Boolean	The *MailTipsExternalRecipientsTipsEnabled* parameter specifies whether MailTips for external recipients are enabled. The default value is `$false`.
MailTipsGroupMetricsEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The

			MailTipsGroupMetricsEnabled parameter specifies whether MailTips that rely on group metrics data are enabled. The default value is `$true`.
MailTipsLargeAudienceThreshold	Optional	System.UInt32	The *MailTipsLargeAudienceThreshold* parameter specifies what a large audience is. The default value is `25`.
MailTipsMailboxSourcedTipsEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *MailTipsMailboxSourcedTipsEnabled* parameter specifies whether MailTips that rely on mailbox data (out-of-office or full mailbox) are enabled.
ManagedFolderHomepage	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *ManagedFolderHomepage* parameter specifies the URL of the web page that's displayed when users click the **Managed**

			Folders folder in Microsoft Outlook. If a URL isn't specified, Outlook doesn't display a managed folders home page.
MapiHttpEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *MapiHttpEnabled* parameter enables or disables Microsoft Outlook connections to mailboxes by using the MAPIHTTP protocol. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
MaxConcurrentMigrations	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxConcurrentMigrations* parameter specifies the maximum number of concurrent migrations that your organization can configure at any specific time.
MicrosoftExchangeRecipientEmailAddresses	Optional	Microsoft.Exchange.Data.ProxyAddressCollec	This parameter is available only in on-

		tion	premises Exchange 2013. The *MicrosoftExchangeRecipientEmailAddresses* parameter specifies one or more email addresses for the recipient. All valid Microsoft Exchange email address types may be used. You can specify multiple values for this parameter as a comma-delimited list. If the *MicrosoftExchangeRecipientEmailAddressPolicyEnabled* parameter is set to `$true`, the email addresses are automatically generated by the default email address policy. This means you can't use the *MicrosoftExchangeRecipientEmailAddresses* parameter. Email addresses that you specify by using the *MicrosoftExchangeRecipientEmailAddresses* parameter replace any existing email addresses already configured.

MicrosoftExchangeRecipientEmailAddressPolicyEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *MicrosoftExchangeRecipientEmailAddressPolicyEnabled* parameter specifies whether the default email address policy is automatically applied to the Exchange recipient. The default value is `$true`. If this parameter is set to `$true`, Exchange automatically adds new email addresses to the Exchange recipient when email address policies are added or modified in the Exchange organization. If this parameter is set to `$false`, you must manually add new email addresses to the Exchange recipient when email address policies are added or modified. If you change the value of the *MicrosoftExchangeRecipientEmailAddressPolicyEnabled* parameter from

			`$false` to `$true`, any email addresses that you defined by using the *MicrosoftExchangeRecipientEmailAddresses* parameter are preserved. However, the value of the *MicrosoftExchangeRecipientPrimarySmtpAddress* parameter reverts to MicrosoftExchange329e71ec88ae4615bbc36ab6ce41109e@<*Accepted Domain in Highest Priority Email Address Policy*>.
MicrosoftExchangeRecipientPrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in on-premises Exchange 2013. The *MicrosoftExchangeRecipientPrimarySmtpAddress* parameter specifies the primary return SMTP email address for the Exchange recipient. If the *MicrosoftExchangeRecipientEmailAddressPolicyEnabled* parameter is set to `$true`, you can't use the *MicrosoftExchangeRecipientPrimarySmtpAddress*

			parameter. If you modify the value of the *MicrosoftExchangeRecipientPrimarySmtpAddress* parameter, the value is automatically added to the list of email addresses defined in the *MicrosoftExchangeRecipientEmailAddresses* parameter. The *MicrosoftExchangeRecipientPrimarySmtpAddress* parameter is meaningful only if the Exchange recipient has more than one defined SMTP email address. If the *MicrosoftExchangeRecipientEmailAddresses* parameter has only one defined SMTP email address, the value of the *MicrosoftExchangeRecipientPrimarySmtpAddress* parameter and the *MicrosoftExchangeRecipientEmailAddresses* parameter are the same.
MicrosoftExchangeRec	Optional	Microsoft.Exchange.Co	This parameter is

ipientReplyRecipient		nfiguration.Tasks.RecipientIdParameter	available only in on-premises Exchange 2013. The *MicrosoftExchangeRecipientReplyRecipient* parameter specifies the recipient that should receive messages sent to the Exchange recipient. Typically, you would configure a mailbox to receive the messages sent to the Exchange recipient. You can use any of the following values for the specified recipient: • DN • Canonical name • GUID • Name • Display name • Alias • Exchange DN • Primary SMTP email address If you don't configure a recipient for the Exchange recipient, messages sent to the Exchange recipient are discarded.
OrganizationSummary	Optional	Microsoft.Exchange.Data.MultiValuedPropert	This parameter is available only in on-

		y	premises Exchange 2013. The *OrganizationSummary* parameter specifies a summarized description that best represents your organization.
PreferredInternetCodePageForShiftJis	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
PublicComputersDetectionEnabled	Optional	System.Boolean	The *PublicComputersDetectionEnabled* parameter specifies whether Exchange Online will detect when a user signs in to Outlook Web App from a public or private computer or network, and then enforces the attachment handling settings from public networks. The default is `$false`. However, if you set this parameter to `$true`, Exchange Online will determine if the user is signing in to Outlook Web App from a public computer, and all public attachment handling rules will be applied and enforced.

PublicFolderMigrationComplete	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *PublicFolderMigrationComplete* parameter is used during public folder migration. When you set the *PublicFolderMigrationComplete* parameter to `$true`, transport starts rerouting the queued messages to a new destination. The default value is `$false`.
PublicFoldersEnabled	Optional	Microsoft.Exchange.Data.Directory.PublicFoldersDeployment	The *PublicFoldersEnabled* parameter specifies how public folders are deployed in your organization. This parameter uses one of the following values. • `Local` The public folders are deployed locally in your organization. • `Remote` The public folders are deployed in the remote forest. • `None` No public folders are deployed for this organization.
PublicFoldersLockedF	Optional	System.Boolean	This parameter is available only in on-

orMigration			premises Exchange 2013. The *PublicFoldersLockedForMigration* parameter specifies whether users are locked out from accessing down level public folder servers. When you set the *PublicFoldersLockedForMigration* parameter to `$true`, users are locked out from accessing down level public folder servers. This is used for public folder migration during final stages. The default value is `$false`, which means that the user is able to access public folder servers.
ReadTrackingEnabled	Optional	System.Boolean	The *ReadTrackingEnabled* parameter specifies whether the tracking for read status for messages in an organization is enabled. The default value is `$false`.
ReleaseTrack	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ReleaseTrack	This parameter is reserved for internal Microsoft use.

RemotePublicFolderMailboxes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RemotePublicFolderMailboxes* parameter specifies the identities of the public folder objects (represented as mail user objects locally) corresponding to the public folder mailboxes created in the remote forest. The public folder values set here are used only if the public folder deployment is a remote deployment.
RequiredCharsetCoverage	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
RmsoSubscriptionStatus	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.RmsoSubscriptionStatusFlags	This parameter is reserved for internal Microsoft use.
SCLJunkThreshold	Optional	System.Int32	This parameter is available only in on-premises Exchange 2013. The *SCLJunkThreshold* parameter specifies the spam confidence level (SCL) threshold. Messages with an SCL greater than the value that you specify for the *SCLJunkThreshold*

			parameter are moved to the Junk Email folder. Valid values are integers from 0 through 9, inclusive.
SharePointUrl	Optional	System.Uri	This parameter is reserved for internal Microsoft use.
SIPAccessService	Optional	Microsoft.Exchange.Data.ProtocolConnectionSettings	This parameter is reserved for internal Microsoft use.
SIPSessionBorderController	Optional	Microsoft.Exchange.Data.ProtocolConnectionSettings	This parameter is reserved for internal Microsoft use.
SiteMailboxCreationURL	Optional	System.Uri	The *SiteMailboxCreationURL* parameter specifies whether to create a site from a Microsoft Exchange Server 2013 mailbox. **Note:** The mailbox must be running Microsoft Outlook 2013.
TenantRelocationsAllowed	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
UMAvailableLanguages	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The

			UMAvailableLanguages parameter will be removed in future versions of the product.
WACDiscoveryEndpoint	Optional	System.String	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Permissions cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-09*

Active Directory permissions cmdlets

Add-ADPermission

Get-ADPermission

Remove-ADPermission

Management role cmdlets

Get-ManagementRole

New-ManagementRole

Remove-ManagementRole

Role assignment cmdlets

Get-ManagementRoleAssignment

New-ManagementRoleAssignment

Remove-ManagementRoleAssignment

Set-ManagementRoleAssignment

Role assignment policy cmdlets

Get-RoleAssignmentPolicy

New-RoleAssignmentPolicy

Remove-RoleAssignmentPolicy

Set-RoleAssignmentPolicy

Role entry cmdlets

Get-ManagementRoleEntry

Add-ManagementRoleEntry

Remove-ManagementRoleEntry

Set-ManagementRoleEntry

Role group cmdlets

Get-RoleGroup

New-RoleGroup

Remove-RoleGroup

Set-RoleGroup

Add-RoleGroupMember

Get-RoleGroupMember

Remove-RoleGroupMember

Update-RoleGroupMember

Role scope cmdlets

Get-ManagementScope

New-ManagementScope

Remove-ManagementScope

Set-ManagementScope

Add-ADPermission

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Add-ADPermission** cmdlet to add permissions to an Active Directory object.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-ADPermission -Identity <ADRawEntryIdParameter> -Owner
<SecurityPrincipalIdParameter> <COMMON PARAMETERS>


Add-ADPermission -Identity <ADRawEntryIdParameter> -User
<SecurityPrincipalIdParameter> [-AccessRights <ActiveDirectoryRights[]>]
[-ChildObjectTypes <ADSchemaObjectIdParameter[]>] [-Deny
<SwitchParameter>] [-ExtendedRights <ExtendedRightIdParameter[]>] [-
InheritanceType <None | All | Descendents | SelfAndChildren | Children>]
[-InheritedObjectType <ADSchemaObjectIdParameter>] [-Properties
<ADSchemaObjectIdParameter[]>] <COMMON PARAMETERS>
```

```
Add-ADPermission -Instance <ADAcePresentationObject> [-AccessRights
<ActiveDirectoryRights[]>] [-ChildObjectTypes <ADSchemaObjectIdParameter[]
>] [-Deny <SwitchParameter>] [-ExtendedRights <ExtendedRightIdParameter[]
>] [-Identity <ADRawEntryIdParameter>] [-InheritanceType <None | All |
Descendents | SelfAndChildren | Children>] [-InheritedObjectType
<ADSchemaObjectIdParameter>] [-Properties <ADSchemaObjectIdParameter[]>]
[-User <SecurityPrincipalIdParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example grants Send As permissions for Aaron Painter to Terry Adams's mailbox.

```
Add-ADPermission -Identity "Terry Adams" -User AaronPainter
-AccessRights ExtendedRight -ExtendedRights "Send As"
```

EXAMPLE 2

This example configures the IP Secured Inbound Receive connector to accept anonymous SMTP messages.

Caution:

This example assumes that another security mechanism is used to ensure the Receive connector can't be used to send unsolicited commercial email messages. We recommend that you don't allow external clients to send messages anonymously through a Receive connector.

```
Add-AdPermission "IP Secured Inbound" -User "NT AUTHORITY
\ANONYMOUS LOGON" -ExtendedRights ms-Exch-SMTP-Submit,ms-
Exch-SMTP-Accept-Any-Recipient,ms-Exch-Bypass-Anti-Spam
```

Detailed Description

The **ADPermission** cmdlets can be used to directly modify Active Directory access control lists (ACLs). Although some Microsoft Exchange Server 2013 features may continue to use the **ADPermission** cmdlets to manage permissions, for example transport Send and Receive connectors, Exchange 2013 no longer uses customized ACLs to manage administrative permissions. If you want to grant or deny administrative permissions in Exchange 2013, you must use Role Based Access Control (RBAC). For more information about RBAC, see Permissions.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Legacy permissions" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ADRawEntryIdParameter	The *Identity* parameter specifies the identity of the object that's getting permissions added. You can specify either the distinguished name (DN) of the object or the object's name if it's unique. If the DN or name contains spaces, enclose the name in quotation marks (").
Instance	Required	Microsoft.Exchange.Management.RecipientTasks.ADAcePresentationObject	The *Instance* parameter enables you to pass an entire object to the command to be processed. It's mainly used in scripts where an entire object must be passed to the command.
Owner	Required	Microsoft.Exchange.Configuration.Tasks.SecurityPrincipalIdParameter	The *Owner* parameter specifies the owner of the Active Directory object. If the name of the owner contains spaces, enclose the name in quotation marks ("). The *Owner* parameter can only be used with the

			Identity parameter and no other parameters.
User	Required	Microsoft.Exchange.Configuration.Tasks.SecurityPrincipalIdParameter	The *User* parameter specifies the user that the permissions are being granted to on the object. If the name contains spaces, enclose the name in quotation marks (").
AccessRights	Optional	System.DirectoryServices.ActiveDirectoryRights[]	The *AccessRights* parameter specifies the rights needed to perform the operation. Valid values include: • `CreateChild` • `DeleteChild` • `ListChildren` • `Self` • `ReadProperty` • `WriteProperty` • `DeleteTree` • `ListObject` • `ExtendedRight` • `Delete` • `ReadControl` • `GenericExecute` • `GenericWrite` • `GenericRead` • `WriteDacl` • `WriteOwner` • `GenericAll` • `Synchronize` • `AccessSystemSecurity`
ChildObjectTypes	Optional	Microsoft.Exchange.Configuration.Tasks.ADSchemaObjectIdParameter[]	The *ChildObjectTypes* parameter specifies what type of object the permission should be applied to. The *ChildObjectTypes* parameter can only be

			used if the *AccessRights* parameter is set to `CreateChild` or `DeleteChild`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Deny	Optional	System.Management.Automation.SwitchParameter	The *Deny* switch specifies whether to deny permissions to the user on the Active Directory object.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge

			Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
ExtendedRights	Optional	Microsoft.Exchange.Configuration.Tasks.ExtendedRightIdParameter[]	The *ExtendedRights* parameter specifies the extended rights needed to perform the operation.
InheritanceType	Optional	System.DirectoryServices.ActiveDirectorySecurityInheritance	The *InheritanceType* parameter specifies whether permissions are inherited.
InheritedObjectType	Optional	Microsoft.Exchange.Configuration.Tasks.ADSchemaObjectIdParameter	The *InheritedObjectType* parameter specifies what kind of object inherits this access control entry (ACE).
Properties	Optional	Microsoft.Exchange.Configuration.Tasks.ADSchemaObjectIdParameter[]	The *Properties* parameter specifies what properties the object contains. The *Properties* parameter can only be used if the *AccessRights* parameter is set to `ReadProperty`, `writeProperty` or `Self`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ADPermission

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ADPermission** cmdlet to get permissions on an Active Directory object.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ADPermission [-User <SecurityPrincipalIdParameter>] <COMMON
PARAMETERS>

Get-ADPermission [-Owner <SwitchParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: -Identity <ADRawEntryIdParameter> [-DomainController
<Fqdn>]
```

Examples

EXAMPLE 1

This example returns the permissions that have been applied to the user Ed.

```
Get-ADPermission -Identity Ed
```

EXAMPLE 2

This example returns the permissions that have been granted to the user Chris on the Contoso.com Receive connector.

```
Get-ADPermission "Contoso.com" -User Chris
```

Detailed Description

The **ADPermission** cmdlets can be used to directly modify Active Directory access control lists (ACLs). Although some Microsoft Exchange Server 2013 features may continue to use the **ADPermission** cmdlets to manage permissions, for example transport Send and Receive connectors, Exchange 2013 no longer uses customized ACLs to manage administrative permissions. If you want to grant or deny administrative permissions in Exchange 2013, you must use Role Based Access Control (RBAC). For more information about RBAC, see Permissions.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Legacy permissions" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ADRawEntryIdParameter	The *Identity* parameter specifies the identity of the object for which you're retrieving permissions. You can retrieve the permissions for any Active Directory object

			using its distinguished name (DN). If the object is an Exchange object, you might be able to use the object's name. If the DN or the object's name contains spaces, enclose the value in quotation marks (").
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Owner	Optional	System.Management.Automation.SwitchParameter	The *Owner* switch specifies that the owner of the object specified in the *Identity* parameter should be returned. If the *Owner*

			switch is used, the *User* parameter can't be used.
User	Optional	Microsoft.Exchange.Configuration.Tasks.SecurityPrincipalIdParameter	The *User* parameter specifies that only the access control entries (ACEs) granted to the specified user on the object specified in the *Identity* parameter should be returned. If the *User* parameter is used, the *Owner* switch can't be used. If the name of the user contains spaces, enclose the name in quotation marks (").

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ADPermission

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-ADPermission** cmdlet to remove permissions from an Active Directory object.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ADPermission -Identity <ADRawEntryIdParameter> -User
<SecurityPrincipalIdParameter> [-AccessRights <ActiveDirectoryRights[]>]
[-ChildObjectTypes <ADSchemaObjectIdParameter[]>] [-Deny
<SwitchParameter>] [-ExtendedRights <ExtendedRightIdParameter[]>] [-
InheritanceType <None | All | Descendents | SelfAndChildren | Children>]
[-InheritedObjectType <ADSchemaObjectIdParameter>] [-Properties
<ADSchemaObjectIdParameter[]>] <COMMON PARAMETERS>


Remove-ADPermission -Identity <ADRawEntryIdParameter> <COMMON PARAMETERS>


Remove-ADPermission -Instance <ADAcePresentationObject> [-AccessRights
<ActiveDirectoryRights[]>] [-ChildObjectTypes <ADSchemaObjectIdParameter[]
>] [-Deny <SwitchParameter>] [-ExtendedRights <ExtendedRightIdParameter[]
>] [-Identity <ADRawEntryIdParameter>] [-InheritanceType <None | All |
Descendents | SelfAndChildren | Children>] [-InheritedObjectType
<ADSchemaObjectIdParameter>] [-Properties <ADSchemaObjectIdParameter[]>]
[-User <SecurityPrincipalIdParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the *Send As* permissions from user Kim on the user Administrator.

```
Remove-ADPermission -Identity Administrator -User Kim -
ExtendedRights "Send As"
```

EXAMPLE 2

This example removes the ability for anonymous users to send messages through the Receive connector IP Secured Inbound.

```
Remove-ADPermission "IP Secured Inbound" -User "NT
AUTHORITY\ANONYMOUS LOGON" -ExtendedRights ms-Exch-SMTP-
Submit,ms-Exch-SMTP-Accept-Any-Recipient,ms-Exch-Bypass-
Anti-Spam
```

Detailed Description

The **ADPermission** cmdlets can be used to directly modify Active Directory access control lists (ACLs). Although some Microsoft Exchange Server 2013 features may continue to use the

ADPermission cmdlets to manage permissions, for example transport Send and Receive connectors, Exchange 2013 no longer uses customized ACLs to manage administrative permissions. If you want to grant or deny administrative permissions in Exchange 2013, you must use Role Based Access Control (RBAC). For more information about RBAC, see Permissions.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Legacy permissions" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ADRawEntryIdParameter	The *Identity* parameter specifies the object from which the permission should be removed. You can specify either the distinguished name (DN) of the object or the object's name if it's unique. If the DN or name contains spaces, enclose the name in quotation marks (").
Instance	Required	Microsoft.Exchange.Management.RecipientTasks.ADAcePresentationObject	The *Instance* parameter enables you to pass an entire object to the command to be processed. It's mainly used in scripts where an entire object must be passed to the command.
User	Required	Microsoft.Exchange.Configuration.Tasks.Secu	The *User* parameter specifies the user object

		rityPrincipalIdParameter	that will have permissions removed.
AccessRights	Optional	System.DirectoryServices.ActiveDirectoryRights[]	The *AccessRights* parameter specifies the rights needed to perform the operation. Valid values include: • `CreateChild` • `DeleteChild` • `ListChildren` • `Self` • `ReadProperty` • `WriteProperty` • `DeleteTree` • `ListObject` • `ExtendedRight` • `Delete` • `ReadControl` • `GenericExecute` • `GenericWrite` • `GenericRead` • `WriteDacl` • `WriteOwner` • `GenericAll` • `Synchronize` • `AccessSystemSecurity`
ChildObjectTypes	Optional	Microsoft.Exchange.Configuration.Tasks.ADSchemaObjectIdParameter[]	The *ChildObjectTypes* parameter specifies what type of object the permission should be removed from. The *ChildObjectTypes* parameter can only be used if the *AccessRights* parameter is set to `CreateChild` or `DeleteChild`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that

			appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
Deny	Optional	System.Management.Automation.SwitchParameter	The *Deny* switch specifies whether the permission to remove is a deny permission.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
ExtendedRights	Optional	Microsoft.Exchange.Configuration.Tasks.Exte	The *ExtendedRights* parameter specifies the

		ndedRightIdParameter[]	extended rights to remove.
InheritanceType	Optional	System.DirectoryServices.ActiveDirectorySecurityInheritance	The *InheritanceType* parameter specifies whether permissions are inherited.
InheritedObjectType	Optional	Microsoft.Exchange.Configuration.Tasks.ADSchemaObjectIdParameter	The *InheritedObjectType* parameter specifies what kind of object inherits this access control entry (ACE).
Properties	Optional	Microsoft.Exchange.Configuration.Tasks.ADSchemaObjectIdParameter[]	The *Properties* parameter specifies what properties the object contains. The *Properties* parameter can only be used if the *AccessRights* parameter is set to `ReadProperty`, `WriteProperty`, or `Self`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ManagementRole

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ManagementRole** cmdlet to view management roles that have been created in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ManagementRole [-Cmdlet <String>] [-CmdletParameters <String[]>] [-
Identity <RoleIdParameter>] [-RoleType <Custom | UnScoped |
OrganizationManagement | RecipientManagement |
ViewOnlyOrganizationManagement | DistributionGroupManagement |
MyDistributionGroups | MyDistributionGroupMembership | UmManagement |
RecordsManagement | MyBaseOptions | UmRecipientManagement |
HelpdeskRecipientManagement | GALSynchronizationManagement |
ApplicationImpersonation | UMPromptManagement |
PartnerDelegatedTenantManagement | DiscoveryManagement |
CentralAdminManagement | UnScopedRoleManagement | MyContactInformation |
MyProfileInformation | MyVoiceMail | MyTextMessaging | MyMailSubscriptions
| MyRetentionPolicies | MyOptions | MailRecipients | FederatedSharing |
DatabaseAvailabilityGroups | Databases | PublicFolders | AddressLists |
RecipientPolicies | DisasterRecovery | Monitoring | DatabaseCopies |
UnifiedMessaging | Journaling | RemoteAndAcceptedDomains |
EmailAddressPolicies | TransportRules | SendConnectors | EdgeSubscriptions
| OrganizationTransportSettings | ExchangeServers |
ExchangeVirtualDirectories | ExchangeServerCertificates |
POP3AndIMAP4Protocols | ReceiveConnectors | UMMailboxes | UserOptions |
SecurityGroupCreationAndMembership | MailRecipientCreation |
MessageTracking | RoleManagement | ViewOnlyRecipients |
ViewOnlyConfiguration | DistributionGroups | MailEnabledPublicFolders |
MoveMailboxes | WorkloadManagement | ResetPassword | AuditLogs |
RetentionManagement | SupportDiagnostics | MailboxSearch | LegalHold |
MailTips | PublicFolderReplication | ActiveDirectoryPermissions |
UMPrompts | Migration | DataCenterOperations | TransportHygiene |
TransportQueues | Supervision | CmdletExtensionAgents |
OrganizationConfiguration | OrganizationClientAccess | ExchangeConnectors
| MailboxImportExport | ViewOnlyCentralAdminManagement |
ViewOnlyCentralAdminSupport | ViewOnlyRoleManagement | Reporting |
```

ViewOnlyAuditLogs | TransportAgents | DataCenterDestructiveOperations | InformationRightsManagement | LawEnforcementRequests | MyDiagnostics | MyMailboxDelegation | TeamMailboxes | MyTeamMailboxes | ActiveMonitoring | DataLossPrevention | MyFacebookEnabled | MyLinkedInEnabled | UserApplication | ArchiveApplication | LegalHoldApplication | OfficeExtensionApplication | TeamMailboxLifecycleApplication | CentralAdminCredentialManagement | PersonallyIdentifiableInformation | MailboxSearchApplication | MyMarketplaceApps | MyCustomApps | OrgMarketplaceApps | OrgCustomApps | ExchangeCrossServiceIntegration | NetworkingManagement>] <COMMON PARAMETERS>

Get-ManagementRole -Identity <RoleIdParameter> -Recurse <SwitchParameter> [-RoleType <Custom | UnScoped | OrganizationManagement | RecipientManagement | ViewOnlyOrganizationManagement | DistributionGroupManagement | MyDistributionGroups | MyDistributionGroupMembership | UmManagement | RecordsManagement | MyBaseOptions | UmRecipientManagement | HelpdeskRecipientManagement | GALSynchronizationManagement | ApplicationImpersonation | UMPromptManagement | PartnerDelegatedTenantManagement | DiscoveryManagement | CentralAdminManagement | UnScopedRoleManagement | MyContactInformation | MyProfileInformation | MyVoiceMail | MyTextMessaging | MyMailSubscriptions | MyRetentionPolicies | MyOptions | MailRecipients | FederatedSharing | DatabaseAvailabilityGroups | Databases | PublicFolders | AddressLists | RecipientPolicies | DisasterRecovery | Monitoring | DatabaseCopies | UnifiedMessaging | Journaling | RemoteAndAcceptedDomains | EmailAddressPolicies | TransportRules | SendConnectors | EdgeSubscriptions | OrganizationTransportSettings | ExchangeServers | ExchangeVirtualDirectories | ExchangeServerCertificates | POP3AndIMAP4Protocols | ReceiveConnectors | UMMailboxes | UserOptions | SecurityGroupCreationAndMembership | MailRecipientCreation | MessageTracking | RoleManagement | ViewOnlyRecipients | ViewOnlyConfiguration | DistributionGroups | MailEnabledPublicFolders | MoveMailboxes | WorkloadManagement | ResetPassword | AuditLogs | RetentionManagement | SupportDiagnostics | MailboxSearch | LegalHold | MailTips | PublicFolderReplication | ActiveDirectoryPermissions | UMPrompts | Migration | DataCenterOperations | TransportHygiene | TransportQueues | Supervision | CmdletExtensionAgents | OrganizationConfiguration | OrganizationClientAccess | ExchangeConnectors | MailboxImportExport | ViewOnlyCentralAdminManagement | ViewOnlyCentralAdminSupport | ViewOnlyRoleManagement | Reporting | ViewOnlyAuditLogs | TransportAgents | DataCenterDestructiveOperations | InformationRightsManagement | LawEnforcementRequests | MyDiagnostics | MyMailboxDelegation | TeamMailboxes | MyTeamMailboxes | ActiveMonitoring | DataLossPrevention | MyFacebookEnabled | MyLinkedInEnabled | UserApplication | ArchiveApplication | LegalHoldApplication | OfficeExtensionApplication | TeamMailboxLifecycleApplication | CentralAdminCredentialManagement | PersonallyIdentifiableInformation | MailboxSearchApplication | MyMarketplaceApps | MyCustomApps | OrgMarketplaceApps | OrgCustomApps | ExchangeCrossServiceIntegration | NetworkingManagement>] <COMMON PARAMETERS>

Get-ManagementRole [-Identity <RoleIdParameter>] [-Script <String>] [-ScriptParameters <String[]>] <COMMON PARAMETERS>

Get-ManagementRole -GetChildren <SwitchParameter> -Identity <RoleIdParameter> [-RoleType <Custom | UnScoped | OrganizationManagement | RecipientManagement | ViewOnlyOrganizationManagement | DistributionGroupManagement | MyDistributionGroups | MyDistributionGroupMembership | UmManagement | RecordsManagement | MyBaseOptions | UmRecipientManagement | HelpdeskRecipientManagement | GALSynchronizationManagement | ApplicationImpersonation | UMPromptManagement | PartnerDelegatedTenantManagement | DiscoveryManagement | CentralAdminManagement | UnScopedRoleManagement | MyContactInformation | MyProfileInformation | MyVoiceMail | MyTextMessaging | MyMailSubscriptions | MyRetentionPolicies | MyOptions | MailRecipients | FederatedSharing | DatabaseAvailabilityGroups | Databases | PublicFolders | AddressLists | RecipientPolicies | DisasterRecovery | Monitoring | DatabaseCopies | UnifiedMessaging | Journaling | RemoteAndAcceptedDomains | EmailAddressPolicies | TransportRules | SendConnectors | EdgeSubscriptions | OrganizationTransportSettings | ExchangeServers | ExchangeVirtualDirectories | ExchangeServerCertificates | POP3AndIMAP4Protocols | ReceiveConnectors | UMMailboxes | UserOptions | SecurityGroupCreationAndMembership | MailRecipientCreation | MessageTracking | RoleManagement | ViewOnlyRecipients | ViewOnlyConfiguration | DistributionGroups | MailEnabledPublicFolders | MoveMailboxes | WorkloadManagement | ResetPassword | AuditLogs |

```
RetentionManagement | SupportDiagnostics | MailboxSearch | LegalHold |
MailTips | PublicFolderReplication | ActiveDirectoryPermissions |
UMPrompts | Migration | DataCenterOperations | TransportHygiene |
TransportQueues | Supervision | CmdletExtensionAgents |
OrganizationConfiguration | OrganizationClientAccess | ExchangeConnectors
| MailboxImportExport | ViewOnlyCentralAdminManagement |
ViewOnlyCentralAdminSupport | ViewOnlyRoleManagement | Reporting |
ViewOnlyAuditLogs | TransportAgents | DataCenterDestructiveOperations |
InformationRightsManagement | LawEnforcementRequests | MyDiagnostics |
MyMailboxDelegation | TeamMailboxes | MyTeamMailboxes | ActiveMonitoring |
DataLossPrevention | MyFacebookEnabled | MyLinkedInEnabled |
UserApplication | ArchiveApplication | LegalHoldApplication |
OfficeExtensionApplication | TeamMailboxLifecycleApplication |
CentralAdminCredentialManagement | PersonallyIdentifiableInformation |
MailboxSearchApplication | MyMarketplaceApps | MyCustomApps |
OrgMarketplaceApps | OrgCustomApps | ExchangeCrossServiceIntegration |
NetworkingManagement>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example lists all the roles that have been created in your organization.

```
Get-ManagementRole
```

EXAMPLE 2

This example lists all the roles that are children of the Mail Recipients management role. The command performs a recursive query of all the child roles of the specified parent role. This recursive query finds every child role from the immediate children of the parent to the last child role in the hierarchy. In a recursive list, the parent role is also returned in the list.

```
Get-ManagementRole "Mail Recipients" -Recurse
```

EXAMPLE 3

This example lists all the roles that contain both the *Identity* and *Database* parameters. Roles that contain only one parameter or the other aren't returned.

```
Get-ManagementRole -CmdletParameters Identity, Database
```

EXAMPLE 4

This example lists all the roles that have a type of `UnScopedTopLevel`. These roles contain custom scripts or non-Exchange cmdlets.

```
Get-ManagementRole -RoleType UnScopedTopLevel
```

EXAMPLE 5

This example retrieves only the Transport Rules role and passes the output of the **Get-ManagementRole** cmdlet to the **Format-List** cmdlet. The **Format-List** cmdlet then shows only the *Name* and *RoleType* properties of the Transport Rules role. For more information about pipelining and the **Format-List** cmdlet, see Pipelining and Working with command output.

```
Get-ManagementRole "Transport Rules" | Format-List Name,
RoleType
```

EXAMPLE 6

This example lists the immediate children of the Mail Recipients role. Only the child roles that hold the Mail Recipients role as their parent role are returned. The Mail Recipients role isn't returned in the list.

```
Get-ManagementRole "Mail Recipients" -GetChildren
```

Detailed Description

You can view management roles in several ways, from listing all the roles in your organization to listing only the child roles of a specified parent role. You can also view the details of a specific role by piping the output of the **Get-ManagementRole** cmdlet to the **Format-List** cmdlet.

For more information about management roles, see Understanding management roles.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
GetChildren	Required	System.Management.Automation.SwitchParameter	The *GetChildren* parameter retrieves a list of all the roles that were created based on the parent role specified in the *Identity* parameter. Only the immediate child roles of the parent role are

			included. The *GetChildren* parameter can only be used with the *Identity* and *RoleType* parameters.
Recurse	Required	System.Management.Automation.SwitchParameter	The *Recurse* parameter retrieves a list of all the roles that were created based on the parent role specified in the *Identity* parameter. The role specified in the *Identity* parameter, its child roles, and their children are returned. The *Recurse* parameter can only be used with the *Identity* and *RoleType* parameters.
Cmdlet	Optional	System.String	The *Cmdlet* parameter returns a list of all roles that include the specified cmdlet.
CmdletParameters	Optional	System.String[]	The *CmdletParameters* parameter returns a list of all roles that include the specified parameter or parameters. You can specify more than one parameter by separating each

			parameter with a comma. If you specify multiple parameters, only the roles that include all of the specified parameters are returned.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.RoleIdParameter	The *Identity* parameter specifies the role you want to view. If the role you want to view contains spaces, enclose the name in quotation marks ("). You can use the wildcard character (*) and a partial role name to match multiple roles.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.Orga	The *Organization* parameter is reserved

		nizationIdParameter	for internal Microsoft use.
RoleType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.RoleType	The *RoleType* parameter returns a list of roles that match the specified role type. For a list of valid role types, see Understanding management roles.
Script	Optional	System.String	The *Script* parameter returns a list of all roles that include the specified script.
ScriptParameters	Optional	System.String[]	The *ScriptParameters* parameter returns a list of all roles that include the specified parameter or parameters. You can specify more than one parameter by separating each parameter with a comma. If you specify multiple parameters, only the roles that include all of the specified parameters are returned.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ManagementRole

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-ManagementRole** cmdlet to create a management role based on an existing role or create an unscoped management role.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-ManagementRole -Parent <RoleIdParameter> <COMMON PARAMETERS>

New-ManagementRole -UnScopedTopLevel <SwitchParameter> <COMMON PARAMETERS>

COMMON PARAMETERS: -Name <String> [-Confirm [<SwitchParameter>]] [-
Description <String>] [-DomainController <Fqdn>] [-Force
<SwitchParameter>] [-Organization <OrganizationIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the management role Redmond Journaling View-Only based on the Journaling parent role.

```
New-ManagementRole -Name "Redmond Journaling View-Only" -
Parent Journaling
```

After the role is created, the **Remove-ManagementRoleEntry** cmdlet is used along with the **Where** cmdlet to remove all the management role entries that aren't needed on the role. You can't add role entries to the newly created role because it already has all the role entries that exist on its parent role, Journaling. The *WhatIf* switch is used to verify that the correct role entries are removed.

```
Get-ManagementRoleEntry "Redmond Journaling View-Only\*" |
Where { $_.Name -NotLike "Get*" } | Remove-
ManagementRoleEntry -WhatIf
```

After confirmation that the command removes the correct role entries, the same command is run again without the *WhatIf* switch.

```
Get-ManagementRoleEntry "Redmond Journaling View-Only\*" |
Where { $_.Name -NotLike "Get*" } | Remove-
ManagementRoleEntry
```

For more information about pipelining and the **Where** cmdlet, see the following topics:

- Pipelining
- Working with command output

EXAMPLE 2

This example creates the unscoped management role In-house scripts. The user running the command, or the role group the user is a member of, is assigned the Unscoped Role Management management role. This assignment is required to use the *UnScopedTopLevel* switch.

```
New-ManagementRole -Name "In-house scripts" -
UnScopedTopLevel
```

Detailed Description

You can either create a management role based on an existing role, or you can create an unscoped role that's empty. If you create a role based on an existing role, you start with the management role entries that exist on the existing role. You can then remove entries to customize the role. If you create an unscoped role, the role can contain custom scripts or cmdlets that aren't part of Exchange.

Caution:
An unscoped role doesn't have any scope restrictions applied. Scripts or third-party cmdlets included in an unscoped role can view or modify any object in the Exchange organization. The ability to create an unscoped management role isn't granted by default. To create an unscoped management role, you must assign the Unscoped Role Management management role to a role group you're a member of. For more information about how to create an unscoped management role, see Create an unscoped role.

After you create a role, you can change the management role entries on the role and assign the role with a management scope to a user or universal security group (USG).

For more information about management roles, see Understanding management roles.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name of the role. The maximum length of the name is 64 characters. If the name contains spaces, enclose the name in quotation marks (").
Parent	Required	Microsoft.Exchange.Configuration.Tasks.RoleIdParameter	The *Parent* parameter specifies the identity of the role to copy. If the name of the role contains spaces, enclose the name in quotation marks ("). If you specify the *Parent* parameter, you can't use the *UnScopedTopLevel* switch.
UnScopedTopLevel	Required	System.Management.Automation.SwitchParameter	The *UnScopedTopLevel* switch specifies that the role should be a custom, empty role. If you specify the

			UnScopedTopLevel switch, you can't use the *Parent* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies the description that's displayed when the management role is viewed using the **Get-ManagementRole** cmdlet. Enclose the description in quotation marks (").
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on

			the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ManagementRole

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-ManagementRole** cmdlet to remove custom management roles that you don't need anymore.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ManagementRole -Identity <RoleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-Recurse <SwitchParameter>] [-UnScopedTopLevel <SwitchParameter>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the single role ExampleRole1.

```
Remove-ManagementRole ExampleRole1
```

EXAMPLE 2

This example runs the **Remove-ManagementRole** cmdlet with the *WhatIf* switch. The *WhatIf* switch lets the command run as if it were going to perform the action you specified but doesn't commit any changes. Instead, it displays the results of what would have happened, so you can verify that the actions are correct.

```
Remove-ManagementRole ExampleRole2 -Recurse -WhatIf
```

If the results are as expected, the following command can be used to remove the `ExampleRole2` parent role and all its child roles.

```
Remove-ManagementRole ExampleRole2 -Recurse
```

EXAMPLE 3

This example uses the **Get-ManagementRole** cmdlet to get a list of roles that contain the string "Example" in the role name, and then pipes the list to the **Remove-ManagementRole** cmdlet. The **Remove-ManagementRole** cmdlet, because the *WhatIf* switch is specified, displays the roles that would have been removed but doesn't commit any changes. If the results are as expected, the command can be run again without the *WhatIf* switch to remove the roles.

```
Get-ManagementRole *Example* | Remove-ManagementRole -
WhatIf
```

EXAMPLE 4

This example removes the In-house scripts unscoped top-level management role. Because this is an unscoped top-level role, the *UnScopedTopLevel* switch must be used.

```
Remove-ManagementRole "In-house scripts" -UnScopedTopLevel
```

For more information about unscoped top-level management roles, see Understanding management roles.

Detailed Description

You need to remove all the management role assignments from a role before you delete it. If the role is the parent of child roles, the child roles must be removed before you remove the parent role, or you must use the *Recurse* parameter when you remove the parent role. You can only remove custom roles. Built-in roles, such as the Mail Recipients role, can't be removed. For more information about how to remove a custom role, see Remove a role.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Management roles" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleIdParameter	The *Identity* parameter specifies the custom role to remove. If the name of the role contains spaces, enclose the name in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
Recurse	Optional	System.Management.Automation.SwitchParameter	The *Recurse* parameter removes all child roles of the role specified with the *Identity* parameter, and then removes the specified role. **Caution:** The *Recurse* parameter removes all child roles of the specified role. We recommend that you first

			use the command with the *WhatIf* switch to confirm that the action to be taken is correct.
UnScopedTopLevel	Optional	System.Management.Automation.SwitchParameter	The *UnScopedTopLevel* switch specifies that the role you're trying to remove is an unscoped top-level role. You must use this switch if you want to remove an unscoped top-level role.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ManagementRoleAssignment

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ManagementRoleAssignment** cmdlet to retrieve management role assignments.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ManagementRoleAssignment [-AssignmentMethod <AssignmentMethod[]>] [-
Role <RoleIdParameter>] [-RoleAssignee <RoleAssigneeIdParameter>] <COMMON
PARAMETERS>


Get-ManagementRoleAssignment [-Identity <RoleAssignmentIdParameter>]
<COMMON PARAMETERS>


COMMON PARAMETERS: [-ConfigWriteScope <None | NotApplicable |
OrganizationConfig | CustomConfigScope | PartnerDelegatedTenantScope |
ExclusiveConfigScope>] [-CustomConfigWriteScope
<ManagementScopeIdParameter>] [-CustomRecipientWriteScope
<ManagementScopeIdParameter>] [-Delegating <$true | $false>] [-
DomainController <Fqdn>] [-Enabled <$true | $false>] [-Exclusive <$true |
$false>] [-ExclusiveConfigWriteScope <ManagementScopeIdParameter>] [-
ExclusiveRecipientWriteScope <ManagementScopeIdParameter>] [-
GetEffectiveUsers <SwitchParameter>] [-IgnoreDehydratedFlag
<SwitchParameter>] [-Organization <OrganizationIdParameter>] [-
RecipientOrganizationalUnitScope <OrganizationalUnitIdParameter>] [-
RecipientWriteScope <None | NotApplicable | Organization | MyGAL | Self |
MyDirectReports | OU | CustomRecipientScope | MyDistributionGroups |
MyExecutive | ExclusiveRecipientScope | MailboxICanDelegate>] [-
RoleAssigneeType <User | SecurityGroup | RoleAssignmentPolicy |
MailboxPlan | ForeignSecurityPrincipal | RoleGroup |
PartnerLinkedRoleGroup | LinkedRoleGroup | Computer>] [-WritableDatabase
<DatabaseIdParameter>] [-WritableRecipient <GeneralRecipientIdParameter>]
[-WritableServer <ServerIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves the Denver Help Desk role assignment using the **Get-ManagementRoleAssignment** cmdlet and pipes the output to the **Format-List** cmdlet. For more information about the **Format-List** cmdlet, see Working with command output.

```
Get-ManagementRoleAssignment "Denver Help Desk" | Format-
List
```

EXAMPLE 2

This example retrieves all the role assignments that are enabled and have been designated as delegating role assignments.

```
Get-ManagementRoleAssignment -Enabled $true -Delegating
$true
```

EXAMPLE 3

This example retrieves all the role assignments that include the MyGAL recipient-based scope restriction type.

```
Get-ManagementRoleAssignment -RecipientWriteScope MyGAL
```

EXAMPLE 4

This example retrieves all the role assignments associated with the Organization Management management role.

```
Get-ManagementRoleAssignment -Role "Mail Recipients"
```

EXAMPLE 5

This example retrieves a list of all the users and the role assignments that can modify the recipient Bob.

```
Get-ManagementRoleAssignment -WritableRecipient Bob -
GetEffectiveUsers
```

EXAMPLE 6

This example retrieves a list of all exclusive scopes that can modify server objects that match Redmond Executive Servers. The command also lists the users who are effectively assigned the role assignments through role groups or USGs.

```
Get-ManagementRoleAssignment -ExclusiveConfigWriteScope
"Redmond Executive Servers" -GetEffectiveUsers
```

EXAMPLE 7

This example retrieves all the role assignments that can modify the database Contoso Sales.

```
Get-ManagementRoleAssignment -WritableDatabase "Contoso
Sales"
```

Detailed Description

You can retrieve role assignments in a variety of ways including by assignment type, scope type, or name, and whether the assignment is enabled or disabled. You can also view a list of role assignments that provide access to a specified recipient, server, or database.

For more information about management role assignments, see Understanding management role assignments.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role assignments" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
AssignmentMethod	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AssignmentMethod[]	The *AssignmentMethod* parameter specifies the type of role assignment to include in the results returned by the cmdlet. You can specify one or more of the following values: • `Direct` • `SecurityGroup` • `RoleGroup` • `RoleAssignmentPolicy` If you provide more than one value, separate each value with a comma. You must specify a value with the *RoleAssignee* parameter if you use the *AssignmentMethod* parameter.
ConfigWriteScope	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ConfigWrit	The *ConfigWriteScope* parameter specifies the type of management

		eScopeType	configuration scope to include in the results returned by the cmdlet. The valid values are `None`, `OrganizationConfig`, `CustomConfigScope`, and `ExclusiveConfigScope`.
CustomConfigWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	This parameter is available only in on-premises Exchange 2013. The *CustomConfigWriteScope* parameter returns only the regular role assignments that include the specified configuration-based regular scope. This parameter can only be used to retrieve regular configuration-based scopes. To retrieve a list of exclusive configuration-based scopes, use the *ExclusiveConfigWriteScope* parameter instead. If the scope name contains spaces, enclose it in quotation marks (").
CustomRecipientWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.Man	The *CustomRecipientWriteSco*

		agementScopeIdParameter	*pe* parameter returns only the regular role assignments that include the specified recipient-based regular scope. This parameter can only be used to retrieve regular recipient-based scopes. To retrieve a list of exclusive recipient-based scopes, use the *ExclusiveRecipientWriteScope* parameter instead. If the scope name contains spaces, enclose it in quotation marks (").
Delegating	Optional	System.Boolean	The *Delegating* parameter specifies whether delegating or regular role assignments should be returned. By default, both delegating and regular scopes are returned. To return only delegating role assignments, specify a value of `$true`. To return only regular role assignments, specify a value of `$false`.
DomainController	Optional	Microsoft.Exchange.Da	This parameter is

		ta.Fqdn	available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether enabled or disabled role assignments should be returned. To return enabled role assignments, specify a value of `$true`. To return disabled role assignments, specify a value of `$false`.
Exclusive	Optional	System.Boolean	The *Exclusive* parameter specifies whether exclusive or regular role assignments should be returned. By default, both exclusive and regular scopes are returned. To return only exclusive role assignments, specify a value of `$true`. To return only regular role

			assignments, specify a value of `$false`.
ExclusiveConfigWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	This parameter is available only in on-premises Exchange 2013. The *ExclusiveConfigWriteScope* parameter returns only the exclusive role assignments that include the specified configuration-based exclusive scope. This parameter can only be used to retrieve exclusive configuration-based scopes. To retrieve a list of regular configuration-based scopes, use the *CustomConfigWriteScope* parameter instead. If the scope name contains spaces, enclose it in quotation marks (").
ExclusiveRecipientWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	The *ExclusiveRecipientWriteScope* parameter returns only the exclusive role assignments that include the specified recipient-

			based exclusive scope. This parameter can only be used to retrieve exclusive recipient-based scopes. To retrieve a list of regular recipient-based scopes, use the *CustomRecipientWriteScope* parameter instead. If the scope name contains spaces, enclose it in quotation marks (").
GetEffectiveUsers	Optional	System.Management.Automation.SwitchParameter	The *GetEffectiveUsers* switch specifies that the command should show the list of users in the role groups, assignment policies, or USGs associated with a role assignment. The users are effectively assigned the role assignment through their role group, assignment policy, or USG.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.RoleAssignmentIdParameter	The *Identity* parameter specifies the name of the role assignment to retrieve. If the name of the role assignment contains spaces, enclose it in

			quotation marks ("). If the *RoleAssignee* parameter is used, you can't use the *Identity* parameter.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
RecipientOrganizationalUnitScope	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientOrganizationalUnitScope* parameter returns only the role assignments that include the specified organizational unit (OU). If the OU tree contains spaces, enclose it in quotation marks (").
RecipientWriteScope	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.RecipientWriteScopeType	The *RecipientWriteScope* parameter returns only the role assignments associated with the recipient scope restriction type specified. The valid values are `None`, `MyGAL`, `Self`, `OU`, `CustomRecipientScope`, `MyDistributionGroups`, and

			`ExclusiveRecipientScope`.
Role	Optional	Microsoft.Exchange.Configuration.Tasks.RoleIdParameter	The *Role* parameter returns only the role assignments associated with the specified management role. If the name of the role contains spaces, enclose it in quotation marks (").
RoleAssignee	Optional	Microsoft.Exchange.Configuration.Tasks.RoleAssigneeIdParameter	The *RoleAssignee* parameter specifies the role group, assignment policy, user, or universal security group (USG) for which you want to view role assignments. If the *RoleAssignee* parameter is used, you can't use the *Identity* parameter. By default, the command returns both direct role assignments to the role assignee, and indirect role assignments granted to a role assignee through role groups or assignment policies. If the name of the user or USG contains spaces, enclose it in quotation marks (").

RoleAssigneeType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.RoleAssigneeType	The *RoleAssigneeType* parameter specifies the type of role assignee to return. The valid values are `User`, `SecurityGroup`, `RoleAssignmentPolicy`, `ForeignSecurityPrincipal`, `RoleGroup`, `LinkedRoleGroup`, and `Computer`.
WritableDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *WritableDatabase* parameter specifies the database object you want to test to determine which role assignments allow it to be modified. The command takes into account the roles and scopes associated with each role assignment. If the database name contains spaces, enclose it in quotation marks ("). If this parameter is used with the *GetEffectiveUsers* switch, all the users who can modify the database object indirectly through role groups and USGs are

			also returned. Without the *GetEffectiveUsers* switch, only the role groups, users, and USGs directly assigned the role assignment are returned.
WritableRecipient	Optional	Microsoft.Exchange.Configuration.Tasks.GeneralRecipientIdParameter	The *WritableRecipient* parameter specifies the recipient object you want to test to determine which role assignments allow it to be modified. The command takes into account the roles and scopes associated with each role assignment. If the recipient name contains spaces, enclose it in quotation marks ("). If this parameter is used with the *GetEffectiveUsers* switch, all of the users who can modify the recipient object indirectly through role groups and USGs are also returned. Without the *GetEffectiveUsers* switch, only the role groups, users, and USGs directly assigned the role assignment are returned.

WritableServer	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	This parameter is available only in on-premises Exchange 2013. The *WritableServer* parameter specifies the server object you want to test to determine which role assignments allow it to be modified. The command takes into account the roles and scopes associated with each role assignment. If the server object name contains spaces, enclose it in quotation marks ("). If this parameter is used with the *GetEffectiveUsers* switch, all of the users who can modify the server object indirectly through role groups and USGs are also returned. Without the *GetEffectiveUsers* switch, only the role groups, users, and USGs directly assigned the role assignment are returned.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ManagementRoleAssignment

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-ManagementRoleAssignment** cmdlet to assign a management role to a management role group, management role assignment policy, user, or universal security group (USG).

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-ManagementRoleAssignment -User <UserIdParameter> [-Delegating
<SwitchParameter>] <COMMON PARAMETERS>


New-ManagementRoleAssignment -SecurityGroup <SecurityGroupIdParameter> [-
Delegating <SwitchParameter>] <COMMON PARAMETERS>


New-ManagementRoleAssignment -Policy <MailboxPolicyIdParameter> <COMMON
PARAMETERS>


New-ManagementRoleAssignment -Computer <ComputerIdParameter> <COMMON
PARAMETERS>


COMMON PARAMETERS: -Role <RoleIdParameter> [-Confirm [<SwitchParameter>]]
[-CustomConfigWriteScope <ManagementScopeIdParameter>] [-
CustomRecipientWriteScope <ManagementScopeIdParameter>] [-DomainController
<Fqdn>] [-ExclusiveConfigWriteScope <ManagementScopeIdParameter>] [-
ExclusiveRecipientWriteScope <ManagementScopeIdParameter>] [-Force
<SwitchParameter>] [-IgnoreDehydratedFlag <SwitchParameter>] [-Name
<String>] [-Organization <OrganizationIdParameter>] [-
RecipientOrganizationalUnitScope <OrganizationalUnitIdParameter>] [-
RecipientRelativeWriteScope <None | NotApplicable | Organization | MyGAL |
Self | MyDirectReports | OU | CustomRecipientScope | MyDistributionGroups
| MyExecutive | ExclusiveRecipientScope | MailboxICanDelegate>] [-
UnScopedTopLevel <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example assigns the Mail Recipients role to the Tier 2 Help Desk role group.

```
New-ManagementRoleAssignment -Role "Mail Recipients" -
SecurityGroup "Tier 2 Help Desk"
```

EXAMPLE 2

This example assigns the MyVoiceMail role to the "Sales end-users" role assignment policy. First, the **IsEndUserRole** property on the MyVoiceMail role is verified to be sure it's set to `$true`, indicating it's an end-user role:

```
Get-ManagementRole "MyVoiceMail" | Format-Table Name,
IsEndUserRole
```

After the role has been verified to be an end-user role, the role is assigned to the "Sales end-users" role assignment policy.

```
New-ManagementRoleAssignment -Role "MyVoiceMail" -Policy
"Sales end-users"
```

EXAMPLE 3

This example assigns the Eng Help Desk role to the Eng HD Personnel role group. The assignment restricts the recipient write scope of the role to the contoso.com/Engineering/Users OU. Users who are members of the Eng HD Personnel role group can only create, modify, or remove objects contained within that OU.

```
New-ManagementRoleAssignment -Role "Eng Help Desk" -
SecurityGroup "Eng HD Personnel" -
RecipientOrganizationalUnitScope contoso.com/Engineering/
Users
```

EXAMPLE 4

This example assigns the Distribution Groups role to the North America Exec Assistants role group. The assignment restricts the recipient write scope of the role to the scope specified in the North America Recipients custom recipient management scope. Users who are members of the North America Exec Assistants role group can only create, modify, or remove distribution group objects that match the specified custom recipient management scope.

```
New-ManagementRoleAssignment -Role "Distribution Groups" -
SecurityGroup "North America Exec Assistants" -
```

```
CustomRecipientWriteScope "North America Recipients"
```

EXAMPLE 5

This example assigns the Exchange Servers role to John. Because John should only manage the servers running Exchange located in Sydney, the role assignment restricts the configuration write scope of the role to the scope specified in the Sydney Servers custom configuration role group. John can only manage servers that match the specified custom configuration management scope.

```
New-ManagementRoleAssignment -Name "Exchange Servers_John"
-Role "Exchange Servers" -User John -CustomConfigWriteScope
"Sydney Servers"
```

EXAMPLE 6

This example assigns the Mail Recipients role to the Executive Administrators role group. The assignment restricts the recipient write scope of the role to the scope specified in the Exclusive-Executive Recipients exclusive recipient management scope. Because the Exclusive-Executive Recipients scope is an exclusive scope, only users of the Executive Administrators can manage the executive recipients that match the exclusive recipient scope. No other users, unless they're also assigned an assignment that uses an exclusive scope that matches the same users, can modify the executive recipients.

```
New-ManagementRoleAssignment -Name "Excl-Mail
Recipients_Executive Administrators" -Role "Mail
Recipients" -SecurityGroup "Executive Administrators" -
ExclusiveRecipientWriteScope "Exclusive-Executive
Recipients"
```

EXAMPLE 7

This example assigns the Mail Recipients role to the Contoso Sub - Seattle role group. The administrators in this role group should only be allowed to create and manage mail recipients in specific databases that have been allocated for use by the Contoso subsidiary, A. Datum Corporation (adatum.com). Also, this group of administrators should only be allowed to manage the Contoso employees located in the Seattle office. This is done by creating a role assignment with both a database scope, to limit management of mail recipients to only the databases in the database scope, and a recipient OU scope, to limit access to only the recipient objects within the Contoso Seattle OU.

```
New-ManagementRoleAssignment -Name "Mail Recipients_Contoso
Seattle" -Role "Mail Recipients" -SecurityGroup "Contoso
Sub - Seattle" -CustomConfigWriteScope "Contoso Databases"
-RecipientOrganizationalUnitScope adatum.com/Contoso/
```

```
Seattle/Users
```

Detailed Description

When you add a new role assignment, you can specify a built-in or custom role that was created using the **New-ManagementRole** cmdlet and specify an organizational unit (OU) or predefined or custom management scope to restrict the assignment.

You can create custom management scopes using the **New-ManagementScope** cmdlet and can view a list of existing scopes using the **Get-ManagementScope** cmdlet. If you choose not to specify an OU, or predefined or custom scope, the implicit write scope of the role applies to the role assignment.

For more information about management role assignments, see Understanding management role assignments.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role assignments" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Computer	Required	Microsoft.Exchange.Configuration.Tasks.ComputerIdParameter	This parameter is available only in on-premises Exchange 2013. The *Computer* parameter specifies the name of the computer to assign the management role to. If you specify the *Computer* parameter, you can't specify the *SecurityGroup*, *User*, or *Policy* parameters.

Policy	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Policy* parameter specifies the name of the management role assignment policy to assign the management role to. The **IsEndUserRole** property of the role you specify using the *Role* parameter must be set to `$true`. If you specify the *Policy* parameter, you can't specify the *SecurityGroup*, *Computer*, or *User* parameters. If the policy name contains spaces, enclose the name in quotation marks (").
Role	Required	Microsoft.Exchange.Configuration.Tasks.RoleIdParameter	The *Role* parameter specifies the existing role to assign. If the role name contains spaces, enclose the name in quotation marks (").
SecurityGroup	Required	Microsoft.Exchange.Configuration.Tasks.SecurityGroupIdParameter	The *SecurityGroup* parameter specifies the name of the management role group or universal USG to assign the management

			role to. If you specify the *SecurityGroup* parameter, you can't specify the *Policy*, *Computer*, or *User* parameters. If the role group or USG name contains spaces, enclose the name in quotation marks (").
User	Required	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	The *User* parameter specifies the name or alias of the user to assign the management role to. If you specify the *User* parameter, you can't specify the *SecurityGroup*, *Computer*, or *Policy* parameters. If the value contains spaces, enclose the name in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before

			processing continues. You don't have to specify a value with the *Confirm* switch.
CustomConfigWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	This parameter is available only in on-premises Exchange 2013. The *CustomConfigWriteScope* parameter specifies the existing configuration scope to associate with this management role assignment. If you use the *CustomConfigWriteScope* parameter you can't use the *ExclusiveConfigWriteScope* parameter. If the management scope name contains spaces, enclose the name in quotation marks (").
CustomRecipientWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	The *CustomRecipientWriteScope* parameter specifies the existing recipient-based management scope to associate with

			this management role assignment. If the management scope name contains spaces, enclose the name in quotation marks ("). If you use the *CustomRecipientWriteScope* parameter, you can't use the *RecipientOrganizationalUnitScope* or *ExclusiveRecipientWriteScope* parameters.
Delegating	Optional	System.Management.Automation.SwitchParameter	The *Delegating* parameter specifies whether the user or USG assigned to the role can delegate the role to other users or groups. You don't have to specify a value with the *Delegating* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
ExclusiveConfigWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	This parameter is available only in on-premises Exchange 2013. The *ExclusiveConfigWriteScope* parameter specifies the exclusive configuration-based management scope to associate with the new role assignment. If you use the *ExclusiveConfigWriteScope* parameter, you can't use the *CustomConfigWriteScope* parameter. If the scope name contains spaces, enclose the name in quotation marks (").
ExclusiveRecipientWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	The *ExclusiveRecipientWriteScope* parameter specifies the exclusive recipient-based management scope to

			associate with the new role assignment. If you use the *ExclusiveRecipientWriteScope* parameter, you can't use the *CustomRecipientWriteScope* or *RecipientOrganizationalUnitScope* parameters. If the scope name contains spaces, enclose the name in quotation marks (").
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.

IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Name	Optional	System.String	The *Name* parameter specifies a name for the new management role assignment. The maximum length of the name is 64 characters. If the management role assignment name contains spaces, enclose the name in quotation marks ("). If you don't specify a name, one will be created automatically.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
RecipientOrganizationalUnitScope	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientOrganizationalUnitScope* parameter specifies the OU to scope the new role assignment to. If you use the *RecipientOrganizationalUnitScope* parameter, you can't use the

			CustomRecipientWriteScope or *ExclusiveRecipientWriteScope* parameters. To specify an OU, use the syntax: *domain/ou*. If the OU name contains spaces, enclose the domain and OU in quotation marks (").
RecipientRelativeWriteScope	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.RecipientWriteScopeType	The *RecipientRelativeWriteScope* parameter specifies the type of restriction to apply to a recipient scope. The available types are `None`, `Organization`, `MyGAL`, `Self`, and `MyDistributionGroups`. The *RecipientRelativeWriteScope* parameter is automatically set when the *CustomRecipientWriteScope* or *RecipientOrganizationalUnitScope* parameters are used. **Note:** Even though the `NotApplicable`, `OU`,

			`MyDirectReports`, `CustomRecipientScope`, `MyExecutive`, `MailboxICanDelegate` and `ExclusiveRecipientScope` values appear in the syntax block for this parameter, they can't be used directly on the command line. They are used internally by the cmdlet.
UnScopedTopLevel	Optional	System.Management.Automation.SwitchParameter	The *UnScopedTopLevel* switch specifies that the role provided with the *Role* parameter is an unscoped top level management role. You can only create a role assignment using the *UnScopedTopLevel* switch if the role specified using the *Role* parameter is an unscoped top level role.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ManagementRoleAssignment

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-ManagementRoleAssignment** cmdlet to remove management role assignments.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ManagementRoleAssignment -Identity <RoleAssignmentIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Force
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the Recipients_Seattle Recipient Management role assignment.

```
Remove-ManagementRoleAssignment "Recipients_Seattle
Recipient Management"
```

EXAMPLE 2

This example retrieves a list of role assignments that begin with the string "Detroit" and attempts to remove them with the **Remove-ManagementRoleAssignment** cmdlet. Because the *WhatIf* switch is included with the **Remove-ManagementRoleAssignment** command, the command displays the changes that would have occurred but doesn't commit any changes.

```
Get-ManagementRoleAssignment Detroit* | Remove-
ManagementRoleAssignment -WhatIf
```

After the list of role assignments to be removed is confirmed, this command is used to remove the role assignments.

```
Get-ManagementRoleAssignment Detroit* | Remove-
ManagementRoleAssignment
```

Detailed Description

When you remove a role assignment, the management role group, management role assignment, user, or universal security group (USG) that was assigned the associated role can no longer access the cmdlets or parameters made available by the role. For more information about management role assignments, see Understanding management role assignments.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role assignments" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleAssignmentIdParameter	The *Identity* parameter specifies the name of the role assignment to remove. If the role assignment name contains spaces, enclose the name in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchPara	The *Confirm* switch causes the command to

		meter	pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't

			provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ManagementRoleAssignment

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-ManagementRoleAssignment** cmdlet to modify existing management role assignments.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ManagementRoleAssignment [-CustomConfigWriteScope
<ManagementScopeIdParameter>] [-RecipientRelativeWriteScope <None |
NotApplicable | Organization | MyGAL | Self | MyDirectReports | OU |
CustomRecipientScope | MyDistributionGroups | MyExecutive |
ExclusiveRecipientScope | MailboxICanDelegate>] <COMMON PARAMETERS>


Set-ManagementRoleAssignment [-CustomConfigWriteScope
<ManagementScopeIdParameter>] [-CustomRecipientWriteScope
<ManagementScopeIdParameter>] <COMMON PARAMETERS>


Set-ManagementRoleAssignment [-CustomConfigWriteScope
<ManagementScopeIdParameter>] [-RecipientOrganizationalUnitScope
<OrganizationalUnitIdParameter>] <COMMON PARAMETERS>


Set-ManagementRoleAssignment [-ExclusiveConfigWriteScope
<ManagementScopeIdParameter>] [-ExclusiveRecipientWriteScope
<ManagementScopeIdParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Identity <RoleAssignmentIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-Force <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the Mail Recipients_Denver Help Desk role assignment. When a role assignment is disabled, the users assigned the role can no longer run cmdlets granted by the role.

```
Set-ManagementRoleAssignment "Mail Recipients_Denver Help
Desk" -Enabled $false
```

EXAMPLE 2

This example changes the recipient scope for the MyGAL_KimA role assignment to MyGAL. When the recipient scope is changed to a predefined value, any previously defined OUs or custom scopes are overwritten.

```
Set-ManagementRoleAssignment "MyGAL_KimA" -
RecipientRelativeWriteScope MyGAL
```

EXAMPLE 3

This example restricts the Mail Recipients_Marketing Admins role assignment to the contoso.com/North America/Marketing/Users OU. Users who are members of the Marketing Admins role group assigned the role assignment can create, modify, and remove objects only in the specified OU. When the *RecipientOrganizationalUnitScope* parameter is used, any predefined or custom scopes on the role assignment are overwritten.

```
Set-ManagementRoleAssignment "Mail Recipients_Marketing
Admins" -RecipientOrganizationalUnitScope "contoso.com/
North America/Marketing/Users"
```

EXAMPLE 4

This example restricts the Distribution Groups_Cairns Admins role assignment using the Cairns Recipients custom recipient management scope. Users that are members of the Cairns Admins role group assigned the role assignment can create, modify, and remove only the distribution group objects that match the Cairns Recipients custom recipient management scope.

```
Set-ManagementRoleAssignment "Distribution Groups_Cairns
Admins" -CustomRecipientWriteScope "Cairns Recipients"
```

Detailed Description

When you modify a role assignment, you can specify a new predefined or custom management scope or provide an organizational unit (OU) to scope the existing role assignment.

You can create custom management scopes using the **New-ManagementScope** cmdlet and can view a list of existing scopes using the **Get-ManagementScope** cmdlet. If you choose not to specify an OU, predefined scope, or custom scope, the implicit write scope of the role applies to the role assignment.

For more information about management role assignments, see Understanding management role assignments.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role assignments" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.RoleAssignmentIdParameter	specifies the name of the management role assignment to modify. If the name of the management role contains spaces, enclose it in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CustomConfigWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	This parameter is available only in on-premises Exchange 2013. The *CustomConfigWriteScope* parameter specifies the existing configuration management scope to associate with this management role assignment. If the management scope name contains spaces,

			enclose it in quotation marks ("). If you use the *CustomConfigWriteScope* parameter, you can't use the *ExclusiveConfigWriteScope* parameter. To remove a scope, specify a value of `$null`.
CustomRecipientWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	The *CustomRecipientWriteScope* parameter specifies the existing recipient-based management scope to associate with this management role assignment. If the management scope name contains spaces, enclose it in quotation marks ("). If you use the *CustomRecipientWriteScope* parameter, you can't use the *RecipientOrganizationalUnitScope*, *RecipientRelativeWriteScope*, or *ExclusiveRecipientWriteScope* parameters, and

			any configured OU or predefined scope on the role assignment is overwritten. To remove a scope, specify a value of `$null`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the management role assignment is enabled or disabled. The valid values are `$true` and `$false`.
ExclusiveConfigWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	This parameter is available only in on-premises Exchange 2013. The *ExclusiveConfigWriteSco*

			pe parameter specifies the existing configuration exclusive management scope to associate with this management role assignment. If the management scope name contains spaces, enclose it in quotation marks ("). If you use the *ExclusiveConfigWriteScope* parameter, you can't use the *CustomConfigWriteScope* parameter. To remove a scope, specify a value of `$null`.
ExclusiveRecipientWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	The *ExclusiveRecipientWriteScope* parameter specifies the existing recipient-based exclusive management scope to associate with this management role assignment. If the management scope name contains spaces, enclose it in quotation marks (").

			If you use the *ExclusiveRecipientWriteScope* parameter, you can't use the *CustomRecipientWriteScope*, *RecipientOrganizationalUnitScope*, or *RecipientRelativeWriteScope* parameters, and any configured OU or predefined scope on the role assignment is overwritten. To remove a scope, specify a value of `$null`.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to

			specify a value with this parameter.
RecipientOrganizationalUnitScope	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientOrganizationalUnitScope* parameter specifies the OU to scope the new role assignment to. If the OU name contains spaces, enclose the domain and OU in quotation marks ("). If you use the *RecipientOrganizationalUnitScope* parameter, you can't use the *CustomRecipientWriteScope*, *ExclusiveRecipientWriteScope*, or *RecipientRelativeWriteScope* parameters, and any predefined scopes or custom scopes on the role assignment are overwritten. To specify an OU, use the syntax: *domain/ou*. To remove an OU, specify a value of `$null`.
RecipientRelativeWrite	Optional	Microsoft.Exchange.Dat	The *RecipientRelativeWriteS*

Scope		a.Directory.SystemConfiguration.RecipientWriteScopeType	*cope* parameter specifies the type of restriction to apply to a recipient scope. If you use the *RecipientRelativeWriteScope* parameter, you can't use the *CustomRecipientWriteScope*, *ExclusiveRecipientWriteScope*, or *RecipientOrganizationalUnitScope* parameters. The available types are: `None`, `Organization`, `MyGAL`, `Self`, and `MyDistributionGroups`. If you specify a predefined scope, any custom scope or configured OU on the role assignment is overwritten. **Note:** Even though the `NotApplicable`, `OU`, `MyDirectReports`, `CustomRecipientScope`, `MyExecutive`, `MailboxICanDelegate`, and `ExclusiveRecipientScope` values appear in the syntax block for this parameter, they can't be

			used directly on the command line. They're used internally by the cmdlet.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-ManagementRoleEntry

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Add-ManagementRoleEntry** cmdlet to add management role entries to an existing management role.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-ManagementRoleEntry -Identity <RoleEntryIdParameter> [-Parameters
<String[]>] [-PSSnapinName <String>] [-SkipScriptExistenceCheck
<SwitchParameter>] [-Type <Cmdlet | Script | ApplicationPermission |
WebService | All>] [-UnScopedTopLevel <SwitchParameter>] <COMMON
PARAMETERS>


Add-ManagementRoleEntry -ParentRoleEntry <RoleEntryIdParameter> -Role
<RoleIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Force <SwitchParameter>] [-Overwrite <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds a new role entry for the **Get-Mailbox** cmdlet to the Recipient Administrators management role. The role entry for the **Get-Mailbox** cmdlet is added exactly as it's configured in the Recipient Administrators parent role.

```
Add-ManagementRoleEntry "Recipient Administrators\Get-
Mailbox"
```

EXAMPLE 2

This example adds a new role entry for the **Get-Mailbox** cmdlet to the Recipient Administrators role. Only the *Identity*, *Anr*, *Server*, and *Filter* parameters are added to the new role entry.

```
Add-ManagementRoleEntry "Recipient Administrators\Get-
Mailbox" -Parameters Identity, Anr, Server, Filter
```

EXAMPLE 3

This example uses the **Get-ManagementRoleEntry** cmdlet to retrieve a list of all the role entries that exist on the Mail Recipients management role that contain the string "Mailbox" in the cmdlet name, and then adds them to the Mailbox Administrators role using the **Add-ManagementRoleEntry** cmdlet. The role entries are added to the child role exactly as they're configured on the parent role.

```
Get-ManagementRoleEntry "Mail Recipients\*Mailbox*" | Add-
ManagementRoleEntry -Role "Mailbox Administrators"
```

EXAMPLE 4

This example adds the `MailboxAudit` script with the *Department* and *Location* parameters to the IT Scripts unscoped top-level role.

```
Add-ManagementRoleEntry "IT Scripts\MailboxAudit" -
Parameters Department, Location -UnScopedTopLevel
```

Detailed Description

The cmdlet and its parameters that you add to a role entry must exist in the parent role. You can't add role entries to built-in roles.

Important:
You can only add a role entry to a management role if the role entry exists in the role's parent role. For example, if you try to add the **Search-Mailbox** role entry to a role that's a child of the Mail Recipients role, you'll receive an error. This error occurs because the **Search-Mailbox** role entry doesn't exist in the Mail Recipients role. To add the **Search-Mailbox** role entry to a role, you need to create a role that's a child of the Mailbox Import Export role, which contains the **Search-Mailbox** role entry. Then you can use the **Add-ManagementRoleEntry** cmdlet to add the **Search-Mailbox** role entry to the new child role.

For more information about management role entries, see Understanding management roles.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Management role entries" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleEntryIdParameter	The *Identity* parameter specifies the role entry to add. You must specify the value of the *Identity* parameter in the format: *<management role>**<role entry name>*, for example, *ExampleRole**Set-*

			Mailbox. For more information about how management role entries work, see Understanding management roles. The role entry you want to add must exist in the parent role. If the role entry name contains spaces, you must enclose the name in quotation marks (").
ParentRoleEntry	Required	Microsoft.Exchange.Configuration.Tasks.RoleEntryIdParameter	The *ParentRoleEntry* parameter specifies the role entry in the parent role to add to the role specified with the *Role* parameter. This parameter generally isn't used directly, but exists to enable the piping of role entries from the **Get-ManagementRoleEntry** cmdlet. If you use the *ParentRoleEntry* parameter, you can't use the *UnScopedTopLevel* switch.

Role	Required	Microsoft.Exchange.Configuration.Tasks.RoleIdParameter	The *Role* parameter specifies the role to which the new role entry, specified by the *ParentRoleEntry* parameter, is added.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchPara	This parameter is reserved for internal

		meter	Microsoft use.
Overwrite	Optional	System.Management.Automation.SwitchParameter	The *Overwrite* parameter causes existing role entries to be overwritten by the role entries being added.
Parameters	Optional	System.String[]	The *Parameters* parameter specifies the parameters to be included in the role being added. The parameters specified must exist on the cmdlet associated with the role entry. You can specify multiple parameters, separated with commas.
PSSnapinName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *PSSnapinName* parameter specifies the Windows PowerShell snap-in that contains the cmdlet associated with the role being added. Use the **Get-PSSnapin** cmdlet to

			retrieve a list of available Windows PowerShell snap-ins.
SkipScriptExistenceCheck	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Type	Optional	Microsoft.Exchange.Data.Directory.Management.ManagementRoleEntryType	The *Type* parameter specifies the type of role entry being added. The valid values are `Cmdlet`, `Script`, and `ApplicationPermission`.
UnScopedTopLevel	Optional	System.Management.Automation.SwitchParameter	The *UnScopedTopLevel* switch specifies that you're adding a custom script or non-Exchange cmdlet to an unscoped top-level management role. You can only use the *UnScopedTopLevel* switch when you add a role entry to an unscoped top-level role. If you use the *UnScopedTopLevel* switch, you can't use the *ParentRoleEntry* parameter.
WhatIf	Optional	System.Management.Automation.SwitchPara	The *WhatIf* switch instructs the command

		meter	to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ManagementRoleEntry

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ManagementRoleEntry** cmdlet to retrieve management role entries that have been configured on management roles.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ManagementRoleEntry -Identity <RoleEntryIdParameter> [-
DomainController <Fqdn>] [-Parameters <String[]>] [-PSSnapinName <String>]
```

```
[-Type <ManagementRoleEntryType[]>]
```

Examples

EXAMPLE 1

This example retrieves a list of all the role entries that exist on the Transport Rules management role.

```
Get-ManagementRoleEntry "Transport Rules\*"
```

EXAMPLE 2

This example retrieves a list of all the role entries that contain the **Get-Recipient** cmdlet.

```
Get-ManagementRoleEntry *\Get-Recipient
```

EXAMPLE 3

This example retrieves the Tier 2 Help Desk\Set-Mailbox role entry and pipes the output of the **Get-ManagementRoleEntry** cmdlet to the **Format-List** cmdlet. The **Format-List** cmdlet then outputs only the *Name*, *Parameters*, *Role*, and *Type* properties from the role entry.

```
Get-ManagementRoleEntry "Tier 2 Help Desk\Set-Mailbox" |
Format-List Name, Parameters, Role, Type
```

Detailed Description

The **Get-ManagementRoleEntry** cmdlet retrieves role entries that have been configured on roles. You can retrieve specific role entries that match specific criteria such as role name, cmdlet name, parameter name, or a combination of each, or role entry type or the associated Windows PowerShell snap-in.

For more information about management role entries, see Understanding management roles.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Management role entries" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.RoleEntryIdParameter	specifies the role entry to retrieve. You must specify the value of the *Identity* parameter in the format, *<management role>\<role entry name>*, for example, *ExampleRole\Set-Mailbox*. For more information about how management role entries work, see Understanding management roles. You can use the wildcard character (*) instead of the role, cmdlet name, or both. If the role entry name contains spaces, enclose the name in quotation marks (").
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that retrieves data from Active Directory.
Parameters	Optional	System.String[]	The *Parameters* parameter includes only the role entries that contain the parameters specified. You can specify multiple parameters, separated by commas. You can use the wildcard character (*) with partial parameter names to retrieve all parameters that match the value you specify. This parameter is useful when you use the wildcard character (*) with the value you specify in the *Identity* parameter.
PSSnapinName	Optional	System.String	The *PSSnapinName* parameter specifies the Windows PowerShell snap-in that contains the role entry to return. Use the **Get-PSSnapin** cmdlet to retrieve a list of available Windows

			PowerShell snap-ins.
Type	Optional	Microsoft.Exchange.Data.Directory.Management.ManagementRoleEntryType[]	The *Type* parameter specifies the type of role entry to return. The valid values for the *Type* parameter are any combination of the following parameters, separated by commas: `Cmdlet`, `Script`, and `ApplicationPermission`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ManagementRoleEntry

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-ManagementRoleEntry** cmdlet to remove existing management role entries.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ManagementRoleEntry -Identity <RoleEntryIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
```

```
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the **New-Mailbox** role entry from the Tier 1 Help Desk role.

```
Remove-ManagementRoleEntry "Tier 1 Help Desk\New-Mailbox"
```

EXAMPLE 2

This example removes all the role entries that have the verb `New` on the Tier 1 Help Desk role by piping the output of the **Get-ManagementRoleEntry** cmdlet to the **Remove-ManagementRoleEntry** cmdlet. Because the *WhatIf* switch has been specified along with the **Remove-ManagementRoleEntry** cmdlet, the cmdlet lists what changes would have been made but doesn't commit any changes.

```
Get-ManagementRoleEntry "Tier 1 Help Desk\New-*" | Remove-ManagementRoleEntry -WhatIf
```

After you verify that the correct role entries will be removed, run the same command without the *WhatIf* switch to remove the role entries.

```
Get-ManagementRoleEntry "Tier 1 Help Desk\New-*" | Remove-ManagementRoleEntry
```

Detailed Description

The **Remove-ManagementRoleEntry** cmdlet removes existing role entries. However, you can't remove role entries from built-in management roles.

For more information about management role entries, see Understanding management roles.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Management role entries" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleE	The *Identity* parameter specifies the role entry

		ntryIdParameter	to remove. You must specify the value of the *Identity* parameter in the format, *<management role>\<role entry name>*, for example, *ExampleRole\Set-Mailbox*. For more information about how management role entries work, see Understanding management roles. If the role entry name contains spaces, enclose the name in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Dat	This parameter is available only in on-

		a.Fqdn	premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on

			the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ManagementRoleEntry

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-ManagementRoleEntry** cmdlet to change the available parameters on an existing management role entry.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ManagementRoleEntry -Identity <RoleEntryIdParameter> [-AddParameter
<SwitchParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Force <SwitchParameter>] [-Parameters <String[]>] [-
RemoveParameter <SwitchParameter>] [-SkipScriptExistenceCheck
<SwitchParameter>] [-UnScopedTopLevel <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the *Anr* and *Database* parameters from the **Get-Mailbox** role entry on the Help Desk Personnel role.

```
Set-ManagementRoleEntry "Help Desk Personnel\Get-Mailbox" -
Parameters Anr, Database -RemoveParameter
```

EXAMPLE 2

This example retrieves a list of role entries on the Help Desk Personnel role and adds the *WhatIf* switch to each role entry using the **Set-ManagementRoleEntry** cmdlet.

```
Get-ManagementRoleEntry "Help Desk Personnel\*" | Set-
ManagementRoleEntry -Parameters WhatIf -AddParameter
```

EXAMPLE 3

This example adds the *DisplayName* and *ForwardingAddress* parameters to the **Set-Mailbox** role entry on the Tier 1 Help Desk role and removes all other parameters from the role entry.

```
Set-ManagementRoleEntry "Tier 1 Help Desk\Set-Mailbox" -
Parameters DisplayName, ForwardingAddress
```

EXAMPLE 4

This example adds the *Location* parameter to the `MailboxAudit` custom script on the IT Scripts unscoped top level role.

```
Set-ManagementRoleEntry "IT Scripts\MailboxAudit" -
Parameters Location -AddParameter -UnScopedTopLevel
```

Detailed Description

The **Set-ManagementRoleEntry** cmdlet changes the available parameters on an existing role entry. If you want to add parameters to a role entry, the parameters must exist in the role entry in the parent management role. If you want to remove parameters from a role entry, there can be no role entries in child roles that inherit those parameters from the role entry you want to change. You can't change role entries associated with built-in roles.

For more information about management role entries, see Understanding management roles.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Management role entries" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleEntryIdParameter	The *Identity* parameter specifies the role entry to change. You must specify the value of the *Identity* parameter in the format, *<management role>**<role entry name>*, for example, ExampleRole\Set-Mailbox. For more information about how management role entries work, see Understanding management roles. If the role entry name contains spaces, enclose it in quotation marks (").
AddParameter	Optional	System.Management.Automation.SwitchParameter	The *AddParameter* parameter adds the specified parameters to the specified role entry. Use the *Parameters* parameter to specify the parameters to add. You can't use the *AddParameter* parameter

			in the same command as the *RemoveParameter* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for

			administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
Parameters	Optional	System.String[]	The *Parameters* parameter specifies the parameters to be added to or removed from the role entry. The *Parameters* parameter has the following modes: • When used with the *AddParameter* parameter, the parameters you specify are added to the role entry. • When used with the *RemoveParameter* parameter, the parameters you specify are removed from the role entry. • When neither the *AddParameter* nor *RemoveParameter* parameters are used,

			only the parameters you specify are included in the role entry. If you specify a value of `$Null` and neither the *AddParameter* nor *RemoveParameter* parameters are used, all of the parameters on the role entry are removed. You can specify multiple parameters, separated with commas.
RemoveParameter	Optional	System.Management.Automation.SwitchParameter	The *RemoveParameter* parameter removes the specified parameters from the specified role entry. Use the *Parameters* parameter to specify the parameters to remove. You can't use the *RemoveParameter* parameter in the same command as the *AddParameter* parameter.
SkipScriptExistenceCheck	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
UnScopedTopLevel	Optional	System.Management.Automation.SwitchPar	The *UnScopedTopLevel* switch must be used when

		ameter	you want to modify a role entry on an unscoped top level role.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ManagementScope

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ManagementScope** cmdlet to return a list of management scopes.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ManagementScope [-Identity <ManagementScopeIdParameter>] [-
DomainController <Fqdn>] [-Exclusive <$true | $false>] [-Organization
<OrganizationIdParameter>] [-Orphan <SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves all the management scopes that start with the string Redmond.

```
Get-ManagementScope Redmond*
```

EXAMPLE 2

This example retrieves the Redmond Servers Scope using the **Get-ManagementScope** cmdlet and pipes the output to the **Format-List** cmdlet. For more information about the **Format-List** cmdlet, see Working with command output.

```
Get-ManagementScope "Redmond Servers Scope" | Format-List
```

EXAMPLE 3

This example retrieves a list of management scopes that aren't associated with any role assignments.

```
Get-ManagementScope -Orphan
```

EXAMPLE 4

This example retrieves a list of exclusive scopes.

```
Get-ManagementScope -Exclusive $True
```

Detailed Description

You can retrieve one scope or many, retrieve only scopes that aren't associated with management role assignments, or retrieve scopes that are exclusive or regular scopes.

For more information about regular and exclusive scopes, see Understanding management role scopes.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the

"Management scopes" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Exclusive	Optional	System.Boolean	The *Exclusive* parameter specifies whether exclusive scopes should be returned. If the *Exclusive* parameter isn't specified, regular scopes and exclusive scopes are returned. If the *Exclusive* parameter is set to `$True`, only exclusive scopes are returned. If the *Exclusive* parameter is set to `$False`, only regular scopes are returned. The valid

			values are `$True` and `$False`.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	The *Identity* parameter specifies the name of the management scope to return. If the management scope name contains spaces, enclose it in quotation marks (").
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Orphan	Optional	System.Management.Automation.SwitchParameter	The *Orphan* parameter returns only the management scopes that aren't associated with role assignments.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ManagementScope

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-ManagementScope** cmdlet to create a regular or exclusive management scope.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-ManagementScope -RecipientRestrictionFilter <String> [-Exclusive
<SwitchParameter>] [-Force <SwitchParameter>] [-RecipientRoot
<OrganizationalUnitIdParameter>] <COMMON PARAMETERS>


New-ManagementScope -ServerRestrictionFilter <String> [-Exclusive
<SwitchParameter>] [-Force <SwitchParameter>] <COMMON PARAMETERS>


New-ManagementScope -ServerList <ServerIdParameter[]> [-Exclusive
<SwitchParameter>] [-Force <SwitchParameter>] <COMMON PARAMETERS>


New-ManagementScope -DatabaseList <DatabaseIdParameter[]> [-Exclusive
<SwitchParameter>] [-Force <SwitchParameter>] <COMMON PARAMETERS>


New-ManagementScope -DatabaseRestrictionFilter <String> [-Exclusive
<SwitchParameter>] [-Force <SwitchParameter>] <COMMON PARAMETERS>


New-ManagementScope -PartnerDelegatedTenantRestrictionFilter <String>
<COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a scope that includes only the servers MailboxServer1, MailboxServer2, and MailboxServer3. Users assigned roles using management role assignments that have the scope in this example can only perform against the servers included in the scope.

```
New-ManagementScope -Name "Mailbox Servers 1 through 3" -
ServerList MailboxServer1, MailboxServer2, MailboxServer3
```

EXAMPLE 2

This example creates the Redmond Site Scope scope and sets a server restriction filter that matches only the servers located in the "CN=Redmond,CN=Sites,CN=Configuration,DC=contoso,DC=com" Active Directory Domain Services (AD DS) site.

```
New-ManagementScope -Name "Redmond Site Scope" -
ServerRestrictionFilter {ServerSite -eq
"CN=Redmond,CN=Sites,CN=Configuration,DC=contoso,DC=com"}
```

EXAMPLE 3

This example creates the Executive Mailboxes scope. Only mailboxes located within the Executives OU in the contoso.com domain match the recipient restriction filter.

```
New-ManagementScope -Name "Executive Mailboxes" -
RecipientRoot "contoso.com/Executives" -
RecipientRestrictionFilter {RecipientType -eq
"UserMailbox"}
```

EXAMPLE 4

This example creates the Protected Exec Users exclusive scope. Users that contain the string "VP" in their title match the recipient filter for the scope. When the exclusive scope is created, all users are immediately blocked from modifying the recipients that match the exclusive scope until the scope is associated with a management role assignment. If other role assignments are associated with other exclusive scopes that match the same recipients, those assignments can still modify the recipients.

```
New-ManagementScope -Name "Protected Exec Users" -
RecipientRestrictionFilter { Title -Like "*VP*" } -
Exclusive
```

The exclusive scope is then associated with a management role assignment that assigns the Mail Recipients management role to the Executive Administrators role group. This role group contains administrators who are allowed to modify the mailboxes of high-profile executives. Only the administrators of the Executive Administrators role group can modify users with the string "VP" in their title.

```
New-ManagementRoleAssignment -SecurityGroup "Executive
Administrators" -Role "Mail Recipients" -
CustomRecipientWriteScope "Protected Exec Users"
```

EXAMPLE 5

This example creates the Seattle Databases scope and sets a database restriction filter that matches only the databases that begin with the string "SEA".

```
New-ManagementScope -Name "Seattle Databases" -
DatabaseRestrictionFilter {Name -Like "SEA*" }
```

Detailed Description

After you create a regular or exclusive scope, you need to associate the scope with a management role assignment. To associate a scope with a role assignment, use the New-ManagementRoleAssignment cmdlet. For more information about adding new management scopes, see Create a regular or exclusive scope.

Important:
Database scopes are only enforced on servers running Microsoft Exchange Server 2010 Service Pack 1 (SP1) and Exchange Server 2013. If a user connects to a server running the release to manufacturing (RTM) version of Exchange 2010, management role assignments associated with database scopes aren't applied. Also, database scopes aren't visible to the **Get-ManagementScope** cmdlet when it's run on an Exchange 2010 RTM server.

For more information about regular and exclusive scopes, see Understanding management role scopes.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Management scopes" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
DatabaseList	Required	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter[]	This parameter is available only in on-premises Exchange 2013. The *DatabaseList* parameter specifies a list of databases to which the scope should be applied. Multiple databases can be specified, separated by commas. If you use the *DatabaseList* parameter, you can't use the *DatabaseRestrictionFilter*, *ServerList*,

			RecipientRestrictionFilter, *RecipientRoot*, or *ServerRestrictionFilter* parameters.
DatabaseRestrictionFilter	Required	System.String	This parameter is available only in on-premises Exchange 2013. The *DatabaseRestrictionFilter* parameter specifies the filter to apply to database objects. Only database objects that match the filter are included in the scope. If you use the *DatabaseRestrictionFilter* parameter, you can't use the *RecipientRestrictionFilter*, *ServerRestrictionFilter*, *RecipientRoot*, *DatabaseList*, or *ServerList* parameters. For a list of filterable database properties, see Understanding management role scopes.
Name	Required	System.String	The *Name* parameter specifies the name of the management scope. The name can be up to 64 characters. If the name

			contains spaces, enclose the name in quotation marks (").
PartnerDelegatedTenantRestrictionFilter	Required	System.String	This parameter is reserved for internal Microsoft use.
RecipientRestrictionFilter	Required	System.String	The *RecipientRestrictionFilter* parameter specifies the filter to apply to recipient objects. Only recipient objects that match the filter are included in the scope. If you use the *RecipientRestrictionFilter* parameter, you can't use the *DatabaseRestrictionFilter*, *DatabaseList*, *ServerList*, or *ServerRestrictionFilter* parameters.
ServerList	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter[]	This parameter is available only in on-premises Exchange 2013. The *ServerList* parameter specifies a list of servers to which the scope should be applied. Multiple servers can be specified, separated by commas. If you use the

			ServerList parameter, you can't use the *RecipientRestrictionFilter*, *RecipientRoot*, *DatabaseRestrictionFilter*, *DatabaseList*, or *ServerRestrictionFilter* parameters.
ServerRestrictionFilter	Required	System.String	This parameter is available only in on-premises Exchange 2013. The *ServerRestrictionFilter* parameter specifies the filter to apply to server objects. Only server objects that match the filter are included in the scope. If you use the *ServerRestrictionFilter* parameter, you can't use the *RecipientRestrictionFilter*, *RecipientRoot*, *DatabaseRestrictionFilter*, *DatabaseList*, or *ServerList* parameters. For a list of filterable server properties, see Understanding management role scopes.
Confirm	Optional	System.Management.	The *Confirm* switch

		Automation.SwitchParameter	causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Exclusive	Optional	System.Management.Automation.SwitchParameter	The *Exclusive* switch specifies that the role should be an exclusive scope. **Caution:** When you create exclusive management scopes, only users or universal security groups (USG) assigned exclusive scopes that contain objects to be modified can access those objects. Users or USGs that aren't assigned an exclusive

			scope that contains the objects immediately lose access to those objects.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies that an exclusive scope should be created without showing the warning that the exclusive scope takes effect immediately.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
RecipientRoot	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientRoot* parameter specifies the organizational unit (OU) under which the filter specified with the *RecipientRestrictionFilter* parameter should be applied. If you use the *RecipientRoot* parameter, you can't use the *DatabaseRestrictionFilter*, *DatabaseList*, *ServerList*, or *ServerRestrictionFilter* parameters.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the

			WhatIf switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ManagementScope

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-ManagementScope** cmdlet to remove an existing management scope.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ManagementScope -Identity <ManagementScopeIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the Redmond Servers scope.

```
Remove-ManagementScope "Redmond Servers"
```

EXAMPLE 2

This example retrieves a list of all the orphaned scopes using the **Get-ManagementScope** cmdlet and pipes the output to the **Remove-ManagementScope** cmdlet. Because the *WhatIf* switch is used with the **Remove-ManagementScope** cmdlet, the cmdlet only displays the scopes that would have been removed but doesn't commit any changes.

```
Get-ManagementScope -Orphan | Remove-ManagementScope -
WhatIf
```

After you verify that the scopes to be removed are correct, run the command again without the *WhatIf* switch.

```
Get-ManagementScope -Orphan | Remove-ManagementScope
```

Detailed Description

You can't remove a management scope if it's associated with a management role assignment. Use the **Get-ManagementScope** cmdlet to retrieve a list of orphaned scopes. For more information about regular and exclusive scopes, see Understanding management role scopes.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Management scopes" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	The *Identity* parameter specifies the scope to remove. You can't remove a scope if it's in use by a management role assignment.
Confirm	Optional	System.Management.Automation.SwitchPara	The *Confirm* switch causes the command to

		meter	pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't

			provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ManagementScope

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-ManagementScope** cmdlet to change an existing management scope.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ManagementScope [-RecipientRestrictionFilter <String>] [-RecipientRoot
<OrganizationalUnitIdParameter>] <COMMON PARAMETERS>


Set-ManagementScope -ServerRestrictionFilter <String> <COMMON PARAMETERS>


Set-ManagementScope -DatabaseRestrictionFilter <String> <COMMON
PARAMETERS>


Set-ManagementScope -PartnerDelegatedTenantRestrictionFilter <String>
<COMMON PARAMETERS>


COMMON PARAMETERS: -Identity <ManagementScopeIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-Name <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the recipient restriction filter on the Seattle Mailboxes management scope to match all mailboxes that have Seattle in the **City** mailbox property.

```
Set-ManagementScope "Seattle Mailboxes" -
RecipientRestrictionFilter { City -Eq "Seattle" -And
RecipientType -Eq "UserMailbox" }
```

EXAMPLE 2

This example changes the recipient root for the Sales Recipients management scope to match only recipient objects contained under the contoso.com/Sales OU.

```
Set-ManagementScope "Sales Recipients" -RecipientRoot
contoso.com/Sales
```

EXAMPLE 3

This example changes the Active Directory Domain Services (AD DS) site used in the server restriction filter for the Vancouver Servers management scope to "NA-CDN-

Vancouver,CN=Sites,CN=Configuration,DC=contoso,DC=com".

```
Set-ManagementScope "Vancouver Servers" -
ServerRestrictionFilter { ServerSite -Eq "NA-CDN-
Vancouver,CN=Sites,CN=Configuration,DC=contoso,DC=com" }
```

Detailed Description

If you change a scope that has been associated with management role assignments using the **New-ManagementRoleAssignment** cmdlet, the updated scope applies to all the associated role assignments. For more information about changing scopes, see Change a role scope.

Important:
Database scopes are only enforced on servers running Microsoft Exchange Server 2010 Service Pack 1 and Exchange Server 2013. If a user connects to a server running the release to manufacturing (RTM) version of Exchange 2010, management role assignments associated with database scopes won't be applied. Also, database scopes won't be visible to the **Get-ManagementScope** cmdlet when it's run on an Exchange 2010 RTM server.

For more information about regular and exclusive scopes, see Understanding management role scopes.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Management scopes" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
DatabaseRestrictionFilter	Required	System.String	This parameter is available only in on-premises Exchange 2013. The *DatabaseRestrictionFilter* parameter specifies the filter to apply to database objects. When the

			DatabaseRestrictionFilter parameter is specified, only database objects that match the filter are included in the scope. If you use the *DatabaseRestrictionFilter* parameter, you can't use the *ServerRestrictionFilter*, *RecipientRestrictionFilter* or *RecipientRoot* parameters. For a list of filterable database properties, see Understanding management role scopes.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	The *Identity* parameter specifies the name of the management scope to modify. If the name contains spaces, enclose it in quotation marks (").
PartnerDelegatedTenantRestrictionFilter	Required	System.String	This parameter is reserved for internal Microsoft use.
ServerRestrictionFilter	Required	System.String	This parameter is available only in on-premises Exchange 2013.

			The *ServerRestrictionFilter* parameter specifies the filter to apply to server objects. When the *ServerRestrictionFilter* parameter is specified, only recipient objects that match the filter are included in the scope. If you use the *ServerRestrictionFilter* parameter, you can't use the *DatabaseRestrictionFilter*, *RecipientRestrictionFilter*, or *RecipientRoot* parameters. For a list of filterable server properties, see Understanding management role scopes.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't

			have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this

			parameter.
Name	Optional	System.String	The *Name* parameter specifies the name of the management scope. The management scope name can be a maximum of 64 characters. If the name contains spaces, enclose it in quotation marks (").
RecipientRestrictionFilter	Optional	System.String	The *RecipientRestrictionFilter* parameter specifies the filter to apply to recipient objects. When the *RecipientRestrictionFilter* parameter is specified, only server objects that match the filter are included in the scope. If you use the *RecipientRestrictionFilter* parameter, you can't use the *DatabaseRestrictionFilter* or *ServerRestrictionFilter* parameters.

RecipientRoot	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientRoot* parameter specifies the organizational unit (OU) under which the filter specified with the *RecipientRestrictionFilter* parameter should be applied. If you use the *RecipientRoot* parameter, you can't use the *ServerRestrictionFilter* or *DatabaseRestrictionFilter* parameters.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-RoleAssignmentPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-RoleAssignmentPolicy** cmdlet to view an existing management role assignment policy on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-RoleAssignmentPolicy [-Identity <MailboxPolicyIdParameter>] [-
DomainController <Fqdn>] [-IgnoreDehydratedFlag <SwitchParameter>] [-
Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example returns a list of all the existing role assignment policies.

```
Get-RoleAssignmentPolicy
```

EXAMPLE 2

This example returns the details of the specified assignment policy. The output of the **Get-RoleAssignmentPolicy** cmdlet is piped to the **Format-List** cmdlet.

```
Get-RoleAssignmentPolicy "End User Policy" | Format-List
```

For more information about pipelining and the **Format-List** cmdlet, see Pipelining and Working with command output.

EXAMPLE 3

This example returns the default assignment policy.

The output of the **Get-RoleAssignmentPolicy** cmdlet is piped to the **Where** cmdlet. The **Where** cmdlet filters out all of the policies except the policy that has the *IsDefault* property set to `$True`.

```
Get-RoleAssignmentPolicy | Where { $_.IsDefault -eq $True }
```

For more information about pipelining and the **Format-List** cmdlet, see Pipelining and Working with command output.

Detailed Description

For more information about assignment policies, see Understanding management role assignment policies.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Assignment policies" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server

			uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the name of the assignment policy to view. If the name contains spaces, enclose the name in quotation marks (").
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-RoleAssignmentPolicy

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-RoleAssignmentPolicy** cmdlet to create a management role assignment policy on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-RoleAssignmentPolicy -Name <String> [-Confirm [<SwitchParameter>]] [-
Description <String>] [-DomainController <Fqdn>] [-IgnoreDehydratedFlag
<SwitchParameter>] [-IsDefault <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-Roles <RoleIdParameter[]>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates an assignment policy.

```
New-RoleAssignmentPolicy -Name "End User Policy"
```

After the assignment policy is created, you can assign the assignment policy to a mailbox using the **Set-Mailbox** cmdlet.

```
Set-Mailbox Joe -RoleAssignmentPolicy "End User Policy"
```

EXAMPLE 2

This example creates an assignment policy using the *IsDefault* switch.

```
New-RoleAssignmentPolicy -Name "Default End User Policy" -
IsDefault
```

EXAMPLE 3

This example creates an assignment policy that enables users to modify their personal information, manage their distribution group membership, and manage their voice mail. The new assignment policy is created as the new default assignment policy. Then, all existing mailboxes are configured to use the new assignment policy.

First, the new assignment policy is created and set as the new default assignment policy.

```
New-RoleAssignmentPolicy -Name "Limited End User Policy" -
Roles "MyPersonalInformation",
```

```
"MyDistributionGroupMembership", "MyVoiceMail" -IsDefault
```

Because setting the new role assignment as default applies only to new mailboxes or mailboxes moved to an Exchange 2013 server, the **Set-Mailbox** cmdlet is used to configure the new assignment policy on all existing mailboxes.

```
Get-Mailbox -ResultSize Unlimited | Set-Mailbox -
RoleAssignmentPolicy "Limited End User Policy"
```

Detailed Description

When you create an assignment policy, you can assign it to users using the **New-Mailbox**, **Set-Mailbox**, or **Enable-Mailbox** cmdlets. If you make the new assignment policy the default assignment policy, it's assigned to all new mailboxes that don't have an explicit assignment policy assigned to them.

You can add management roles to the new assignment policy when you create it, or you can create the assignment policy and add roles later. You must assign at least one management role to the new assignment policy for it to apply permissions to a mailbox. Without any roles assigned to the new assignment policy, users assigned to it won't be able to manage their mailbox configuration. To assign a management role after the assignment policy has been created, use the **New-ManagementRoleAssignment** cmdlet. For more information, see Manage role assignment policies.

For more information about assignment policies, see Understanding management role assignment policies.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Assignment policies" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name of the new assignment policy. If the assignment policy name contains spaces, enclose the name in quotation

			marks ("). The maximum length of the name is 64 characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies the description that's displayed when the role assignment policy is viewed using the **Get-RoleAssignmentPolicy** cmdlet. Enclose the description in quotation marks (").
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IsDefault	Optional	System.Management.Automation.SwitchParameter	The *IsDefault* switch specifies whether the new assignment policy should become the default assignment policy. New mailboxes or mailboxes moved to an Exchange 2013 server are assigned the default assignment policy when an explicit assignment policy isn't provided. You don't have to specify a value with this switch.

			Note: Setting an assignment policy as default doesn't change the role assignment on existing mailboxes. To change the assignment policies on existing mailboxes, use the **Set-Mailbox** cmdlet.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Roles	Optional	Microsoft.Exchange.Configuration.Tasks.RoleIdParameter[]	The *Roles* parameter specifies the management roles to assign to the role assignment policy when it's created. If a role name contains spaces, enclose the name in quotation marks ("). If you want to assign more that one role, separate the role names with commas. For a list of built-in management roles that you can assign to a role group, see Built-in management roles.
WhatIf	Optional	System.Management.Automation.SwitchPara	The *WhatIf* switch instructs the command

		meter	to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-RoleAssignmentPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-RoleAssignmentPolicy** cmdlet to remove an existing management role assignment policy from a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-RoleAssignmentPolicy -Identity <MailboxPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
```

```
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the assignment policy End User Policy. First, find all the mailboxes assigned the assignment policy.

```
Get-Mailbox | Where {$_.RoleAssignmentPolicy -Eq "End User Policy"}
```

Next, use the list you gathered to assign each mailbox a new assignment policy. You may want to assign all the mailboxes the same new assignment policy, such as Seattle End User Policy.

```
Get-Mailbox | Where {$_.RoleAssignmentPolicy -Eq "End User Policy"} | Set-Mailbox -RoleAssignmentPolicy "Seattle End User Policy"
```

Then, remove all the management role assignments assigned to the previous assignment policy End User Policy.

```
Get-ManagementRoleAssignment -RoleAssignee "End User Policy" | Remove-ManagementRoleAssignment
```

Finally, remove the assignment policy.

```
Remove-RoleAssignmentPolicy "End User Policy"
```

For more information about the **Where** cmdlet and pipelining, see Working with command output and Pipelining.

Detailed Description

The assignment policy you want to remove can't be assigned to any mailboxes or management roles. Also, if you want to remove the default assignment policy, it must be the last assignment policy. Do the following before you attempt to remove an assignment policy:

- Use the **Set-Mailbox** cmdlet to change the assignment policy for any mailbox assigned the assignment policy you want to remove.
- If the assignment policy is the default assignment policy, use the **Set-RoleAssignmentPolicy** cmdlet to select a new default assignment policy. You don't need to do this if you're removing the last assignment policy.
- Use the **Remove-ManagementRoleAssignment** cmdlet to remove any management role assignments assigned to the policy.

For more information about assignment policies, see Understanding management role assignment policies.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Assignment policies" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the assignment policy to remove. If the assignment policy name has spaces, enclose the name in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-RoleAssignmentPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-RoleAssignmentPolicy** cmdlet to modify an existing management role assignment policy on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-RoleAssignmentPolicy -Identity <MailboxPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-Description <String>] [-DomainController <Fqdn>] [-
IsDefault <SwitchParameter>] [-Name <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the default assignment policy. New mailboxes or mailboxes moved to an Exchange 2013 server are assigned the default assignment policy when an explicit assignment policy isn't provided. You don't have to specify a value with the *IsDefault* switch.

```
Set-RoleAssignmentPolicy "End User Policy" -IsDefault
```

Detailed Description

You can use the **Set-RoleAssignmentPolicy** cmdlet to change the name of an assignment policy or

to set the assignment policy as the default assignment policy.

For more information about assignment policies, see Understanding management role assignment policies.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Assignment policies" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the name of the assignment policy to modify. If the name contains spaces, enclose the name in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies the description that's displayed when the

			role assignment policy is viewed using the **Get-RoleAssignmentPolicy** cmdlet. Enclose the description in quotation marks (").
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
IsDefault	Optional	System.Management.Automation.SwitchPara	The *IsDefault* switch makes the assignment

		meter	policy the default assignment policy. New mailboxes or mailboxes moved to an Exchange 2013 server are assigned the default assignment policy when an explicit assignment policy isn't provided. You don't have to specify a value with this switch.
Name	Optional	System.String	The *Name* parameter specifies the new name of the assignment policy. If the assignment policy name contains spaces, enclose the name in quotation marks ("). The maximum length of the name is 64 characters.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of

			those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-RoleGroup

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-RoleGroup** cmdlet to retrieve a list of management role groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-RoleGroup [-Identity <RoleGroupIdParameter>] [-AccountPartition
<AccountPartitionIdParameter>] [-DomainController <Fqdn>] [-Filter
<String>] [-Organization <OrganizationIdParameter>] [-
ReadFromDomainController <SwitchParameter>] [-ResultSize <Unlimited>] [-
ShowPartnerLinked <SwitchParameter>] [-SortBy <String>] [-
UsnForReconciliationSearch <Int64>]
```

Examples

EXAMPLE 1

This example retrieves a list of role groups.

```
Get-RoleGroup
```

EXAMPLE 2

This example retrieves the details for the Recipient Administrators role group.

```
Get-RoleGroup "Recipient Administrators" | Format-List
```

For more information about the **Format-List** cmdlet and pipelining, see Working with command output and Pipelining.

EXAMPLE 3

This example retrieves a list of role groups as seen by the domain controller closest to the user.

```
Get-RoleGroup -ReadFromDomainController
```

EXAMPLE 4

This example retrieves a list of all linked role groups and the Active Directory security identifier (SID) of the foreign universal security groups (USG) that are linked to each of them. You can then use the SIDs to find the USGs so you can modify their members.

```
Get-RoleGroup -Filter { RoleGroupType -Eq "Linked" } |
Format-Table Name, LinkedGroup
```

Detailed Description

For more information about role groups, see Understanding management role groups.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange

			2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter specifies the property to be used to filter the role groups. Only the role groups that match the criteria you specify are returned. You can filter on the `LinkedGroup`, `ManagedBy`, `Members`, `Name`, `RoleGroupType`, and `DisplayName` properties. If you create a filter using the `RoleGroupType` property, the only values you can use in the filter are `Standard` and `Linked`.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.RoleGroupIdParameter	The *Identity* parameter specifies the role group to retrieve. If the name of the role group contains spaces,

			enclose the name in quotation marks ("). If the *Identity* parameter isn't specified, all role groups are returned.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* switch specifies that the role group information should be read from a domain controller in the user's domain. If you use the **Set-AdServerSettings** cmdlet to include scope commands to the entire forest and don't use this switch, it's possible that the role group information is read from a global catalog with outdated

			information.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all the role groups, use `Unlimited` for the value of this parameter. The default value is `1000`.
ShowPartnerLinked	Optional	System.Management.Automation.SwitchParameter	This *ShowPartnerLinked* switch specifies whether to return built-in role groups that are of type `PartnerRoleGroup`. Role groups of this type are used in the cloud-based services to allow partner service providers to manage their customer organizations. These role groups can't be edited and are therefore not shown by default. This parameter is available only in the cloud-based service.

SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. You can sort by only one attribute at a time. You can sort by the **Name** attribute. The results are sorted in ascending order.
UsnForReconciliationSearch	Optional	System.Int64	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-RoleGroup

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-RoleGroup** cmdlet to create a management role group on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-RoleGroup <COMMON PARAMETERS>

New-RoleGroup -LinkedPartnerGroupId <String> -LinkedPartnerOrganizationId
<String> <COMMON PARAMETERS>

New-RoleGroup -LinkedDomainController <String> -LinkedForeignGroup
<UniversalSecurityGroupIdParameter> [-LinkedCredential <PSCredential>]
<COMMON PARAMETERS>

COMMON PARAMETERS: -Name <String> [-Confirm [<SwitchParameter>]] [-
CustomConfigWriteScope <ManagementScopeIdParameter>] [-
CustomRecipientWriteScope <ManagementScopeIdParameter>] [-Description
<String>] [-DisplayName <String>] [-DomainController <Fqdn>] [-
ExternalDirectoryObjectId <String>] [-Force <SwitchParameter>] [-ManagedBy
<MultiValuedProperty>] [-Members <MultiValuedProperty>] [-Organization
<OrganizationIdParameter>] [-PartnerManaged <SwitchParameter>] [-
RecipientOrganizationalUnitScope <OrganizationalUnitIdParameter>] [-Roles
<RoleIdParameter[]>] [-SamAccountName <String>] [-ValidationOrganization
<String>] [-WellKnownObjectGuid <Guid>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a role group. The Mail Recipients and Mail Enabled Public Folders roles are assigned to the role group, and the users Kim and Martin are added as members. Because no scopes were provided, Kim and Martin can manage any recipient and reset passwords for any users in the organization.

```
New-RoleGroup -Name "Limited Recipient Management" -Roles
"Mail Recipients", "Mail Enabled Public Folders" -Members
Kim, Martin
```

EXAMPLE 2

This example creates a role group with a custom recipient scope. The custom recipient scope, Seattle Recipients, limits the scope of the roles assigned to the role group to recipients who have their **City** property set to Seattle. The Mail Recipients and Mail Enabled Public Folders roles are assigned to the role group, and the users John and Carol are added as members.

```
New-RoleGroup -Name "Seattle Limited Recipient Management"
-Roles "Mail Recipients", "Mail Enabled Public Folders" -
Members John, Carol -CustomRecipientWriteScope "Seattle
Recipients"
```

EXAMPLE 3

This example creates a role group and enables Isabel to add or remove members to or from the role group by adding her to the **ManagedBy** property. The Transport Rules role is assigned to the role group, and the Compliance Group USG is added as a member.

```
New-RoleGroup -Name "Transport Rules Management" -Roles
"Transport Rules" -Members "Compliance Group" -ManagedBy
Isabel
```

EXAMPLE 4

This example creates a linked role group that enables the members of the Toronto Administrators USG in the Contoso user forest to manage recipients located in the Toronto office. The custom recipient scope, Toronto Recipients, limits the scope of the roles assigned to the role group to recipients who have their **City** property set to Toronto. The Mail Recipients role is assigned to the role group.

First, the credentials for the user forest need to be stored in a variable so they can be used with the **New-RoleGroup** cmdlet. The following command retrieves the credentials using the **Get-Credential** cmdlet and stores them in the `$Credentials` variable.

```
$Credentials = Get-Credential
```

Then the linked role group is created using the following command.

```
New-RoleGroup -Name "ContosoUsers: Toronto Recipient
Admins" -LinkedDomainController
dc02.contosousers.contoso.com -LinkedCredential
$Credentials -LinkedForeignGroup "Toronto Administrators" -
CustomRecipientWriteScope "Toronto Recipients" -Roles "Mail
Recipients"
```

EXAMPLE 5

This example takes an existing role group and copies the roles from that role group into a new custom role group. This can be useful if you want to create a role group similar to an existing role group but don't want to manually create all the role assignments. For example, you might want to create a role group that has most, but not all, of the management roles assigned to the Recipient Management role group.

First, store the existing role group in a variable using the following command.

```
$RoleGroup = Get-RoleGroup "Recipient Management"
```

Then, create the custom role group using the following command.

```
New-RoleGroup "Limited Recipient Management" -Roles
$RoleGroup.Roles
```

The new Limited Recipient Management role group now has the same roles as Recipient

Management. Remove the role assignments you don't want using the **Remove-ManagementRoleAssignment** cmdlet. For example, you might not want members of the Limited Recipient Management role group to manage distribution groups. Use the following command to remove the role assignment between the Distribution Groups management role and the Limited Recipient Management role group.

```
Remove-ManagementRoleAssignment "Distribution Groups-Limited Recipient Management"
```

This example uses variables to store information. For more information about variables, see User-defined variables.

Detailed Description

You don't have to add members or assign management roles to the role group when you create it. However, until you add members or assign roles to the role group, the role group grants no permissions to users. You can also specify custom configuration or recipient scopes when you create a role group. These scopes are applied to the management role assignments created when the role group is created.

When you create a role group, you can create the group and add members to it directly, or you can create a linked role group. A linked role group links the role group to a universal security group (USG) in another forest. Creating a linked role group is useful if your servers running Exchange reside in a resource forest and your users and administrators reside in a separate user forest. If you create a linked role group, you can't add members directly to it. You must add the members to the USG in the foreign forest.

For more information about role groups, see Understanding management role groups.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
LinkedDomainController	Required	System.String	This parameter is available only in on-premises Exchange 2013. The

			LinkedDomainController parameter specifies the fully qualified domain name (FQDN) or IP address of the domain controller in the forest where the foreign USG resides. The domain controller you specify is used to get security information for the foreign USG specified by the *LinkedForeignGroup* parameter. If you use the *LinkedDomainController* parameter, you must specify a foreign USG with the *LinkedForeignGroup* parameter, and you can't use the *Members* parameter.
LinkedForeignGroup	Required	Microsoft.Exchange.Configuration.Tasks.UniversalSecurityGroupIdParameter	This parameter is available only in on-premises Exchange 2013. The *LinkedForeignGroup* parameter specifies the name of the foreign

			USG you want to link this role group to. If the foreign USG name contains spaces, enclose the name in quotation marks ("). If you use the *LinkedForeignGroup* parameter, you must specify a domain controller in the *LinkedDomainController* parameter, and you can't use the *Members* parameter.
LinkedPartnerGroupId	Required	System.String	This parameter is reserved for internal Microsoft use.
LinkedPartnerOrganizationId	Required	System.String	This parameter is reserved for internal Microsoft use.
Name	Required	System.String	The *Name* parameter specifies the name of the new role group. The name can have a maximum of 64 characters. If the name contains spaces, enclose the name in quotation marks ("). **Note:**

			If you create a linked role group, we recommend that you include the name of the foreign forest in the name of the role group so that you can more easily associate the linked role group and the associated foreign forest. This is especially important if you have multiple forests.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CustomConfigWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	This parameter is available only in on-premises Exchange 2013. The *CustomConfigWriteScope* parameter specifies the existing configuration-based management scope to associate with management role assignments created

			with this role group. If the management scope name contains spaces, enclose the name in quotation marks ("). Use the Get-ManagementScope cmdlet to retrieve a list of existing management scopes.
CustomRecipientWriteScope	Optional	Microsoft.Exchange.Configuration.Tasks.ManagementScopeIdParameter	The *CustomRecipientWriteScope* parameter specifies the existing recipient-based management scope to associate with management role assignments created with this role group. If the management scope name contains spaces, enclose the name in quotation marks ("). Use the Get-ManagementScope cmdlet to retrieve a list of existing management scopes. If you use the *CustomRecipientWriteScope* parameter, you can't use the

			RecipientOrganizationalUnitScope parameter.
Description	Optional	System.String	The *Description* parameter specifies the description that's displayed when the role group is viewed using the **Get-RoleGroup** cmdlet. Enclose the description in quotation marks (").
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the friendly name of the role group. If the name contains spaces, enclose the name in quotation marks ("). This parameter can have a maximum length of 256 characters.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration

			change to Active Directory.
ExternalDirectoryObjectId	Optional	System.String	This parameter is reserved for internal Microsoft use.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
LinkedCredential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *LinkedCredential* parameter specifies credentials to use to access the domain

			controller specified by the *LinkedDomainController* parameter. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
ManagedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ManagedBy* parameter specifies the users or USGs who can modify the configuration of a role group or add and remove members to or from a role group. You can use the name, distinguished name (DN), or primary SMTP address of the user or USG that you want to add. If the name of the user or USG contains spaces, enclose the name in quotation marks ("). If you want to add more than one user or USG,

			separate them using commas.
Members	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Members* parameter specifies the mailboxes or USGs to add as a member of the role group. You can use the name, DN, or primary SMTP address of the user or USG you want to add. If the name of the user or USG contains spaces, enclose the name in quotation marks ("). If you want to add more than one user or USG, separate them using commas. If you use the *Members* parameter, you can't use the *LinkedForeignGroup*, *LinkedDomainController*, or *LinkedCredential* parameters.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
PartnerManaged	Optional	System.Management.Automation.SwitchPara	This parameter is reserved for internal

		meter	Microsoft use.
RecipientOrganizationalUnitScope	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	This parameter is available only in on-premises Exchange 2013. The *RecipientOrganizationalUnitScope* parameter specifies the organizational unit (OU) scope added to the role assignments created when the role group is created. If you use the *RecipientOrganizationalUnitScope* parameter, you can't use the *CustomRecipientWriteScope* parameter. To specify an OU, use the syntax: *domain/ou*. If the OU name contains spaces, enclose the domain and OU in quotation marks (").
Roles	Optional	Microsoft.Exchange.Configuration.Tasks.RoleIdParameter[]	The *Roles* parameter specifies the management roles to assign to the role group when it's created. If a role name contains spaces, enclose the

			name in quotation marks ("). If you want to assign more that one role, separate the role names with commas. For a list of built-in management roles that you can assign to a role group, see Built-in management roles.
SamAccountName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *SamAccountName* parameter specifies the logon name used to support clients and servers running older versions of the operating system, such as Microsoft Windows NT 4.0, Windows 98, Windows 95, and LAN Manager. This attribute must contain fewer than 20 characters. If you don't specify this parameter, Active Directory creates a value for the *SamAccountName*

			parameter automatically, based on the user principal name (UPN).
ValidationOrganization	Optional	System.String	This parameter is reserved for internal Microsoft use.
WellKnownObjectGuid	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-RoleGroup

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-RoleGroup** cmdlet to remove a management role group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-RoleGroup -Identity <RoleGroupIdParameter> [-
BypassSecurityGroupManagerCheck <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-ForReconciliation <SwitchParameter>] [-RemoveWellKnownObjectGuid
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the Training Administrators role group.

```
Remove-RoleGroup "Training Administrators"
```

EXAMPLE 2

This example removes the Vancouver Recipient Administrators role group. Because the user running the command wasn't added to the **ManagedBy** property of the role group, the *BypassSecurityGroupManagerCheck* switch must be used. The user is assigned the Role Management role, which enables the user to bypass the security group manager check.

```
Remove-RoleGroup "Vancouver Recipient Administrators" -
BypassSecurityGroupManagerCheck
```

Detailed Description

When you remove a role group, all the management role assignments assigned management roles to the role group are also removed. The management roles aren't removed. Members of a removed role group can no longer manage a feature if the role group was the only means by which they were granted access to the feature.

You can't remove built-in role groups.

If the **ManagedBy** property has been populated with role group managers, the user removing the role group must be a role group manager. Alternately, if the user is a member of the Organization Management role group or is directly or indirectly assigned the Role Management role, the *BypassSecurityGroupManagerCheck* switch can be used to override the security group management check.

For more information about role groups, see Understanding management role groups.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleGroupIdParameter	The *Identity* parameter specifies the role group to remove. If the role group name contains spaces, enclose the name in quotation marks (").
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* switch enables a user who hasn't been added to the `ManagedBy` property to remove a role group. The user must be a member of the Organization Management role group or be assigned, either directly or indirectly, the Role

			Management role.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and

			prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
ForReconciliation	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
RemoveWellKnownObjectGuid	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-RoleGroup

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-RoleGroup** cmdlet to modify who can add or remove members to or from management role groups or change the name of the role group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-RoleGroup <COMMON PARAMETERS>

Set-RoleGroup -LinkedForeignGroupSid <SecurityIdentifier> <COMMON
PARAMETERS>

Set-RoleGroup -LinkedDomainController <String> -LinkedForeignGroup
<UniversalSecurityGroupIdParameter> [-LinkedCredential <PSCredential>]
<COMMON PARAMETERS>

COMMON PARAMETERS: -Identity <RoleGroupIdParameter> [-
BypassSecurityGroupManagerCheck <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-Description <String>] [-DisplayName <String>] [-
DomainController <Fqdn>] [-ExternalDirectoryObjectId <Guid>] [-Force
<SwitchParameter>] [-ManagedBy <MultiValuedProperty>] [-Name <String>] [-
WellKnownObjectGuid <Guid>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the role group managers list to David and Christine on the London Recipient

Administrators role group.

```
Set-RoleGroup "London Recipient Administrators" -ManagedBy
"David", "Christine"
```

EXAMPLE 2

This example sets the role group managers list to the Seattle Role Administrators USG on the Seattle Administrators role group. Because the user running the command wasn't added to the **ManagedBy** property of the role group, the *BypassSecurityGroupManagerCheck* switch must be used. The user is assigned the Role Management role, which enables the user to bypass the security group manager check.

```
Set-RoleGroup "Seattle Administrators" -ManagedBy "Seattle
Role Administrators" -BypassSecurityGroupManagerCheck
```

EXAMPLE 3

This example modifies the linked foreign USG on the existing linked role group ContosoUsers: Toronto Recipient Admins. The foreign USG that should be linked is Toronto Tier 2 Administrators.

First, the credentials for the user forest need to be stored in a variable so they can be used with the **Set-RoleGroup** cmdlet. The following command retrieves the credentials using the **Get-Credential** cmdlet and stores them in the `$Credentials` variable.

```
$Credentials = Get-Credential
```

Then, the foreign USG on the ContosoUsers: Toronto Recipient Admins linked role group is modified using the following command.

```
Set-RoleGroup "ContosoUsers: Toronto Recipient Admins" -
LinkedDomainController dc02.contosousers.contoso.com -
LinkedCredential $Credentials -LinkedForeignGroup "Toronto
Tier 2 Administrators"
```

Detailed Description

If you want to add or remove members to or from an existing role group, use the **Add-RoleGroupMember** or **Remove-RoleGroupMember** cmdlets. If you want to add or remove management role assignments to or from a role group, use the **New-ManagementRoleAssignment** or **Remove-ManagementRoleAssignment** cmdlets. If you want to add or remove members to or from a linked role group, you must add or remove the members to or from the foreign universal security group (USG) in the foreign forest. To find the foreign USG, use the **Get-RoleGroup** cmdlet.

If the **ManagedBy** property is populated with role group managers, the user configuring a role group must be a role group manager. Alternately, if the user is a member of the Organization Management role group or is directly or indirectly assigned the Role Management role, the *BypassSecurityGroupManagerCheck* switch can be used to override the security group management check.

For more information about role groups, see Understanding management role groups.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleGroupIdParameter	The *Identity* parameter specifies the name of the role group to modify. If the name contains spaces, enclose the name in quotation marks (").
LinkedDomainController	Required	System.String	This parameter is available only in on-premises Exchange 2013. The *LinkedDomainController* parameter specifies the fully qualified domain name (FQDN) or IP address of the domain controller in the forest where the foreign USG resides. The domain controller

			you specify is used to get security information for the foreign USG specified by the *LinkedForeignGroup* parameter. You can only use the *LinkedDomainController* parameter with a linked role group.
LinkedForeignGroup	Required	Microsoft.Exchange.Configuration.Tasks.UniversalSecurityGroupIdParameter	This parameter is available only in on-premises Exchange 2013. The *LinkedForeignGroup* parameter specifies the name of the foreign USG you want to link this role group to. If the foreign USG name contains spaces, enclose the name in quotation marks ("). You can only use the *LinkedForeignGroup* parameter to change the foreign USG linked to an existing linked role group. You can't change a standard role

			group to a linked role group by using the **Set-RoleGroup** cmdlet. You must create a role group using the **New-RoleGroup** cmdlet. If you use the *LinkedForeignGroup* parameter, you must specify a domain controller in the *LinkedDomainController* parameter.
LinkedForeignGroupSid	Required	System.Security.Principal.SecurityIdentifier	This parameter is reserved for internal Microsoft use.
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* switch enables a user who hasn't been added to the **ManagedBy** property to modify a role group. The user must be a member of the Organization Management role group or be assigned, either directly or indirectly, the Role Management role.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies the description displayed when the role group is viewed using the **Get-RoleGroup** cmdlet. Enclose the description in quotation marks (').
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the friendly name of the role group. If the name contains spaces, enclose the name in quotation marks ('). This parameter can have a maximum length of 256 characters.
DomainController	Optional	Microsoft.Exchange.Dat	This parameter is available only in on-

		a.Fqdn	premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExternalDirectoryObjectId	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.

LinkedCredential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *LinkedCredential* parameter specifies credentials to use to access the domain controller specified by the *LinkedDomainController* parameter. You can only use the *LinkedCredential* parameter with a linked role group. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
ManagedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ManagedBy* parameter specifies the users or USG who can modify the configuration of a role group or add or

			remove members to or from a role group. The list you specify with this parameter overwrites the existing `ManagedBy` list. To add or remove individual role group managers, and for more information about modifying multivalued properties, see Modifying multivalued properties. You can use the name, distinguished name (DN), or primary SMTP address of the user or USG you want to add. If the name of the user or USG contains spaces, enclose the name in quotation marks ("). If you want to add more than one user or USG, separate them using commas.
Name	Optional	System.String	The *Name* parameter specifies the name of the role group. The name can contain up to 64 characters. If the

			name contains spaces, enclose the name in quotation marks (").
WellKnownObjectGuid	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-RoleGroupMember

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-07-22*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Add-RoleGroupMember** cmdlet to add members to a management role group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-RoleGroupMember -Identity <RoleGroupIdParameter> -Member
<SecurityPrincipalIdParameter> [-BypassSecurityGroupManagerCheck
<SwitchParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-IncludeSoftDeletedObjects <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds the user David to the role group Recipient Management.

```
Add-RoleGroupMember "Recipient Management" -Member David
```

EXAMPLE 2

This example finds all the mailboxes that are part of the Sales department and adds them to the Sales and Marketing Group role group.

```
Get-User -Filter { Department -Eq "Sales" -And
RecipientType -Eq "UserMailbox" } | Get-Mailbox | Add-
RoleGroupMember "Sales and Marketing Group"
```

If you want to verify that the correct members will be added before committing changes, use the *WhatIf* parameter.

```
Get-User -Filter { Department -Eq "Sales" -And
RecipientType -Eq "UserMailbox" } | Get-Mailbox | Add-
RoleGroupMember "Sales and Marketing Group" -WhatIf
```

For more information about pipelining, and the *WhatIf* parameter, see the following topics:

- Pipelining
- WhatIf, Confirm, and ValidateOnly switches

EXAMPLE 3

This example adds the Training Assistants USG to the Training Administrators role group. Because the user running the command wasn't added to the **ManagedBy** property of the role group, the

BypassSecurityGroupManagerCheck switch must be used. The user is assigned the Role Management role, which enables the user to bypass the security group manager check.

```
Add-RoleGroupMember "Training Administrators" -Member
"Training Assistants" -BypassSecurityGroupManagerCheck
```

Detailed Description

When you add a member to a role group, that mailbox, universal security group (USG), or computer is given the effective permissions provided by the management roles assigned to the role group.

If the **ManagedBy** property has been populated with role group managers, the user adding a role group member must be a role group manager. Alternately, if the user is a member of the Organization Management role group or is directly or indirectly assigned the Role Management role, the *BypassSecurityGroupManagerCheck* switch can be used to override the security group management check.

If the role group is a linked role group, you can't use the **Add-RoleGroupMember** cmdlet to add members to the role group. Instead, you need to add members to the foreign USG that's linked to the linked role group. To find the foreign USG that's linked to a role group, use the **Get-RoleGroup** cmdlet.

For more information about role groups, see Understanding management role groups.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleGroupIdParameter	The *Identity* parameter specifies the role group to add a member to. If the role group name contains spaces, enclose the name in quotation marks (").
Member	Required	Microsoft.Exchange.Configuration.Tasks.Secur	The *Member* parameter specifies the mailbox,

		ityPrincipalIdParameter	USG, or computer to add to a role group. You can only specify one member at a time. If the member name contains spaces, enclose the name in quotation marks (").
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* switch enables a user who hasn't been added to the **ManagedBy** property to add a member to a role group. The user must be a member of the Organization Management role group or be assigned, either directly or indirectly, the Role Management role.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't

			have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IncludeSoftDeletedObjects	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-RoleGroupMember

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-07-22*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-RoleGroupMember** cmdlet to retrieve a list of members of a management role group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-RoleGroupMember -Identity <RoleGroupMemberIdParameter> [-
DomainController <Fqdn>] [-IncludeSoftDeletedObjects <SwitchParameter>] [-
ReadFromDomainController <SwitchParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example retrieves a list of all the members of the Recipient Administrators role group.

```
Get-RoleGroupMember "Recipient Administrators"
```

EXAMPLE 2

This example retrieves a list of all the members of the Organization Administrators role group as seen by the domain controller closest to the user running the command.

```
Get-RoleGroupMember "Organization Administrators" -
ReadFromDomainController
```

Detailed Description

For more information about role groups, see Understanding management role groups.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleGroupMemberIdParameter	The *Identity* parameter specifies the role group for which member information should be retrieved. If the role group name contains spaces, enclose the name in quotation marks (").
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
IncludeSoftDeletedObjects	Optional	System.Management.Automation.SwitchPara	This parameter is reserved for internal

		meter	Microsoft use.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* switch specifies that the role group information should be read from a domain controller in the user's domain. If you use the **Set-AdServerSettings** cmdlet to include all role groups in the forest and don't use this parameter, it's possible that the role group information is read from a global catalog with outdated information.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all the members of a role group, use `Unlimited` for the value of this parameter. The default

			value is 1,000.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-RoleGroupMember

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-07-22*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-RoleGroupMember** cmdlet to remove a member of a management role group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-RoleGroupMember -Identity <RoleGroupIdParameter> -Member
<SecurityPrincipalIdParameter> [-BypassSecurityGroupManagerCheck
<SwitchParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-IncludeSoftDeletedObjects <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the user David from the role group Recipient Management.

```
Remove-RoleGroupMember "Recipient Management" -Member David
```

EXAMPLE 2

This example finds all the mailboxes that are part of the Sales department and removes them from the Sales and Marketing Group role group.

```
Get-User -Filter { Department -Eq "Sales" -And -
RecipientType -Eq "UserMailbox" } | Get-Mailbox | Remove-
RoleGroupMember "Sales and Marketing Group"
```

If you want to verify that the correct members will be removed before committing changes, use the *WhatIf* parameter.

```
Get-User -Filter { Department -Eq "Sales" -And -
RecipientType -Eq "UserMailbox" } | Get-Mailbox | Remove-
RoleGroupMember "Sales and Marketing Group" -WhatIf
```

For more information about pipelining, and the **WhatIf** parameter, see the following topics:

- Pipelining
- WhatIf, Confirm, and ValidateOnly switches

EXAMPLE 3

This example removes the Training Assistants USG from the Training Administrators role group. Because the user running the command wasn't added to the **ManagedBy** property of the role group, the *BypassSecurityGroupManagerCheck* switch must be used. The user is assigned the Role Management role, which enables the user to bypass the security group manager check.

```
Remove-RoleGroupMember "Training Administrators" -Member
"Training Assistants" -BypassSecurityGroupManagerCheck
```

Detailed Description

When you remove a member from a role group, that member can no longer manage the features made available by the role group if the role group is the only means by which the member is granted access to the feature.

If the **ManagedBy** property has been populated with role group managers, the user removing a role group member must be a role group manager. Alternately, if the user is a member of the Organization Management role group or is directly or indirectly assigned the Role Management role, the *BypassSecurityGroupManagerCheck* switch can be used to override the security group management check.

If the role group is a linked role group, you can't use the **Remove-RoleGroupMember** cmdlet to remove members from the role group. Instead, you need to remove members from the foreign universal security group (USG) linked to the linked role group. To find the foreign USG linked to a role group, use the **Get-RoleGroup** cmdlet.

For more information about role groups, see Understanding management role groups.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleGroupIdParameter	The *Identity* parameter specifies the role group that you want to remove a member from. If the role group name contains spaces, enclose the name in quotation marks (").
Member	Required	Microsoft.Exchange.Configuration.Tasks.SecurityPrincipalIdParameter	The *Member* parameter specifies the mailbox or USG to remove from a role group. You can only specify one member at a time. If the member name contains spaces, enclose the name in quotation marks (").
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* switch enables a user who hasn't been added to the **ManagedBy** property to remove a member from a role

			group. The user must be a member of the Organization Management role group or be assigned, either directly or indirectly, the Role Management role.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

IncludeSoftDeletedObjects	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-RoleGroupMember

Exchange Management Shell > Exchange 2013 cmdlets > Permissions cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-07-22*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Update-RoleGroupMember** cmdlet to modify the members of a management role group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-RoleGroupMember -Identity <RoleGroupIdParameter> [-
BypassSecurityGroupManagerCheck <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-
IncludeSoftDeletedObjects <SwitchParameter>] [-Members
<MultiValuedProperty>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the Recipient Administrators role group membership list to Mark, Jane, Mary, and Fred.

```
Update-RoleGroupMember "Recipient Administrators" -Members
"Mark", "Jane", "Mary", "Fred"
```

EXAMPLE 2

This example sets the Recipient Administrators role group membership list to Mark, Jane, Mary, and Fred. Because the user running the command wasn't added to the **ManagedBy** property of the role group, the *BypassSecurityGroupManagerCheck* switch must be used. The user is assigned the Role Management role, which enables the user to bypass the security group manager check.

```
Update-RoleGroupMember "Recipient Administrators" -Members
"Mark", "Jane", "Mary", "Fred" -
BypassSecurityGroupManagerCheck
```

EXAMPLE 3

This example adds multiple members to, and removes multiple members from, a role group without replacing all the existing members on the role group. This example makes use of multivalued property syntax that's described in the topic Modifying multivalued properties. When you use this multivalued property syntax, you must manually retrieve the *Identity* of the mailbox or security group that you want to add to or remove from the role group. Use the syntax that matches the type of object you want to add or remove:

- **Mailbox** If you want to add or remove a mailbox, use the syntax `(Get-Mailbox "<Alias or Name>").Identity`
- **Security Group** If you want to add or remove a security group, use the syntax `(Get-Group "<Name>").Identity`

```
Update-RoleGroupMember "Organization Management" -Members
@{Add=(Get-Mailbox David).Identity, (Get-Group "Help Desk
```

```
Managers").Identity; Remove=(Get-Mailbox
"Christine").Identity, (Get-Mailbox "Isabel").Identity}
```

Detailed Description

The **Update-RoleGroupMember** cmdlet enables you to replace the entire membership list for a role group or perform programmatic addition or removal of multiple members at a single time. The membership list that you specify with the *Members* parameter on this cmdlet replaces the membership list for the specific role group. For this reason, take care when using this cmdlet so you don't mistakenly overwrite role group membership.

The **Add-RoleGroupMember** and **Remove-RoleGroupMember** cmdlets can be used to add or remove role group members. You can combine these cmdlets with other cmdlets, such as **Get-Mailbox**, to add or remove multiple members without overwriting the entire membership list at once.

If the **ManagedBy** property has been populated with role group managers, the user updating role group membership must be a role group manager. Alternately, if the user is a member of the Organization Management role group or is directly or indirectly assigned the Role Management role, the *BypassSecurityGroupManagerCheck* switch can be used to override the security group management check.

If the role group is a linked role group, you can't use the **Update-RoleGroupMember** cmdlet to modify members on the role group. Instead, you need to modify members on the foreign universal security group (USG) that's linked to the linked role group. To find the foreign USG that's linked to a role group, use the **Get-RoleGroup** cmdlet.

For more information about role groups, see Understanding management role groups.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Role groups" entry in the Role management permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RoleGroupIdParameter	The *Identity* parameter specifies the role group whose membership you want to modify. If the name of the role group contains spaces, enclose

			the name in quotation marks (").
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* switch enables a user who hasn't been added to the **ManagedBy** property to modify a role group's membership. The user must be a member of the Organization Management role group or be assigned, either directly or indirectly, the Role Management role.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IncludeSoftDeletedObjects	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Members	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Members* parameter specifies the mailboxes or USG that should be members of a security group. If the member name contains spaces, enclose the name in quotation marks ("). Separate multiple members using commas. **Caution:** The list that you specify using the *Members* parameter overwrites the existing membership list of the role group. If you want to add or remove individual members to or from a role group, use the **Add-RoleGroupMember** or **Remove-RoleGroupMember** cmdlets. If you want to add or remove multiple

			members without replacing the entire membership list, see the Examples section.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Policy and compliance cmdlets

Exchange Server 2013 > Exchange Management Shell > Exchange 2013 cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-10*

Archive mailbox cmdlets

Test-ArchiveConnectivity

Audit log cmdlets

Administrator audit log cmdlets

Search-AdminAuditLog

Write-AdminAuditLog

Get-AdminAuditLogConfig

Set-AdminAuditLogConfig

New-AdminAuditLogSearch

Get-AuditLogSearch

Mailbox audit log cmdlets

Get-MailboxAuditBypassAssociation

Set-MailboxAuditBypassAssociation

New-MailboxAuditLogSearch

Search-MailboxAuditLog

Data Loss Prevention (DLP) cmdlets

Get-ClassificationRuleCollection

New-ClassificationRuleCollection

Remove-ClassificationRuleCollection

Set-ClassificationRuleCollection

Get-DataClassification

New-DataClassification

Remove-DataClassification

Set-DataClassification

Get-DlpPolicy

New-DlpPolicy

Remove-DlpPolicy

Set-DlpPolicy

Export-DlpPolicyCollection

Import-DlpPolicyCollection

Get-DlpPolicyTemplate

Import-DlpPolicyTemplate

Remove-DlpPolicyTemplate

New-Fingerprint

Get-PolicyTipConfig

New-PolicyTipConfig

Remove-PolicyTipConfig

Set-PolicyTipConfig

In-Place eDiscovery and In-Place Hold cmdlets

Get-MailboxSearch

New-MailboxSearch

Remove-MailboxSearch

Set-MailboxSearch

Start-MailboxSearch

Stop-MailboxSearch

Information Rights Management (IRM) cmdlets

Get-IRMConfiguration

Set-IRMConfiguration

Test-IRMConfiguration

Disable-OutlookProtectionRule

Enable-OutlookProtectionRule

Get-OutlookProtectionRule

New-OutlookProtectionRule

Remove-OutlookProtectionRule

Set-OutlookProtectionRule

Get-RMSTemplate

Journaling cmdlets

Disable-JournalRule

Enable-JournalRule

Get-JournalRule

New-JournalRule

Remove-JournalRule

Set-JournalRule

Export-JournalRuleCollection

Import-JournalRuleCollection

Message classification cmdlets

Get-MessageClassification

New-MessageClassification

Remove-MessageClassification

Set-MessageClassification

Messaging records management cmdlets

Start-ManagedFolderAssistant

Stop-ManagedFolderAssistant

Get-RetentionPolicy

New-RetentionPolicy

Remove-RetentionPolicy

Set-RetentionPolicy

Get-RetentionPolicyTag

New-RetentionPolicyTag

Remove-RetentionPolicyTag

Set-RetentionPolicyTag

Transport rules cmdlets

Disable-TransportRule

Enable-TransportRule

Get-TransportRule

New-TransportRule

Remove-TransportRule

Set-TransportRule

Get-TransportRuleAction

Export-TransportRuleCollection

Import-TransportRuleCollection

Get-TransportRulePredicate

Search-AdminAuditLog

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Search-AdminAuditLog** cmdlet to search the contents of the administrator audit log.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Search-AdminAuditLog [-Cmdlets <MultiValuedProperty>] [-EndDate
<ExDateTime>] [-ExternalAccess <$true | $false>] [-Identity
<OrganizationIdParameter>] [-IsSuccess <$true | $false>] [-ObjectIds
<MultiValuedProperty>] [-Parameters <MultiValuedProperty>] [-ResultSize
<Int32>] [-StartDate <ExDateTime>] [-StartIndex <Int32>] [-UserIds
<MultiValuedProperty>] [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example finds all the administrator audit log entries that contain either the **New-RoleGroup** or the **New-ManagementRoleAssignment** cmdlet.

```
Search-AdminAuditLog -Cmdlets New-RoleGroup, New-
ManagementRoleAssignment
```

EXAMPLE 2

This example finds all the administrator audit log entries that match the following criteria:

- *Cmdlets* **Set-Mailbox**
- *Parameters* *UseDatabaseQuotaDefaults*, *ProhibitSendReceiveQuota*, *ProhibitSendQuota*
- *StartDate* 01/24/2012
- *EndDate* 02/12/2012
- The command completed successfully

```
Search-AdminAuditLog -Cmdlets Set-Mailbox -Parameters
UseDatabaseQuotaDefaults, ProhibitSendReceiveQuota,
ProhibitSendQuota -StartDate 01/24/2012 -EndDate 02/12/2012
-IsSuccess $true
```

EXAMPLE 3

This example displays all the comments written to the administrator audit log by the **Write-AdminAuditLog** cmdlet.

First, store the audit log entries in a temporary variable using the following command.

```
$LogEntries = Search-AdminAuditLog -Cmdlets Write-
AdminAuditLog
```

Then, iterate through all the audit log entries returned and display the **Parameters** property using the following command.

```
$LogEntries | ForEach { $_.CmdletParameters }
```

EXAMPLE 4

This example returns entries in the administrator audit log of an Exchange Online organization for cmdlets run by Microsoft datacenter administrators between September 17, 2013 and October 2, 2013.

```
Search-AdminAuditLog -ExternalAccess $true -StartDate
09/17/2013 -EndDate 10/02/2013
```

Detailed Description

If you run the **Search-AdminAuditLog** cmdlet without any parameters, up to 1,000 log entries are returned by default.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "View-only administrator audit logging" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Cmdlets	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Cmdlets* parameter specifies the cmdlets you want to search for in the administrator audit log. Only the log entries that contain the cmdlets you specify are returned. If you want to specify more than one cmdlet, separate each cmdlet with a comma.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
EndDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *EndDate* parameter specifies the scope of the administrator audit log results to log

			entries that occurred on or before the specified date. Specify the day, month, and year in the format specified in your regional settings.
ExternalAccess	Optional	System.Boolean	The *ExternalAccess* parameter returns only audit log entries for cmdlets that were run by a user outside of your organization. In Exchange Online, use this parameter to return audit log entries for cmdlets run by Microsoft datacenter administrators.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
IsSuccess	Optional	System.Boolean	The *IsSuccess* parameter specifies whether only administrator audit log entries that indicated a success or failure should be returned. Valid values are `$true` and `$false`.

ObjectIds	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ObjectIds* parameter specifies that only administrator audit log entries that contain the specified changed objects should be returned. This parameter accepts a variety of objects, such as mailbox aliases, Send connector names, and so on. If you want to specify more than one object ID, separate each ID with a comma.
Parameters	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Parameters* parameter specifies the parameters you want to search for in the administrator audit log. Only the log entries that contain the parameters you specify are returned. You can only use this parameter if you use the *Cmdlets* parameter. If you want to specify more than one parameter, separate each parameter with a

			comma.
ResultSize	Optional	System.Int32	The *ResultSize* parameter specifies the maximum number of administrator audit log entries to return. The minimum value is 1 and the maximum value is 250000. The default value is 1000.
StartDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *StartDate* parameter specifies the scope of the administrator audit log results to log entries that occurred on or after the specified date. Specify the day, month, and year in the format specified in your regional settings.
StartIndex	Optional	System.Int32	The *StartIndex* parameter specifies the position in the result set where the displayed results start.
UserIds	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UserIds* parameter specifies that only the administrator audit log entries that contain the specified ID of the user

			who ran the cmdlet should be returned. If you want to specify more than one user ID, separate each ID with a comma.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Write-AdminAuditLog

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Write-AdminAuditLog** cmdlet to write a comment to the administrator audit log.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Write-AdminAuditLog -Comment <String> [-Identity
<OrganizationIdParameter>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds a comment to the administrator audit log.

```
Write-AdminAuditLog -Comment "Ran custom script."
```

Detailed Description

When the **Write-AdminAuditLog** cmdlet runs, the value provided in the *Comment* parameter is included in the log entry.

For the **Write-AdminAuditLog** cmdlet to write to the audit log, it must be included in the list of cmdlets being logged by administrator audit logging.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Write to audit log" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Comment	Required	System.String	The *Comment* parameter specifies the comment to add to the administrator audit log. The maximum length is 500 characters. If the comment you specify contains spaces, enclose the comment in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value

			with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AdminAuditLogConfig

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-06*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-AdminAuditLogConfig** cmdlet to view the administrator audit logging configuration settings.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-AdminAuditLogConfig [-Identity <OrganizationIdParameter>] [-
DomainController <Fqdn>] [-IgnoreDehydratedFlag <SwitchParameter>]
```

Examples

EXAMPLE 1

This example displays the administrator audit logging settings. The output of the **Get-AdminAuditLogConfig** cmdlet is piped to the **Format-List** cmdlet. For more information about piping and the **Format-List** cmdlet, see the following topics:

- Pipelining
- Working with command output

```
Get-AdminAuditLogConfig | Format-List
```

Detailed Description

When audit logging is enabled, a log entry is created for each cmdlet that's run, excluding **Get** cmdlets.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Administrator audit logging" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AdminAuditLogConfig

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-AdminAuditLogConfig** cmdlet to configure the administrator audit logging configuration settings.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-AdminAuditLogConfig [-Identity <OrganizationIdParameter>] [-
AdminAuditLogAgeLimit <EnhancedTimeSpan>] [-AdminAuditLogCmdlets
<MultiValuedProperty>] [-AdminAuditLogEnabled <$true | $false>] [-
AdminAuditLogExcludedCmdlets <MultiValuedProperty>] [-
AdminAuditLogParameters <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-IgnoreDehydratedFlag <SwitchParameter>] [-LogLevel <None | Verbose>] [-
Name <String>] [-TestCmdletLoggingEnabled <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables administrator audit logging for every cmdlet and every parameter in the organization, with the exception of **Get** cmdlets.

```
Set-AdminAuditLogConfig -AdminAuditLogEnabled $true -
AdminAuditLogCmdlets * -AdminAuditLogParameters *
```

EXAMPLE 2

This example enables administrator audit logging for specific cmdlets run in the organization. Any parameter used on the specified cmdlets is logged. Every time a specified cmdlet is run, a log entry is added to the audit log.

```
Set-AdminAuditLogConfig -AdminAuditLogEnabled $true -
AdminAuditLogCmdlets *Mailbox, *Management*,
*TransportRule* -AdminAuditLogParameters *
```

EXAMPLE 3

This example enables administrator audit logging only for specific parameters that are specified when running specific cmdlets. The parameter name and the cmdlet name must match the strings

specified with the *AdminAuditLogCmdlets* and *AdminAuditLogParameters* parameters. For example, a log entry is generated only when a parameter with the string "Address" in the name is run on a cmdlet with the string "Mailbox" in its name.

```
Set-AdminAuditLogConfig -AdminAuditLogEnabled $true -
AdminAuditLogCmdlets *Mailbox* -AdminAuditLogParameters
*Address*
```

Detailed Description

When audit logging is enabled, a log entry is created for each cmdlet run, excluding **Get** cmdlets. In Microsoft Exchange Server 2013, log entries are stored in a hidden mailbox and accessed using the **Search-AdminAuditLog** or **New-AdminAuditLogSearch** cmdlets.

Important:
The **Set-AdminAuditLogConfig**, **Enable-CmdletExtensionAgent**, and **Disable-CmdletExtensionAgent** cmdlets are logged when they're run regardless of whether administrator audit logging is enabled or disabled. Administrator audit logging relies on Active Directory replication to replicate the configuration settings you specify to the domain controllers in your organization. Depending on your replication settings, the changes you make may not be immediately applied to all computers running Exchange 2013 in your organization. Changes to the audit log configuration may take up to 60 minutes to be applied on computers that have the Exchange Management Shell open at the time a configuration change is made. If you want to apply the changes immediately, close and reopen the Shell on each computer.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Administrator audit logging" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
AdminAuditLogAgeLimit	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *AdminAuditLogAgeLimit* parameter specifies how long each log entry should be kept before it's deleted. The default age limit is 90 days.

			To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to set the audit log age limit to 120 days, use the syntax `120.00:00:00`. **Caution:** Setting the age limit to a value less than the current limit causes log entries older than the new limit to be deleted. Setting the age limit to `0` purges the audit log of all entries.
AdminAuditLogCmdlets	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AdminAuditLogCmdlets* parameter specifies which cmdlets should be audited. You can specify one or more cmdlets, separated by commas. You can also use the wildcard character (*) to match multiple cmdlets in one or more of the entries in the cmdlet list. To audit all cmdlets, specify only the wildcard character (*).
AdminAuditLogEnable	Optional	System.Boolean	The

d			*AdminAuditLogEnabled* parameter specifies whether administrator audit logging is enabled. The default value is `$false`. The valid values are `$true` and `$false`. You must specify an administrator audit log mailbox before you enable logging. **Note:** Changes to the administrator audit log configuration are always logged, regardless of whether audit logging is enabled or disabled.
AdminAuditLogExcludedCmdlets	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AdminAuditLogExcludedCmdlets* parameter specifies which cmdlets should be excluded from auditing. Use this parameter if you want to exclude specific cmdlets you don't want to audit even if they match a wildcard string specified in the *AdminAuditLogCmdlets* parameter. You can specify one or more cmdlets, separated

			by commas. You can also use the wildcard character (*) to match multiple cmdlets in one or more of the entries in the cmdlet list. You can't specify only the wildcard character (*). If you want to clear the list, specify a value of `$null`.
AdminAuditLogParameters	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AdminAuditLogParameters* parameter specifies which parameters should be audited on the cmdlets you specified using the *AdminAuditLogCmdlets* parameter. You can specify one or more parameters, separated by commas. You can also use the wildcard character (*) to match multiple parameters in one or more of the entries in the parameters list. To audit all parameters, specify only the wildcard character (*).
Confirm	Optional	System.Management.Automation.SwitchPar	The *Confirm* switch causes the command to

		ameter	pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.

Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
LogLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuditLogLevel	The *LogLevel* parameter specifies whether additional properties should be included in the log entries. Valid values are `None` and `Verbose`. By default, the `CmdletName`, `ObjectName`, `Parameters` (values), and the `Caller`, `Succeeded`, and `RunDate` properties are included in log entries. When the `Verbose` value is used, the `ModifiedProperties` (old and new) and `ModifiedObjectResolvedName` properties are included in the log entries.
Name	Optional	System.String	The *Name* parameter specifies the name of the **AdminAuditLogConfig** object. **Note:** You don't need to specify

			this parameter when configuring administrator audit logging. It doesn't affect your configuration or how administrator audit logging works.
TestCmdletLoggingEnabled	Optional	System.Boolean	The *TestCmdletLoggingEnabled* parameter specifies whether the execution of test cmdlets should be logged. Test cmdlets begin with the verb **Test**. Valid values are `$true` and `$false`. The default value is `$false`. **Important:** Test cmdlets can produce a large amount of information. As such, you should only enable logging of test cmdlets for a short period of time.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-AdminAuditLogSearch

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-AdminAuditLogSearch** cmdlet to search the contents of the administrator audit log and send the results to one or more mailboxes that you specify.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-AdminAuditLogSearch -EndDate <ExDateTime> -StartDate <ExDateTime> -
StatusMailRecipients <MultiValuedProperty> [-Organization
<OrganizationIdParameter>] [-Cmdlets <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ExternalAccess <$true |
$false>] [-Name <String>] [-ObjectIds <MultiValuedProperty>] [-Parameters
<MultiValuedProperty>] [-UserIds <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example finds all the administrator audit log entries that match the following criteria and sends the results to the david@contoso.com and chris@contoso.com SMTP addresses:

- **Cmdlets** **Set-Mailbox**
- **Parameters** *UseDatabaseQuotaDefaults*, *ProhibitSendReceiveQuota*, *ProhibitSendQuota*
- **StartDate** 01/24/2012
- **EndDate** 02/12/2012

```
New-AdminAuditLogSearch -Name "Mailbox Quota Change Audit"
-Cmdlets Set-Mailbox -Parameters UseDatabaseQuotaDefaults,
ProhibitSendReceiveQuota, ProhibitSendQuota -StartDate
01/24/2012 -EndDate 02/12/2012 -StatusMailRecipients
david@contoso.com, chris@contoso.com
```

EXAMPLE 2

This example returns entries in the administrator audit log of an Exchange Online organization for cmdlets run by Microsoft datacenter administrators between September 25, 2013 and October 24, 2013. The search results are sent to the admin@contoso.com and pilarp@contoso.com SMTP addresses and the text "Datacenter admin audit log" is added to the subject line of the message.

```
New-AdminAuditLogSearch -ExternalAccess $true -StartDate
07/25/2013 -EndDate 10/24/2013 -StatusMailRecipients
admin@contoso.com,pilarp@contoso.com -Name "Datacenter
admin audit log"
```

Detailed Description

After the **New-AdminAuditLogSearch** cmdlet is run, the report is delivered to the mailboxes you specify within 15 minutes. The log is included as an XML attachment on the report email message. The maximum size of the log that can be generated is 10 megabytes (MB).

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "View-only administrator audit logging" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
EndDate	Required	Microsoft.Exchange.Exchange System.ExDateTime	The *EndDate* parameter specifies the scope of the administrator audit log results to log entries that occurred on or before the specified date. Specify the day, month,

			and year in the format specified in your regional settings.
StartDate	Required	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *StartDate* parameter specifies the scope of the administrator audit log results to log entries that occurred on or after the specified date. Specify the day, month, and year in the format specified in your regional settings.
StatusMailRecipients	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *StatusMailRecipients* parameter specifies the recipients that should receive the administrator audit log report. The recipient must be a valid SMTP address. If you want to specify more than one recipient, separate each SMTP address with a comma.
Cmdlets	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Cmdlets* parameter specifies the cmdlets you want to search for

			in the administrator audit log. Only the log entries that contain the cmdlets you specify are returned. If you want to specify more than one cmdlet, separate each cmdlet with a comma.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

ExternalAccess	Optional	System.Boolean	The *ExternalAccess* parameter returns only audit log entries for cmdlets that were run by a user outside of your organization. In Exchange Online, use this parameter to return audit log entries for cmdlets run by Microsoft datacenter administrators.
Name	Optional	System.String	The *Name* parameter specifies the name of the administrator audit log search. The name is shown in the subject line of the audit log report email message. If the name of the report contains spaces, enclose the name in quotation marks (").
ObjectIds	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ObjectIds* parameter specifies that only administrator audit log entries that contain the specified changed objects should be returned. This parameter accepts a variety of objects, such

			as mailboxes, aliases, Send connector names, and so on. If you want to specify more than one object ID, separate each ID with a comma.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Parameters	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Parameters* parameter specifies the parameters you want to search for in the administrator audit log. Only the log entries that contain the parameters you specify are returned. You can only use this parameter if you use the *Cmdlets* parameter. If you want to specify more than one parameter, separate each parameter with a comma.
UserIds	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UserIds* parameter specifies that only the administrator audit log

			entries that contain the specified ID of the user who ran the cmdlet should be returned. If you want to specify more than one user ID, separate each ID with a comma.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-ArchiveConnectivity

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-ArchiveConnectivity** cmdlet to verify archive functionality for a mailbox user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-ArchiveConnectivity -UserSmtp <SmtpAddress> [-Confirm
[<SwitchParameter>]] [-IncludeArchiveMRMConfiguration <SwitchParameter>]
[-MessageId <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests archive connectivity to Gurinder Singh's archive.

```
Test-ArchiveConnectivity -UserSmtp gsingh@contoso.com
```

EXAMPLE 2

This command retrieves mailboxes that have a cloud-based archive provisioned and tests archive connectivity for each mailbox.

```
Get-Mailbox -Filter {ArchiveGuid -ne $null -and
ArchiveDomain -ne $null} -ResultSize Unlimited | Test-
ArchiveConnectivity
```

Detailed Description

Running the **Test-ArchiveConnectivity** cmdlet validates connectivity to a user's archive mailbox. End-to-end verification includes testing whether an on-premises or cloud-based archive is provisioned for the on-premises mailbox user and whether it's enabled and logging on to the archive mailbox on behalf of the user. Successful completion of the command indicates that processes such as the Managed Folder Assistant and Microsoft Office Outlook Web App are able to successfully access the archive mailbox.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "In-Place Archive – Test connectivity" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
UserSmtp	Required	Microsoft.Exchange.Data.SmtpAddress	The *UserSmtp* parameter specifies the SMTP address of the mailbox.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
IncludeArchiveMRMConfiguration	Optional	System.Management.Automation.SwitchParameter	The *IncludeArchiveMRMConfiguration* switch retrieves retention tags provisioned in the user's archive mailbox and the last time the archive was processed by the Managed Folder Assistant.
MessageId	Optional	System.String	This parameter is

			reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AuditLogSearch

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-07*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-AuditLogSearch** cmdlet to return a list of current audit log searches that were created with the **New-AdminAuditLogSearch** or **New-MailboxAuditLogSearch** cmdlets. The **Get-AuditLogSearch** cmdlet also returns audit log searches that are initiated whenever an administrator uses the Exchange Admin Center (EAC) to export audit logs.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-AuditLogSearch [-CreatedAfter <ExDateTime>] [-CreatedBefore
<ExDateTime>] [-Identity <AuditLogSearchIdParameter>] [-Organization
<OrganizationIdParameter>] [-ResultSize <Int32>] [-Type <String>]
```

Examples

Example 1

This example displays detailed information for all current audit log searches.

```
Get-AuditLogSearch | FL
```

Example 2

This example returns a list of current administrator audit log searches.

```
Get-AuditLogSearch -Type admin
```

Detailed Description

Run the **Get-AuditLogSearch** cmdlet to return a list of pending audit log searches. If an audit log search has been completed, it won't be displayed in the list of audit log searches.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "View-only administrator audit logging" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
CreatedAfter	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *CreatedAfter* parameter filters the results to audit log searches that were created after the specified date.

			Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
CreatedBefore	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *CreatedBefore* parameter filters the results to audit log searches that were created before the specified date. Use the short date format defined in the **Regional Options** settings for the

			computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.AuditLogSearchIdParameter	The *Identity* parameter specifies the GUID for an audit log search. You can run the command `Get-AuditLogSearch | Format-List Identity` to display the GUIDs for all current audit log searches.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
ResultSize	Optional	System.Int32	The *ResultSize*

			parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
Type	Optional	System.String	The *Type* parameter specifies the type of audit log searches to return. Use the value `Admin` to return administrator audit log searches or use `mailbox` to return mailbox audit log searches. If the *Type* parameter isn't used, the cmdlet returns both administrator and mailbox audit log searches.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ClassificationRuleCollection

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-ClassificationRuleCollection** cmdlet to view the classification rule collections in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ClassificationRuleCollection [-Identity
<ClassificationRuleCollectionIdParameter>] [-DomainController <Fqdn>] [-
Organization <OrganizationIdParameter>]
```

Examples

Example 1

This example returns a summary list of all classification rule collections.

```
Get-ClassificationRuleCollection
```

Example 2

This example returns detailed information about the classification rule collection named Microsoft Rule Pack. The command is piped to the **Format-List** cmdlet to display the detailed configuration of the specified classification rule collection.

```
Get-ClassificationRuleCollection "Microsoft Rule Pack" |
Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Management.ClassificationDefinitions.ClassificationRuleCollectionIdParameter	The *Identity* parameter specifies the classification rule collection you want to view. You can use any value that uniquely identifies the classification rule collection. For example, you can specify the name, rule collection name or distinguished name (DN) of the classification rule collection.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ClassificationRuleCollection

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-ClassificationRuleCollection** cmdlet to import new classification rule collections into your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-ClassificationRuleCollection -FileData <Byte[]> [-OutOfBoxCollection
<SwitchParameter>] <COMMON PARAMETERS>


New-ClassificationRuleCollection -InstallDefaultCollection
<SwitchParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example imports the classification rule collection file C:\My Documents\External Classification Rule Collection.xml.

```
New-ClassificationRuleCollection -FileData ([Byte[]]$(Get-
Content -Path "C:\My Documents\External Classification Rule
```

```
Collection.xml" -Encoding Byte -ReadCount 0))
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
FileData	Required	System.Byte[]	The *FileData* parameter specifies the classification rule collection file you want to import. For more information about the syntax required to use this parameter, see Syntax.
InstallDefaultCollection	Required	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OutOfBoxCollection	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ClassificationRuleCollection

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-ClassificationRuleCollection** to remove classification rule collections from your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ClassificationRuleCollection -Identity
<ClassificationRuleCollectionIdParameter> [-Confirm [<SwitchParameter>]]
[-DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example removes the classification rule collection named External Classification Rule Collection from your organization.

```
Remove-ClassificationRuleCollection "External
Classification Rule Collection"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.ClassificationDefinitions.ClassificationRuleCollectionIdParameter	The *Identity* parameter specifies the classification rule collection you want to remove. You can use any value that uniquely identifies the classification rule collection. For example, you can specify the name, rule collection name or distinguished name (DN) of the classification rule collection.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You

			must include a colon (：) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ClassificationRuleCollection

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-ClassificationRuleCollection** cmdlet to update existing classification rule collections in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ClassificationRuleCollection -FileData <Byte[]> [-OutOfBoxCollection
<SwitchParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Name <String>] [-Organization <OrganizationIdParameter>] [-
WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example imports the classification rule collection file C:\My Documents\External Classification Rule Collection.xml.

```
Set-ClassificationRuleCollection -FileData ([Byte[]]$(Get-
Content -Path "C:\My Documents\External Classification Rule
Collection.xml" -Encoding Byte -ReadCount 0))
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
FileData	Required	System.Byte[]	The *FileData* parameter specifies the classification rule collection file you want to import. For more information about the syntax required to use this parameter, see Syntax.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			writes this configuration change to Active Directory.
Name	Optional	System.String	The *Name* parameter specifies a unique name for the classification rule collection. If the value contains spaces, enclose the value in quotation marks (").
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OutOfBoxCollection	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DataClassification

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-DataClassification** cmdlet to view the data classification rules in your organization. This cmdlet shows built-in data classification rules, and rules that you created that use document fingerprints.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-DataClassification [-ClassificationRuleCollectionIdentity
<ClassificationRuleCollectionIdParameter>] <COMMON PARAMETERS>


Get-DataClassification [-Identity <DataClassificationIdParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>]
```

Examples

Example 1

This example returns a summary list of all data classification rules in the organization.

```
Get-DataClassification
```

Example 2

This example returns a summary list of all new data classification rules based on document fingerprints that you created.

```
Get-DataClassification -
ClassificationRuleCollectionIdentity "Fingerprint
Classification Collection"
```

Example 3

This example returns details of the built-in data classification rule named SWIFT Code.

```
Get-DataClassification "SWIFT Code" | Format-List
```

Detailed Description

Classification rule packages are used by data loss prevention (DLP) to detect sensitive content in messages.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
ClassificationRuleCollectionIdentity	Optional	Microsoft.Exchange.Management.ClassificationDefinitions.ClassificationRuleCollectionIdParameter	The *ClassificationRuleCollectionIdentity* parameter filters the results by the name of the data classification rule collection. The data classification rule collection that contains the built-in data classification rules is named `Microsoft Rule Package`. The data classification that contains

			new data classification rules that you create that use document fingerprints is named `Fingerprint Classification Collection`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Management.ClassificationDefinitions.DataClassificationIdParameter	The *Identity* parameter specifies the data classification rule that you want to view. You can use any value that uniquely identifies the data classification rule. For example: • Name • **LocalizedName** • **Identity** GUID value
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-DataClassification

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-09*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-DataClassification** cmdlet to create data classification rules that use document fingerprints.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-DataClassification -Description <String> -Fingerprints
<MultiValuedProperty> -Name <String> [-
ClassificationRuleCollectionIdentity
<ClassificationRuleCollectionIdParameter>] [-Confirm [<SwitchParameter>]]
[-DomainController <Fqdn>] [-Locale <CultureInfo>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example creates a new data classification rule named "Contoso Employee-Customer Confidential" that uses the document fingerprints of the files C:\My Documents\Contoso Employee Template.docx and D:\Data\Contoso Customer Template.docx.

```
$Employee_Template = Get-Content "C:\My Documents\Contoso
Employee Template.docx" -Encoding byte
$Employee_Fingerprint = New-Fingerprint -FileData
$Employee_Template -Description "Contoso Employee Template"
$Customer_Template = Get-Content "D:\Data\Contoso Customer
Template.docx" -Encoding byte
```

```
$Customer_Fingerprint = New-Fingerprint -FileData
$Customer_Template -Description "Contoso Customer Template"
New-DataClassification -Name "Contoso Employee-Customer
Confidential" -Fingerprints
$Employee_Fingerprint,$Customer_Fingerprint -Description
"Message contains Contoso employee or customer
information."
```

Detailed Description

Classification rule packages are used by data loss prevention (DLP) to detect sensitive content in messages.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Description	Required	System.String	The *Description* parameter specifies a description for the data classification rule.
Fingerprints	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *Fingerprints* parameter specifies the byte-encoded files to use as document fingerprints. You can use multiple document fingerprints separated by commas. For instructions on how to import documents to use as templates for fingerprints, see New-

			Fingerprint or the Examples section.
Name	Required	System.String	The *Name* parameter specifies a name for the data classification rule. The value must be less than 256 characters. The value of this parameter is used in the Policy Tip that's presented to users in Outlook Web App.
ClassificationRuleCollectionIdentity	Optional	Microsoft.Exchange.Management.ClassificationDefinitions.ClassificationRuleCollectionIdParameter	The *ClassificationRuleCollectionIdentity* parameter is reserved for future use. New data classification rules that you create are automatically added to the classification rule collection named Fingerprint Classification Collection.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Locale	Optional	System.Globalization.CultureInfo	The *Locale* parameter specifies the language that's associated with the data classification rule. Valid input for this parameter is a Microsoft .NET Framework **CultureInfo** class culture code value. For example, `en` for English or `fr` for French. If you don't specify a value for the *Locale*

			parameter, the default language of your Exchange organization is used when you create the data classification rule. You can add additional language translations to the data classification rule by using the **Set-DataClassification** cmdlet.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-DataClassification

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-09*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-DataClassification** cmdlet to remove data classification rules that use document fingerprints. You can't use this cmdlet to remove built-in data classification rules.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-DataClassification -Identity <DataClassificationIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example removes the data classification rule named "Contoso Confidential".

```
Remove-DataClassification "Contoso Confidential"
```

Detailed Description

Classification rule packages are used by data loss prevention (DLP) to detect sensitive content in messages.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss

prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.ClassificationDefinitions.DataClassificationIdParameter	The *Identity* parameter specifies the data classification rule that you want to remove. You can use any value that uniquely identifies the data classification rule. For example: • Name • **LocalizedName** • **Identity** GUID value
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-DataClassification

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-09*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Set-DataClassification** cmdlet to modify data classification rules that use document fingerprints.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-DataClassification -Identity <DataClassificationIdParameter> [-Confirm
[<SwitchParameter>]] [-Description <String>] [-DomainController <Fqdn>] [-
Fingerprints <MultiValuedProperty>] [-IsDefault <SwitchParameter>] [-
Locale <CultureInfo>] [-Name <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example adds a French translation to the existing data classification rule named "Contoso Confidential", and sets this French translation as the default.

```
Set-DataClassification "Contoso Confidential" -Locale fr -
Name "Contoso Confidentiel" -Description "Ce message
contient des informations confidentielles." -IsDefault
```

Example 2

This example removes the existing Spanish translation from the data classification rule named "Contoso Confidential".

```
Set-DataClassification "Contoso Confidential" -Locale es -
Name $null -Description $null
```

Example 3

This example modifies the existing data classification rule named "Contoso Confidential" by adding a new document fingerprint for the file C:\My Documents\Contoso Benefits Template.docx without affecting any existing document fingerprints that are already defined.

```
$Benefits_Template = Get-Content "C:\My Documents\Contoso
Benefits Template.docx" -Encoding byte
$Benefits_Fingerprint = New-Fingerprint -FileData
$Benefits_Template -Description "Contoso Benefits Template"
$Contoso_Confidential = Get-DataClassification "Contoso
Confidential"
$Array = [System.Collections.ArrayList]
($Contoso_Confidential.Fingerprints)
$Array.Add($Benefits_FingerPrint)
```

```
Set-DataClassification $Contoso_Confidential.Identity -
FingerPrints $Array
```

Example 4

This example modifies the data classification rule named "Contoso Confidential" by removing an existing document fingerprint without affecting other document fingerprints that are already defined.

First, you need to see the list of document fingerprints in the data classification rule by running the following commands:

```
$cc = Get-DataClassification "Contoso Confidential"
$a = [System.Collections.ArrayList]($cc.Fingerprints)
$a
```

The first document fingerprint in the list has the index number 0, the second has the index number 1, and so on. You use the index number to specify the document fingerprint that you want to remove.

To remove the first document fingerprint that's displayed in the list, run the following commands:

```
$a.RemoveAt(0)
Set-DataClassification $cc.Identity -FingerPrints $Array
```

Detailed Description

Classification rule packages are used by data loss prevention (DLP) to detect sensitive content in messages.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.ClassificationDefinitions.DataClassificationIdParameter	The *Identity* parameter specifies the data classification rule that you want to modify. You can use any value that

			uniquely identifies the data classification rule. For example: • Name • **LocalizedName** • **Identity** GUID value
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies a description for the data classification rule. You use the *Description* parameter with the *Locale* and *Name* parameters to specify descriptions for the data classification rule in different languages. The localized values of *Description* appear in the **AllLocalizedDescriptions** property of the data classification rule.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-

			premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Fingerprints	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Fingerprints* parameter specifies the byte-encoded document files that are used as fingerprints by the data classification rule. For instructions on how to import documents to use as templates for fingerprints, see New-Fingerprint or the Examples section. For instructions on how to add and remove document fingerprints from an existing data classification rule, see the Examples section.
IsDefault	Optional	System.Management.Automation.SwitchParameter	The *IsDefault* switch is used with the *Locale* parameter to specify the default language for the data classification rule.

			The default *Locale* value is stored in the **DefaultCulture** property. When you change the default *Locale* value, the *Name* value of the data classification rule changes to match the *Name* value that's associated with the new default locale. The original *Name* value when the rule was created is permanently stored the **LocalizedName** property.
Locale	Optional	System.Globalization.CultureInfo	The *Locale* parameter adds or removes languages that are associated with the data classification rule. Valid input for this parameter is a Microsoft .NET Framework **CultureInfo** class culture code value. For example, `en` for English or `fr` for French. Typically, you use the *Locale* parameter with the *Name* and *Description* parameters to add or remove translated names and descriptions for the

			data classification rule. You can also use the *Locale* parameter with the *IsDefault* switch to designate an existing translated name and description as the default. Before you can remove the default translation, you need to set another translation as the default.
Name	Optional	System.String	The *Name* parameter specifies a name for the data classification rule. The value must be less than 256 characters. You use the *Name* parameter with the *Locale* and *Description* parameters to specify names for the data classification rule in different languages. The localized values of *Name* appear in the **AllLocalizedNames** property of the data classification rule. The value of the *Name* parameter is used in the Policy Tip that's presented to users in Outlook Web

			App. When a translated value of the *Name* parameter matches the client's language, the Policy Tip is displayed in the client's language. If no translated values of the *Name* parameter match the client's language, the default translation that's specified by the *IsDefault* parameter is used for the Policy Tip.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DlpPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-DlpPolicy** cmdlet to view information about existing data loss prevention (DLP) policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-DlpPolicy [-Identity <DlpPolicyIdParameter>] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

Example 1

This example returns a summary list of all DLP policies.

```
Get-DlpPolicy
```

Example 2

This example returns detailed information about the DLP policy named Employee Numbers. The command is piped to the **Format-List** cmdlet to display the detailed configuration of the specified DLP policy.

```
Get-DlpPolicy "Employee Numbers" | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.CompliancePrograms.Tasks.DlpPolicyIdParameter	The *Identity* parameter specifies the DLP policy you want to remove. You can use any value that uniquely identifies the DLP policy. For example, you can specify the name, GUID, or distinguished name (DN) of the DLP policy.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-DlpPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-24*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-DlpPolicy** cmdlet to create data loss prevention (DLP) policies in your Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-DlpPolicy [-Confirm [<SwitchParameter>]] [-Description <String>] [-
DomainController <Fqdn>] [-Mode <Audit | AuditAndNotify | Enforce>] [-Name
<String>] [-Organization <OrganizationIdParameter>] [-Parameters
<Hashtable>] [-State <Enabled | Disabled>] [-Template <String>] [-
TemplateData <Byte[]>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a new DLP policy named Contoso PII with the following values:

- The DLP policy is enabled and set to audit only.
- The DLP policy is based on the existing "U.S. Personally Identifiable Information (PII) Data" DLP policy template.

```
New-DlpPolicy -Name "Contoso PII" -Template "U.S.
Personally Identifiable Information (PII) Data"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies an optional description for the DLP policy.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

Mode	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleMode	The *Mode* parameter specifies the action and notification level of the DLP policy. Valid values for this parameter are: • `Audit`: The actions specified by the DLP policy aren't enforced when a message matches the conditions specified by the policy, and the Policy Tip isn't displayed to the user. • `AuditAndNotify`: The actions specified by the DLP policy aren't enforced when a message matches the conditions specified by the policy, but the Policy Tip is displayed to the user in a supported email client. • `Enforce`: The actions specified by the DLP policy are enforced when a message matches the conditions specified by the policy, and the Policy Tip is displayed to the user in a supported email client. By default, the value of this parameter is set to `Audit` when you create a new DLP policy. If the *State* parameter is set to `Disabled`, the value of the *Mode* parameter is

			irrelevant.
Name	Optional	System.String	The *Name* parameter specifies a descriptive name for the DLP policy.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Parameters	Optional	System.Collections.Hashtable	The *Parameters* parameter specifies the parameter values that are required by the DLP policy template that you specify using the *Template* or *TemplateData* parameters. DLP policy templates may contain parameters that need to be populated with values from your organization. For example, a DLP policy template may include an exception group that defines users who are exempt from the DLP policy. Valid input for this parameter is in the format: @{*<parameter1>="<value1>";<parameter2>="<value2>"...*}.

State	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleState	The *State* parameter enables or disables the DLP policy. Valid input for this parameter is `Enabled` or `Disabled`. By default, a new DLP policy that you create is enabled. If you want to create a disabled DLP policy, specify the value `Disabled` for this parameter.
Template	Optional	System.String	The *Template* parameter specifies the existing DLP policy template from which you can create a new DLP policy. You can't use the *Template* and *TemplateData* parameters in the same command.
TemplateData	Optional	System.Byte[]	The *TemplateData* parameter specifies an external DLP policy template file from which you can create a new DLP policy. You can't use the *TemplateData* and *Template* parameters in the same command. For more information about the syntax required to use this parameter, see Syntax.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-DlpPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-DlpPolicy** cmdlet to remove an existing data loss prevention (DLP) policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-DlpPolicy -Identity <DlpPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
```

```
[<SwitchParameter>]]
```

Examples

Example 1

This example removes the existing DLP policy named Contoso PII.

```
Remove-DlpPolicy "Contoso PII"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.CompliancePrograms.Tasks.DlpPolicyIdParameter	The *Identity* parameter specifies the DLP policy you want to remove. You can use any value that uniquely identifies the DLP policy. For example, you can specify the name, GUID, or distinguished name (DN) of the DLP policy.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt,

			use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-DlpPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-DlpPolicy** cmdlet to modify data loss prevention (DLP) policies in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-DlpPolicy -Identity <DlpPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-Description <String>] [-DomainController <Fqdn>] [-
Mode <Audit | AuditAndNotify | Enforce>] [-Name <String>] [-State <Enabled
| Disabled>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example disables the DLP policy named Employee Numbers.

```
Set-DlpPolicy "Employee Numbers" -State Disabled
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.CompliancePrograms.Tasks.DlpPolicyIdParameter	The *Identity* parameter specifies the DLP policy you want to modify. You can use any value that uniquely identifies the DLP policy. For example, you can specify the name, GUID, or distinguished name (DN) of the DLP policy.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies an optional description for the DLP policy.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Mode	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleMode	The *Mode* parameter specifies the action and notification level of the DLP policy. Valid values for this parameter are: • `Audit`: When a message matches the conditions specified by the DLP policy, the actions specified by the policy aren't enforced, and no notification emails are sent. • `AuditAndNotify`: When a message matches the conditions specified by the DLP policy, the actions specified by the policy aren't enforced, but notification emails are sent. • `Enforce`: When a message matches the conditions specified by the DLP policy, the actions specified by the policy are enforced, and notification emails are sent. If the *State* parameter is set to `Disabled`, the value

			of the *Mode* parameter is irrelevant.
Name	Optional	System.String	The *Name* parameter specifies a unique name for the DLP policy.
State	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleState	The *State* parameter enables or disables the DLP policy. Valid input for this parameter is `Enabled` or `Disabled`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-DlpPolicyCollection

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Export-DlpPolicyCollection** cmdlet to export data loss prevention (DLP) policy collections from your organization to a file.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Export-DlpPolicyCollection [-Identity <DlpPolicyIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example exports all the elements of the existing DLP policies to the file C:\My Documents \Contoso PII.xml.

```
$file = Export-DlpPolicyCollection

Set-Content -Path "C:\My Documents\Contoso PII.xml" -Value
$file.FileData -Encoding Byte
```

Detailed Description

The **Export-DlpPolicyCollection** cmdlet exports the settings of the DLP policies and the associated transport rules. You use the **Import-DlpPolicyCollection** to import the DLP policy collection into your organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.CompliancePrograms.Tasks.DlpPolicyIdParameter	The *Identity* parameter specifies the DLP policy you want to export. You can use any value that uniquely identifies the DLP policy. For example, you can specify the name, GUID, or distinguished name

			(DN) of the DLP policy.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Import-DlpPolicyCollection

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Import-DlpPolicyCollection** cmdlet to import data loss prevention (DLP) policy collections into your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Import-DlpPolicyCollection -FileData <Byte[]> [-Identity
<DlpPolicyIdParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Force <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example imports the DLP policy collection in the file C:\My Documents\DLP Backup.xml.

```
Import-DlpPolicyCollection -FileData ([Byte[]]$(Get-Content
-Path " C:\My Documents\DLP Backup.xml " -Encoding Byte -
ReadCount 0))
```

Detailed Description

The **Import-DlpPolicyCollection** cmdlet imports all the settings of the DLP policies and the associated transport rules. You use the **Export-DlpPolicyCollection** cmdlet to export the DLP policy collection.

Caution:
Importing a DLP policy collection from an XML file removes or overwrites all pre-existing DLP policies that were defined in your organization. Make sure that you have a backup of your current DLP policy collection before you import and overwrite your current DLP policies.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
FileData	Required	System.Byte[]	The *FileData* parameter

			specifies the DLP policy collection file you want to import. For more information about the syntax required to use this parameter, see Syntax.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.A	The *Force* switch

		utomation.SwitchParameter	specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.CompliancePrograms.Tasks.DlpPolicyIdParameter	The *Identity* parameter specifies the name of the DLP policy you want to import. The DLP policy must exist in the XML file you specify with the *FileData* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchPara	The *WhatIf* switch instructs the command

		meter	to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DlpPolicyTemplate

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-DlpPolicyTemplate** cmdlet to view existing data loss prevention (DLP) policy templates in your Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-DlpPolicyTemplate [-Identity <DlpPolicyIdParameter>] [-
DomainController <Fqdn>]
```

Examples

Example 1

This example returns a summary list of all DLP policy templates.

```
Get-DlpPolicyTemplate
```

Example 2

This example returns detailed information about the DLP policy template named GLBA. The command is piped to the **Format-List** cmdlet to display the detailed configuration of the specified DLP policy template.

```
Get-DlpPolicyTemplate GLBA | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Identity	Optional	Microsoft.Exchange.MessagingPolicies.CompliancePrograms.Tasks.DlpPolicyIdParameter	The *Identity* parameter specifies the DLP policy template you want to view. You can use any value that uniquely identifies the DLP policy template. For example, you can specify the name, GUID, or distinguished name (DN) of the DLP policy template.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Import-DlpPolicyTemplate

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Import-DlpPolicyTemplate** cmdlet to import a data loss prevention (DLP) policy template file into your Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Import-DlpPolicyTemplate -FileData <Byte[]> [-Confirm [<SwitchParameter>]]
[-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example imports the DLP policy template file C:\My Documents\External DLP Policy Template.xml.

```
Import-DlpPolicyTemplate -FileData ([Byte[]]$(Get-Content -
Path "C:\My Documents\External DLP Policy Template.xml" -
Encoding Byte -ReadCount 0))
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
FileData	Required	System.Byte[]	The *FileData* parameter specifies the DLP policy template file you want to import. For more information about the syntax required to use this parameter, see Syntax.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing

			continues. You don't have to specify a value with the *Confirm* switch.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-DlpPolicyTemplate

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-DlpPolicyTemplate** cmdlet to remove a data loss prevention (DLP) policy template from your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-DlpPolicyTemplate [-Identity <DlpPolicyIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example removes the DLP policy template named External DLP Policy Template.

```
Remove-DlpPolicyTemplate "External DLP Policy Template"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.CompliancePrograms.Tasks.DlpPolicyIdParameter	The *Identity* parameter specifies the DLP policy template you want to remove. You can use any value that uniquely identifies the DLP policy template. For example, you can specify the name, GUID, or distinguished name (DN) of the DLP policy template.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-Fingerprint

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-09*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-Fingerprint** cmdlet to create document fingerprints that are used with data classification rules. Because the results of **New-Fingerprint** are not stored outside of the data classification rule, you always run **New-Fingerprint** and **New-DataClassification** or **Set-Dataclassification** in the same PowerShell session.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-Fingerprint -Description <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-FileData <Byte[]>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example creates a new document fingerprint based on the file C:\My Documents\Contoso Patent Template.docx. You store the new fingerprint as a variable so you can use it with the **New-DataClassification** cmdlet in the same PowerShell session.

```
$Patent_Template = Get-Content "C:\My Documents\Contoso
```

```
Patent Template.docx" -Encoding byte
$Patent_Fingerprint = New-Fingerprint -FileData
$Patent_Template -Description "Contoso Patent Template"
```

Detailed Description

Classification rule packages are used by data loss prevention (DLP) to detect sensitive content in messages.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Description	Required	System.String	The *Description* parameter specifies a description for the document fingerprint.
FileData	Required	System.Byte[]	The *FileData* parameter specifies the file to use as a document fingerprint. You need to read the file to a byte-encoded object using the **Get-Content** cmdlet. For details, see the Examples section.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-IRMConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-IRMConfiguration** cmdlet to view the Information Rights Management (IRM) configuration in a Microsoft Exchange Server 2013 organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-IRMConfiguration [-Identity <OrganizationIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves the IRM configuration in an Exchange 2013 organization.

```
Get-IRMConfiguration
```

Detailed Description

The **Get-IRMConfiguration** cmdlet provides details about the current IRM configuration, including whether individual IRM features are enabled or disabled, and provides the URLs used for ServiceLocation, PublishingLocation, and LicensingLocation.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-IRMConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-IRMConfiguration** cmdlet to configure Information Rights Management (IRM) features.

For information about the parameter sets in the Syntax section below, see Syntax.

Important:
Configuring and using IRM features in an on-premises Microsoft Exchange Server 2013 deployment requires Active Directory Rights Management Services (AD RMS).

```
Set-IRMConfiguration [-Identity <OrganizationIdParameter>] [-
ClientAccessServerEnabled <$true | $false>] [-Confirm [<SwitchParameter>]]
[-DomainController <Fqdn>] [-EDiscoverySuperUserEnabled <$true | $false>]
[-ExternalLicensingEnabled <$true | $false>] [-Force <SwitchParameter>] [-
InternalLicensingEnabled <$true | $false>] [-
JournalReportDecryptionEnabled <$true | $false>] [-LicensingLocation
<MultiValuedProperty>] [-PublishingLocation <Uri>] [-
RefreshServerCertificates <SwitchParameter>] [-RMSOnlineKeySharingLocation
<Uri>] [-SearchEnabled <$true | $false>] [-ServiceLocation <Uri>] [-
TransportDecryptionSetting <Disabled | Optional | Mandatory>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables journal report decryption.

```
Set-IRMConfiguration -JournalReportDecryptionEnabled $true
```

EXAMPLE 2

This example enables transport decryption and enforces decryption. When decryption is enforced, messages that can't be decrypted are rejected, and an NDR is returned.

```
Set-IRMConfiguration -TransportDecryptionSetting Mandatory
```

EXAMPLE 3

This example enables licensing for external messages.

```
Set-IRMConfiguration -ExternalLicensingEnabled $true
```

Detailed Description

IRM requires the use of an on-premises AD RMS server or the ILS service. IRM features can be selectively enabled or disabled.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
ClientAccessServerEnabled	Optional	System.Boolean	The *ClientAccessServerEnabled* parameter specifies whether to enable IRM in Microsoft Office Outlook Web App and in Microsoft Exchange ActiveSync. Both of these features are enabled by default. To disable them, set the parameter to `$false`. **Important:** Enabling IRM in Outlook Web App requires additional configuration on AD RMS servers. For more information, see Information Rights Management in Outlook Web App.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EDiscoverySuperUserEnabled	Optional	System.Boolean	The *EDiscoverySuperUserEnabled* parameter specifies whether members of the Discovery Management role group can access IRM-protected messages that were returned by a discovery search and are residing in a discovery mailbox. To enable IRM-

			protected message access to the Discovery Management role group, set the value to `$true`. For more information about In-Place eDiscovery and IRM-protected messages, see In-Place eDiscovery.
ExternalLicensingEnabled	Optional	System.Boolean	The *ExternalLicensingEnabled* parameter specifies whether to enable IRM features for messages sent to external recipients. In on-premises deployments, licensing is disabled for external messages by default. To enable licensing, set the value to `$true`.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress the confirmation prompt that appears when you modify the *InternalLicensingEnabled* parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter is reserved for internal Microsoft use.
InternalLicensingEnabl	Optional	System.Boolean	The

ed			*InternalLicensingEnabled* parameter specifies whether to enable IRM features for messages sent to internal recipients. In on-premises deployments, licensing is disabled for internal messages by default. To enable licensing, set the value to `$true`. **Note:** If the *InternalLicensingEnabled* parameter is set to `$false`, no AD RMS templates are returned when you use the **Get-RMSTemplate** cmdlet.
JournalReportDecryptionEnabled	Optional	System.Boolean	The *JournalReportDecryptionEnabled* parameter specifies whether to enable journal report decryption. When enabled, journal report decryption attaches a decrypted copy of an IRM-protected message to the journal report. Journal report decryption is enabled by default. To disable journal report decryption, set the value to `$false`.

			Important: Enabling journal report decryption requires additional configuration on AD RMS servers. For more information, see Journal report decryption.
LicensingLocation	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *LicensingLocation* parameter specifies one or more additional AD RMS licensing URLs in on-premises deployments. It isn't required to populate this parameter if the organization doesn't have cross-forest deployment of licensing servers.
PublishingLocation	Optional	System.Uri	This parameter is available only in the cloud-based service. The *PublishingLocation* parameter specifies one or more AD RMS publishing URLs.
RefreshServerCertificates	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RefreshServerCertificates* switch clears all Rights

			Account Certificates (RACs), Computer Licensor Certificates (CLCs), and cached AD RMS templates from all Microsoft Exchange Server 2010 or Exchange Server 2013 servers in the organization. Clearing RACs, CLCs, and cached templates may be required during troubleshooting or in the event of a change of keys on the AD RMS cluster in your organization. For more information about RACs and CLCs, see Understanding AD RMS Certificates.
RMSOnlineKeySharingLocation	Optional	System.Uri	This parameter is available only in the cloud-based service. The *RMSOnlineKeySharingLocation* parameter specifies the RMS Online URL to obtain the trusted publishing domain (TPD) for the Microsoft Exchange Online organization.

SearchEnabled	Optional	System.Boolean	The *SearchEnabled* parameter specifies whether to enable searching of IRM-encrypted messages in Outlook Web App. Valid values include: • `$true` (default) Enables search of IRM-encrypted messages in Outlook Web App. • `$false` Disables search of IRM-encrypted messages in Outlook Web App.
ServiceLocation	Optional	System.Uri	This parameter is available only in the cloud-based service. The *ServiceLocation* parameter specifies the AD RMS service URL.
TransportDecryptionSetting	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.TransportDecryptionSetting	The *TransportDecryptionSetting* parameter specifies the transport decryption configuration. Valid values include one of the following: • `Disabled` Transport decryption is disabled for internal and external messages. • `Mandatory` Messages that can't be decrypted are rejected, and a non-delivery report (NDR) is

			returned. • `Optional` A best effort approach to decryption is provided. Messages are decrypted if possible, but delivered even if decryption fails.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-IRMConfiguration

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Test-IRMConfiguration** cmdlet to test Information Rights Management (IRM) configuration and functionality.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-IRMConfiguration [-Identity <OrganizationIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Recipient <SmtpAddress[]
>] [-RMSOnline <SwitchParameter>] [-RMSOnlineAuthCertSubjectNameOverride
<String>] [-RMSOnlineOrgOverride <Guid>] [-Sender <SmtpAddress>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests the IRM configuration for messages sent from the sender adams@contoso.com.

```
Test-IRMConfiguration -Sender adams@contoso.com
```

Detailed Description

The **Test-IRMConfiguration** cmdlet performs a series of steps to test IRM configuration and functionality, including availability of an Active Directory Rights Management Services (AD RMS) server, prelicensing, and journal report decryption. In Exchange Online organizations, it checks connectivity to RMS Online and obtains and validates the organization's Trusted Publishing Domain (TPD).

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is available only in on-premises Exchange 2013. The *Identity* parameter is reserved for internal Microsoft use.
Recipient	Optional	Microsoft.Exchange.Data.SmtpAddress[]	The *Recipient* parameter specifies the SMTP address of one or more recipients. The cmdlet tests prelicensing for the specified recipients.

			You can specify multiple recipient addresses separated by commas. If no recipient is specified, the sender address is used as the recipient.
RMSOnline	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *RMSOnline* switch specifies whether to test connectivity from Exchange Online to RMS Online, obtain your Exchange Online organization's TPD, and test its validity.
RMSOnlineAuthCertSubjectNameOverride	Optional	System.String	This parameter is reserved for internal Microsoft use.
RMSOnlineOrgOverride	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
Sender	Optional	Microsoft.Exchange.Data.SmtpAddress	The *Sender* parameter specifies the SMTP address of the sender to be tested. The cmdlet tests prelicensing and

			journal report decryption for the sender.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-JournalRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Disable-JournalRule** cmdlet to disable a journal rule on a Mailbox server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-Journalrule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-LawfulInterception
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the journal rule Brokerage Communications.

```
Disable-JournalRule "Brokerage Communications"
```

EXAMPLE 2

This example disables all journal rules. The Get-JournalRule cmdlet is used to add all journal rules to the pipeline. The results are piped to the **Disable-JournalRule** cmdlet.

```
Get-JournalRule | Disable-JournalRule
```

Detailed Description

You can enable or disable specific journal rules in your organization at any time using the **Disable-JournalRule** and Enable-JournalRule cmdlets.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Journaling" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the journal rule you want to disable. Enter either the name or the GUID of the journal rule. You

			can omit this parameter label.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
LawfulInterception	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-JournalRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-JournalRule** cmdlet to enable an existing journal rule on a Mailbox server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-Journalrule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-LawfulInterception
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the existing journal rule Brokerage Communications.

```
Enable-JournalRule "Brokerage Communications"
```

Detailed Description

You can enable or disable specific journal rules in your organization at any time using the **Enable-JournalRule** and Disable-JournalRule cmdlets. To learn more about journaling in Microsoft Exchange Server 2013, see Journaling.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Journaling" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the journal rule you want to enable. Enter either the name or GUID of the journal rule. You can omit this parameter label.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing

			continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
LawfulInterception	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-JournalRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-JournalRule** cmdlet to view the journal rules in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-JournalRule [-Identity <RuleIdParameter>] [-DomainController <Fqdn>]
[-LawfulInterception <SwitchParameter>] [-Organization
<OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves all journal rules configured in your organization.

```
Get-JournalRule
```

EXAMPLE 2

This example retrieves the specific journal rule Brokerage Communications and pipes the output to the **Format-List** cmdlet to view all the parameters of the rule.

```
Get-JournalRule "Brokerage Communications" | Format-List
```

Detailed Description

The **Get-JournalRule** cmdlet displays journal rules configured in your organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Journaling" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the rule you want to view. Enter either the name or the GUID of the journal rule. You can omit the parameter label.
LawfulInterception	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-JournalRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-JournalRule** cmdlet to create a journal rule in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-Journalrule -JournalEmailAddress <RecipientIdParameter> -Name <String>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true
| $false>] [-ExpiryDate <DateTime>] [-FullReport <$true | $false>] [-
LawfulInterception <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-Recipient <SmtpAddress>] [-Scope <Internal |
External | Global>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates and enables a journal rule. The rule applies to all email messages that pass through the Transport service and contain at least one recipient or sender who is a member of the brokers@contoso.com distribution list.

```
New-JournalRule -Name "Brokerage Communications" -
JournalEmailAddress "Brokers Journal Mailbox" -Scope Global
-Recipient brokers@contoso.com -Enabled $true
```

Detailed Description

The **New-JournalRule** cmdlet creates a journal rule in your organization.

By default, new journal rules are disabled unless the *Enabled* parameter is set to `$true`. For more information about how to enable a new journal rule that was created in a disabled state, see Enable-JournalRule.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Journaling" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
JournalEmailAddress	Required	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *JournalEmailAddress* parameter specifies a recipient object to which journal reports are sent.
Name	Required	System.String	The *Name* parameter specifies the name of the journal rule. The name of the rule can be up to 64 characters long.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the journal rule is enabled or disabled. If the rule is disabled, it isn't applied to any email messages. The default value is `$false`.
ExpiryDate	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.
FullReport	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LawfulInterception	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
Recipient	Optional	Microsoft.Exchange.Data.SmtpAddress	The *Recipient* parameter specifies the SMTP

			address of a mailbox, contact, or distribution group to journal. If you specify a distribution group, all recipients in that distribution group are journaled. All messages sent to or from a recipient are journaled.
Scope	Optional	Microsoft.Exchange.MessagingPolicies.Journaling.JournalRuleScope	The *Scope* parameter specifies the scope of email messages to which the journal rule is applied. Valid values for this parameter are as follows: • `Global` Global rules process all email messages that pass through a Transport service. This includes email messages that were already processed by the external and internal rules. The default value is `Global`. • `Internal` Internal rules process email messages sent and received by recipients in your organization. • `External` External rules process email messages sent to recipients or from senders outside your organization.
WhatIf	Optional	System.Management.Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-JournalRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-JournalRule** cmdlet to remove an existing journal rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-Journalrule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-LawfulInterception
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the journal rule Brokerage Communications that's no longer needed.

```
Remove-JournalRule "Brokerage Communications"
```

Detailed Description

The **Remove-JournalRule** cmdlet removes the specified journal rule from Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Journaling" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the rule you want to remove. Enter either name or the GUID of the journal rule. You can omit the parameter label.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value

			with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
LawfulInterception	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-JournalRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-JournalRule** cmdlet to modify an existing journal rule in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-JournalRule -Identity <RuleIdParameter> [-Confirm [<SwitchParameter>]]
[-DomainController <Fqdn>] [-ExpiryDate <DateTime>] [-FullReport <$true |
$false>] [-JournalEmailAddress <RecipientIdParameter>] [-
LawfulInterception <SwitchParameter>] [-Name <String>] [-Recipient
<SmtpAddress>] [-Scope <Internal | External | Global>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the journal email address to which journal reports are sent by the existing journal rule Consolidated Messenger.

```
Set-JournalRule "Consolidated Messenger" -
JournalEmailAddress "ArchiveMailbox@contoso.com"
```

EXAMPLE 2

This example modifies the journal email address for all journal rules. The **Get-JournalRule** cmdlet is used to retrieve all journal rules. The results are piped to the **Set-JournalRule** cmdlet to modify

the journal recipient.

```
Get-JournalRule | Set-JournalRule -JournalEmailAddress
"Archive Mailbox"
```

Detailed Description

The **Set-JournalRule** cmdlet modifies an existing journal rule used in your organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Journaling" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the name or GUID of the rule you want to modify. The *Identity* parameter is a positional parameter. When using positional parameters in a command, you can omit the parameter label.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExpiryDate	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.
FullReport	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
JournalEmailAddress	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *JournalEmailAddress* parameter specifies a journal recipient. Journal reports for the specified rule are sent to the journal recipient.
LawfulInterception	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Name	Optional	System.String	The *Name* parameter specifies the name of the journal rule. The name of the rule can be up to 64 characters.

Recipient	Optional	Microsoft.Exchange.Data.SmtpAddress	The *Recipient* parameter specifies the SMTP address of a mailbox, contact, or distribution group to journal. If you specify a distribution group, all recipients in that distribution group are journaled. All messages sent to or received from a recipient are journaled.
Scope	Optional	Microsoft.Exchange.MessagingPolicies.Journaling.JournalRuleScope	The *Scope* parameter specifies the scope of email messages to which the journal rule is applied. You can use the following values: • `Global` Global rules process all email messages that pass through a Transport service. This includes email messages that were already processed by the external and internal rules. • `Internal` Internal rules process email messages sent to and received by recipients in your organization. • `External` External rules process email messages sent to recipients or from senders outside your organization.
WhatIf	Optional	System.Management.	The *WhatIf* switch

		Automation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-JournalRuleCollection

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-20*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Export-JournalRuleCollection** cmdlet to export the journal rules in your organization to an XML file.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Export-JournalRuleCollection [-Identity <RuleIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example exports journal rules in a two-step process. In the first step, the **Export-JournalRuleCollection** cmdlet exports journal rules to the variable `$file`. In the second step, the **Set-Content** cmdlet saves the exported data to the XML file JournalRules.xml.

```
$file = Export-JournalRuleCollection
Set-Content -Path "C:\MyDocs\JournalRules.xml" -Value
$file.FileData -Encoding Byte
```

Detailed Description

You can use the **Export-JournalRuleCollection** cmdlet to export journal rules in your organization to create a backup copy of your rules.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Journaling" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Dat	The *DomainController*

		a.Fqdn	parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the name of a journal rule.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Import-JournalRuleCollection

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Import-JournalRuleCollection** cmdlet to import journal rules from an XML file. You can import a journal rule collection you previously exported as a backup, or import rules you exported from an older version of Exchange.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Import-JournalRuleCollection -FileData <Byte[]> [-Identity
<RuleIdParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example imports journal rules from the XML file ExportedJournalRules.xml in a two-step process.

The first step retrieves journal rules from the previously exported XML file ExportedJournalRules.xml using the **Get-Content** cmdlet, and then stores the results in the variable *$Data*. The second step retrieves data from the variable *$Data* and imports journal rules to your organization, overwriting existing journal rules.

```
[Byte[]]$Data = Get-Content -Path "C:\JournalRules
\ExportedJournalRules.xml" -Encoding Byte -ReadCount 0
Import-JournalRuleCollection -FileData $Data
```

Detailed Description

The **Import-JournalRuleCollection** cmdlet imports a journal rule collection you previously

exported.

Caution:
Importing a journal rule collection from an XML file removes or overwrites all pre-existing journal rules in your organization. Make sure that you have a backup of your current journal rule collection before you import and overwrite your current journal rules.

Importing file data is a two-step process. First you must load the data to a variable using the **Get-Content** cmdlet, and then use that variable to transmit the data to the cmdlet.

For more information about how to export a journal rule collection to an XML file, see Export-JournalRuleCollection.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Journaling" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
FileData	Required	System.Byte[]	The *FileData* parameter specifies the variable name that contains the content of the XML file. The content is retrieved using the **Get-Content** cmdlet in the first step of the two-step process used to import file content.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -

			`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the name of a journal rule to be imported.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxAuditBypassAssociation

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxAuditBypassAssociation** cmdlet to retrieve user or computer accounts configured to bypass mailbox audit logging.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxAuditBypassAssociation [-Identity
<MailboxAuditBypassAssociationIdParameter>] [-DomainController <Fqdn>] [-
Organization <OrganizationIdParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example retrieves all user or computer accounts configured for mailbox audit logging bypass.

```
Get-MailboxAuditBypassAssociation -ResultSize unlimited
```

EXAMPLE 2

This example retrieves the mailbox audit bypass association for the Svc-MyApplication account.

```
Get-MailboxAuditBypassAssociation -Identity "Svc-
MyApplication"
```

Detailed Description

When you configure a user or computer account to bypass mailbox audit logging, access or actions taken by the user or computer account to any mailbox isn't logged. By bypassing trusted user or computer accounts that need to access mailboxes frequently, you can reduce the noise in mailbox audit logs.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox audit logging" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxAuditBypassAssociationIdParameter	The *Identity* parameter specifies a user or computer account to retrieve audit logging bypass association for.
Organization	Optional	Microsoft.Exchange.Co	The *Organization*

		nfiguration.Tasks.OrganizationIdParameter	parameter is reserved for internal Microsoft use.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all mailboxes that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxAuditBypassAssociation

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailboxAuditBypassAssociation** cmdlet to configure mailbox audit logging bypass for user or computer accounts such as service accounts for applications that access mailboxes

frequently.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxAuditBypassAssociation -Identity
<MailboxAuditBypassAssociationIdParameter> -AuditBypassEnabled <$true |
$false> [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example bypasses the Svc-MyApplication account from mailbox audit logging.

```
Set-MailboxAuditBypassAssociation -Identity "Svc-
MyApplication" -AuditBypassEnabled $true
```

EXAMPLE 2

This example removes the bypass association for the Svc-MyApplication account.

```
Set-MailboxAuditBypassAssociation -Identity "Svc-
MyApplication" -AuditBypassEnabled $false
```

Detailed Description

When you configure a user or computer account to bypass mailbox audit logging, access or actions taken by the user or computer account to any mailbox isn't logged. By bypassing trusted user or computer accounts that need to access mailboxes frequently, you can reduce the noise in mailbox audit logs.

Caution:
If you use mailbox audit logging to audit mailbox access and actions, you must monitor mailbox audit bypass associations at regular intervals. If a mailbox audit bypass association is added for an account, the account can access any mailbox in the organization to which it has been assigned access permissions, without any mailbox audit logging entries being generated for such access, or any actions taken such as message deletions.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox audit logging" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description

AuditBypassEnabled	Required	System.Boolean	The *AuditBypassEnabled* parameter specifies whether audit bypass is enabled for the user or computer. Valid values include the following: • `$true` Enables mailbox audit logging bypass • `$false` Disables mailbox audit logging bypass
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxAuditBypassAssociationIdParameter	The *Identity* parameter specifies a user or computer account to be bypassed from mailbox audit logging.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Search-MailboxAuditLog

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Search-MailboxAuditLog** cmdlet to search mailbox audit log entries matching the specified search terms.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Search-MailboxAuditLog [-EndDate <ExDateTime>] [-ExternalAccess <$true |
$false>] [-LogonTypes <MultiValuedProperty>] [-Mailboxes
<MultiValuedProperty>] [-ResultSize <Int32>] [-StartDate <ExDateTime>]
<COMMON PARAMETERS>


Search-MailboxAuditLog [-EndDate <ExDateTime>] [-ExternalAccess <$true |
$false>] [-Identity <MailboxIdParameter>] [-LogonTypes
<MultiValuedProperty>] [-Organization <OrganizationIdParameter>] [-
ResultSize <Int32>] [-ShowDetails <SwitchParameter>] [-StartDate
<ExDateTime>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves mailbox audit log entries for Ken Kwok's mailbox for actions performed by `Admin` and `Delegate` logon types between 1/1/2012 and 12/31/2012. A maximum of 2,000 log entries are returned.

```
Search-MailboxAuditLog -Identity kwok -LogonTypes
Admin,Delegate -StartDate 1/1/2012 -EndDate 12/31/2012 -
ResultSize 2000
```

EXAMPLE 2

This example retrieves mailbox audit log entries for Ken Kwok and Ben Smith's mailboxes for actions performed by `Admin` and `Delegate` logon types between 1/1/2012 and 12/31/2012. A maximum of 2,000 log entries are returned.

```
Search-MailboxAuditLog -Mailboxes kwok,bsmith -LogonTypes
Admin,Delegate -StartDate 1/1/2012 -EndDate 12/31/2012 -
ResultSize 2000
```

EXAMPLE 3

This example retrieves mailbox audit log entries for Ken Kwok's mailbox for actions performed by the mailbox owner between 1/1/2012 and 3/1/2012. The results are piped to the **Where-Object** cmdlet and filtered to return only entries with the **HardDelete** action.

```
Search-MailboxAuditLog -Identity kwok -LogonTypes Owner -
```

```
ShowDetails -StartDate 1/1/2012 -EndDate 3/1/2012 | Where-
Object {$_.Operation -eq "HardDelete"}
```

Detailed Description

The **Search-MailboxAuditLog** cmdlet performs a synchronous search of mailbox audit logs for one or more specified mailboxes and displays search results in the Exchange Management Shell window. To search mailbox audit logs for multiple mailboxes and have the results sent by email to specified recipients, use the **New-MailboxAuditLogSearch** cmdlet instead. To learn more about mailbox audit logging, see Mailbox audit logging.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox audit logging" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
EndDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *EndDate* parameter specifies the end date of the date range. Use the short date format defined in the **Regional Options** settings for the computer on which the

			command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
ExternalAccess	Optional	System.Boolean	The *ExternalAccess* parameter returns only mailbox audit log entries for mailbox access by users outside of your organization. In Exchange Online, use this parameter to return audit log entries for mailbox access by Microsoft datacenter administrators.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the mailbox for which to retrieve mailbox audit log entries. You can use this parameter to search a single mailbox.

LogonTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *LogonTypes* parameter specifies the type of logons. Valid values include: • `Admin` Audit log entries for mailbox access by administrator logons are returned. • `Delegate` Audit log entries for mailbox access by delegates are returned, including access by users with Full Mailbox Access permission. • `External` For Exchange Online mailboxes, audit log entries for mailbox access by Microsoft datacenter administrators are returned. • `Owner` Audit log entries for mailbox access by the primary mailbox owner are returned. If you specify the `Owner` logon type, you must use the *ShowDetails* switch.
Mailboxes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Mailboxes* parameter specifies the mailboxes for which to retrieve mailbox audit log entries. You can use this parameter to search audit logs for multiple mailboxes. You can't use the *ShowDetails*

			switch with the *Mailboxes* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ResultSize	Optional	System.Int32	The *ResultSize* parameter specifies the maximum number of mailbox audit log entries to return. Valid values include an integer from 1 through 250000. By default, 1000 entries are returned.
ShowDetails	Optional	System.Management.Automation.SwitchParameter	The *ShowDetails* switch specifies that details of each log entry be retrieved. By default, all fields for each returned log entry are displayed in a list view. You can't use the *Mailboxes* parameter with the *ShowDetails* switch.
StartDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *StartDate* parameter specifies the start date of the date range. Use the short date format defined in the **Regional Options** settings for the computer on which the

			command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailboxAuditLogSearch

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-16*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MailboxAuditLogSearch** cmdlet to search mailbox audit logs and have search results sent via email to specified recipients.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MailboxAuditLogSearch -EndDate <ExDateTime> -StartDate <ExDateTime> -
StatusMailRecipients <MultiValuedProperty> [-Organization
<OrganizationIdParameter>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-ExternalAccess <$true | $false>] [-LogonTypes
<MultiValuedProperty>] [-Mailboxes <MultiValuedProperty>] [-Name <String>]
[-ShowDetails <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a mailbox audit log search to search Ken Kwok and April Stewart's mailboxes for administrator and delegate logons from 1/1/2011 to 12/31/2011. Search results are sent to auditors@contoso.com by email.

```
New-MailboxAuditLogSearch "Admin and Delegate Access" -
Mailboxes "Ken Kwok","April Stewart" -LogonTypes
Admin,Delegate -StartDate 1/1/2011 -EndDate 12/31/2011 -
StatusMailRecipients auditors@contoso.com
```

EXAMPLE 2

This example returns entries from the mailbox audit logs of all users in organization for any mailbox access by Microsoft datacenter administrators between September 1, 2013 and October 24, 2013. The search results are sent to admin@contoso.com.

```
New-MailboxAuditLogSearch -ExternalAccess $true -StartDate
09/01/2013 -EndDate 10/24/2013 -StatusMailRecipients
admin@contoso.com
```

Detailed Description

The **New-MailboxAuditLogSearch** cmdlet performs an asynchronous search of mailbox audit logs for the specified mailboxes and sends the search results by email to the specified recipients. The body of the email message contains search metadata such as search parameters and the time when the search request was submitted. The results are attached in an .xml file.

To search mailbox audit logs for a single mailbox and have the results displayed in the Exchange Management Shell window, use the Search-MailboxAuditLog cmdlet instead. To learn more about mailbox audit logging, see Mailbox audit logging.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox audit logging" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
EndDate	Required	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *EndDate* parameter specifies the end date of the date range. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
StartDate	Required	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *StartDate* parameter specifies the start date of the date range. Use the short date format

			defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for example, **"10/05/2010 5:00 PM"**.
StatusMailRecipients	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *StatusMailRecipients* parameter specifies the email address of one or more recipients to whom search results are sent by email.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a

			value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExternalAccess	Optional	System.Boolean	The *ExternalAccess* parameter returns only audit log entries for mailbox access by a user outside of your organization. In Exchange Online, use this parameter to return audit log entries for access to a mailbox by Microsoft datacenter administrators.
LogonTypes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *LogonTypes* parameter specifies the type of logons. Valid values include: • `Admin` Audit log entries for mailbox access by administrator logons are returned. • `Delegate` Audit log

			entries for mailbox access by delegates are returned, including access by users with Full Mailbox Access permission. • `External` For cloud-based mailboxes, audit log entries for mailbox access by administrators of the cloud-based service are returned. • `Owner` Audit log entries for mailbox access by the primary mailbox owner are returned. If you specify the `Owner` logon type, you must use the *ShowDetails* switch. You can specify multiple values separated by a comma.
Mailboxes	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Mailboxes* parameter specifies one or more mailboxes for which to retrieve mailbox audit log entries. If you don't specify a value, mailbox audit logs for all mailboxes on Microsoft Exchange Server 2013 or Exchange Server 2010 in the Exchange organization are returned.
Name	Optional	System.String	The *Name* parameter specifies a name for the

			search.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ShowDetails	Optional	System.Management.Automation.SwitchParameter	The *ShowDetails* switch specifies that details of each log entry be retrieved.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailboxSearch

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-22*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailboxSearch** cmdlet to view mailbox searches that are in progress, complete, or stopped.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailboxSearch -InPlaceHoldIdentity <String> <COMMON PARAMETERS>

Get-MailboxSearch [-Identity <EwsStoreObjectIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-ResultSize <Unlimited>] [-
ShowDeletionInProgressSearches <SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves a list of all mailbox searches.

```
Get-MailboxSearch -ResultSize "unlimited"
```

EXAMPLE 2

This example retrieves all properties for the mailbox search Project Hamilton.

```
Get-MailboxSearch "Project Hamilton" | FL
```

Note:
The *Identity* parameter is a positional parameter. Positional parameters can be used without the label (Identity). For more information about positional parameters, see Parameters.

EXAMPLE 3

This example retrieves the In-Place Holds that a user is placed on. The following command outputs GUIDs of In-Place Holds.

```
(Get-Mailbox Mark).InPlaceHolds
```

The following command retrieves a mailbox search based on the GUID of the In-Place Hold that the user is placed on.

```
Get-MailboxSearch -InPlaceHoldIdentity
9953d0f0fd03415e949d4b41c5a28cbb
```

Detailed Description

In Microsoft Exchange Server 2013 and Exchange Online, a mailbox search is used to perform an In-Place eDiscovery or to place users on an In-Place Hold. Use the **Get-MailboxSearch** cmdlet to retrieve details of either type of mailbox search.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
InPlaceHoldIdentity	Required	System.String	The *InPlaceHoldIdentity* parameter specifies the GUID of an In-Place Hold. Use this parameter to search for an In-Place Hold that a user is placed on. GUIDs of all In-Place Holds that a user is placed on are added to the user's **InPlaceHolds** property. You can retrieve the property by using the **Get-Mailbox** cmdlet.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.EwsStoreObjectIdParameter	The *Identity* parameter specifies the name of the search query. If a name isn't provided, all mailbox search queries are returned. To improve the performance of this cmdlet in Exchange Online, some mailbox search properties aren't returned if you don't specify the name of a mailbox search. These properties are: • SourceMailboxes • Sources • SearchQuery • ResultsLink • PreviewResultsLink • Errors To view these properties, you have to provide the name of a mailbox search.

Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. The default result size is 1000. Use unlimited to return all mailbox searches.
ShowDeletionInProgressSearches	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailboxSearch

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-23*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MailboxSearch** cmdlet to create a mailbox search and either get an estimate of search results, place search results on In-Place Hold or copy them to a Discovery mailbox. You can

also place all contents in a mailbox on hold by not specifying a search query, which accomplishes similar results as litigation hold in Microsoft Exchange Server 2010.

Caution:

Mailbox searches are performed across all mailboxes on Microsoft Exchange Server 2013 servers in an Exchange organization, unless the search is constrained to fewer mailboxes by using the *SourceMailboxes* parameter.

Important:

When you create a mailbox search using this cmdlet on an Exchange 2013 server, mailboxes on previous versions of Exchange aren't searched. You must search mailboxes on Exchange Server 2010 by running the command on an Exchange 2010 server.

For more information, see In-Place eDiscovery and In-Place Hold.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MailboxSearch -Name <String> [-AllPublicFolderSources <$true |
$false>] [-AllSourceMailboxes <$true | $false>] [-Confirm
[<SwitchParameter>]] [-Description <String>] [-DomainController <Fqdn>] [-
EndDate <ExDateTime>] [-EstimateOnly <SwitchParameter>] [-
ExcludeDuplicateMessages <$true | $false>] [-Force <SwitchParameter>] [-
IncludeKeywordStatistics <SwitchParameter>] [-IncludeUnsearchableItems
<SwitchParameter>] [-InPlaceHoldEnabled <$true | $false>] [-
InPlaceHoldIdentity <String>] [-ItemHoldPeriod <Unlimited>] [-Language
<CultureInfo>] [-LogLevel <Suppress | Basic | Full>] [-MessageTypes
<KindKeyword[]>] [-PublicFolderSources <PublicFolderIdParameter[]>] [-
Recipients <String[]>] [-SearchQuery <String>] [-Senders <String[]>] [-
SourceMailboxes <RecipientIdParameter[]>] [-StartDate <ExDateTime>] [-
StatusMailRecipients <RecipientIdParameter[]>] [-TargetMailbox
<MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the mailbox search Legal-ProjectX. The search uses several parameters to restrict the query:

- *SourceMailboxes* This parameter restricts the search to members of the DG-Marketing and DG-Executives distribution groups.
- *Recipients* This parameter specifies that the search includes all mail sent to the domain contoso.com.
- *SearchQuery* This parameter specifies a KQL query for messages with either the words project or report and for messages with attachments.
- *StartDate* and *EndDate* These parameters specify the start date of January 1, 2011, and end date of December 31, 2011, for the search.
- *TargetMailbox* This parameter specifies that search results should be copied to the discovery mailbox LegalDiscovery.
- *StatusMailRecipeints* This parameter specifies that the distribution group DG-DiscoveryTeam is to receive a notification when the search is complete.

```
New-MailboxSearch -Name "Legal-ProjectX" -SourceMailboxes
DG-Marketing,DG-Executives -TargetMailbox
LegalDiscovery@contoso.com -StartDate "01/01/2011" -EndDate
"12/31/2011" -Recipients "*@contoso.com" -SearchQuery
"project report hasattachments:true" -StatusMailRecipients
"DG-DiscoveryTeam"
```

EXAMPLE 2

This example creates an In-Place Hold Hold-ProjectX and places all members of the distribution group DG-Finance on hold. Because the search doesn't specify the *SearchQuery* and *ItemHoldPeriod* parameters, all messages in mailboxes returned are placed on indefinite In-Place Hold.

```
New-MailboxSearch -Name "Hold-ProjectX" -SourceMailboxes
DG-Finance -InPlaceHoldEnabled $true
```

EXAMPLE 3

This example creates an In-Place Hold Hold-tailspintoys and places all members of the distribution group DG-Research on hold. Because the search specifies the *SearchQuery* parameter, only messages that match the search query are placed on indefinite In-Place Hold.

```
New-MailboxSearch -Name "Hold-tailspintoys" -
SourceMailboxes DG-Research -SearchQuery "'Patent' AND
'Project tailspintoys'" -InPlaceHoldEnabled $true
```

Detailed Description

The **New-MailboxSearch** cmdlet creates an In-Place eDiscovery search or an In-Place Hold. Unless specified, mailboxes on all Exchange 2013 servers in an organization are searched. You can stop, start, modify, or remove the search.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "In-Place eDiscovery" and "In-Place Hold" entries in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies a friendly name

			for the search. Search results are copied to a folder in the mailbox specified by the *TargetMailbox* parameter. The folder name is the same as the search name.
AllPublicFolderSources	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
AllSourceMailboxes	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies a description for the search. The description isn't displayed to users.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EndDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *EndDate* parameter specifies an end date and time for the search. Messages dated on or before the end date will be searched. The default time is the current time. When you enter a specific date, use the short date format, mm/dd/yyyy, even if the Regional Options settings on the local computer are configured with a different format, such as dd/mm/yyyy. For example, use **03/01/2012** to specify March 1, 2012. You can enter the date only, for example, **10/05/2011**. Or you can enter the date and time of day. If you enter a time of day and date, you must enclose the argument in

			quotation marks ("), for example, **"10/05/2011 5:00:00 PM"**.
EstimateOnly	Optional	System.Management.Automation.SwitchParameter	The *EstimateOnly* switch specifies that only an estimate of the number of items that will be returned is provided. If not specified, messages are copied to the target mailbox.
ExcludeDuplicateMessages	Optional	System.Boolean	The *ExcludeDuplicateMessages* parameter eliminates duplication of messages in search results. Set the parameter to `$true` to copy a single instance of a message if the same message exists in multiple folders or mailboxes.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* parameter creates the search, overwriting any existing searches with the same name.
IncludeKeywordStatistics	Optional	System.Management.Automation.SwitchParameter	The *IncludeKeywordStatistics* switch returns keyword statistics (number of instances for each

			keyword) in search results.
IncludeUnsearchableItems	Optional	System.Management.Automation.SwitchParameter	The *IncludeUnsearchableItems* switch specifies whether items that couldn't be indexed by Exchange Search should be included in the search results. The *IncludeUnsearchableItems* parameter doesn't require a value. **Important:** In Exchange 2013, specifying this switch doesn't result in unsearchable items being placed on hold. To place unsearchable items on hold, create a search without specifying search parameters, which accomplishes the same results as litigation hold in Exchange 2010.
InPlaceHoldEnabled	Optional	System.Boolean	The *InPlaceHoldEnabled* parameter specifies whether an In-Place Hold has been placed on items matching the search query. Set the parameter to `$true` to enable In-Place Hold. If the *ItemHoldPeriod* parameter isn't specified,

			items are held until the hold is removed by deleting the search or removing a mailbox from the search. You can add or remove mailboxes from a mailbox search by modifying the *SourceMailboxes* parameter. If you don't specify a search query, all items in the specified mailboxes are placed on hold.
InPlaceHoldIdentity	Optional	System.String	This parameter is reserved for internal Microsoft use.
ItemHoldPeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *ItemHoldPeriod* parameter specifies the number of days for which to hold mailbox items matching the search query. The duration is calculated from the time the item is received or created in the mailbox.
Language	Optional	System.Globalization.CultureInfo	The *Language* parameter specifies a locale for the search.
LogLevel	Optional	Microsoft.Exchange.Data.Storage.Infoworker.MailboxSearch.Loggin	The *LogLevel* parameter specifies the logging level for the search. It can have

		gLevel	one of the following values: • `Suppress` No logs are kept. • `Basic` Basic information about the query and who ran it is kept. • `Full` In addition to the information kept by the `Basic` log level, the `Full` log level adds a complete list of search results.
MessageTypes	Optional	Microsoft.Exchange.Data.Search.AqsParser.KindKeyword[]	The *MessageTypes* parameter specifies the message types to include in the search. Valid values can be one or more of the following: • `Email` • `Meetings` • `Tasks` • `Notes` • `Docs` • `Journals` • `Contacts` • `IM` If not specified, all message types are included.
PublicFolderSources	Optional	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter[]	This parameter is reserved for internal Microsoft use.
Recipients	Optional	System.String[]	The *Recipients* parameter specifies one or more recipients. Messages that have the recipients in the

			To, **Cc**, and **Bcc** fields are returned.
SearchQuery	Optional	System.String	The *SearchQuery* parameter specifies a search string or a query formatted using Keyword Query Language (KQL). If this parameter is empty, all messages from all mailboxes specified in the *SourceMailboxes* parameter are returned.
Senders	Optional	System.String[]	The *Senders* parameter specifies the SMTP address of one or more senders.
SourceMailboxes	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SourceMailboxes* parameter specifies the identity of one or more mailboxes to be searched. **Note:** If not specified, all mailboxes in the Exchange 2013 organization are searched. To enable In-Place Hold, you must specify the *SourceMailboxes* parameter.
StartDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *StartDate* parameter specifies a start date and time for the search.

			For valid date and time formatting options, refer to the description of the *EndDate* parameter.
StatusMailRecipients	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *StatusMailRecipients* parameter specifies one or more recipients to receive a status email message upon completion of the search.
TargetMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *TargetMailbox* parameter specifies the identity of the destination mailbox where search results are copied. You can use the following values: • Alias • Display name • *Domain\Account* • SMTP address • Distinguished name (DN) • Object GUID • User principal name (UPN) • LegacyExchangeDN
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailboxSearch

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MailboxSearch** cmdlet to remove a mailbox search.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MailboxSearch -Identity <EwsStoreObjectIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the mailbox search Project Contoso.

```
Remove-MailboxSearch -Identity "Project Contoso"
```

Detailed Description

In Microsoft Exchange Server 2013, mailbox searches are used for In-Place eDiscovery and In-Place Hold. You can't remove an In-Place Hold without first disabling the hold.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.EwsStoreObjectIdParameter	The *Identity* parameter specifies the name of the mailbox search.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailboxSearch

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-23*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailboxSearch** cmdlet to modify an existing mailbox search.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailboxSearch -Identity <EwsStoreObjectIdParameter> [-
AllPublicFolderSources <$true | $false>] [-AllSourceMailboxes <$true |
$false>] [-Confirm [<SwitchParameter>]] [-Description <String>] [-
DomainController <Fqdn>] [-EndDate <ExDateTime>] [-EstimateOnly <$true |
$false>] [-ExcludeDuplicateMessages <$true | $false>] [-Force
<SwitchParameter>] [-IncludeKeywordStatistics <SwitchParameter>] [-
IncludeUnsearchableItems <$true | $false>] [-InPlaceHoldEnabled <$true |
$false>] [-ItemHoldPeriod <Unlimited>] [-Language <String>] [-LogLevel
<Suppress | Basic | Full>] [-MessageTypes <KindKeyword[]>] [-Name
<String>] [-PublicFolderSources <PublicFolderIdParameter[]>] [-Recipients
<String[]>] [-SearchQuery <String>] [-Senders <String[]>] [-
SourceMailboxes <RecipientIdParameter[]>] [-StartDate <ExDateTime>] [-
StatisticsStartIndex <Int32>] [-StatusMailRecipients
<RecipientIdParameter[]>] [-TargetMailbox <MailboxIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the start date of a mailbox search.

```
Set-MailboxSearch -Identity "Legal-ProjectX" -StartDate
"01/01/2010"
```

Detailed Description

In Microsoft Exchange Server 2013, mailbox searches are used for In-Place eDiscovery and In-Place Hold. For In-Place eDiscovery, unless specified, mailboxes on all Exchange 2013 Mailbox servers in an organization are searched. To create an In-Place Hold, you must specify the mailboxes to place on hold using the *SourceMailboxes* parameter. The search can be stopped, started, modified, and removed.

Important:

When you create a mailbox search using this cmdlet on an Exchange 2013 server, mailboxes on previous versions of Exchange aren't searched. You must search mailboxes on Exchange Server 2010 by running the command on an Exchange 2010 server.

Caution:

Mailbox searches are performed across all Exchange 2013 servers in an Exchange organization, unless the search is constrained to fewer mailboxes by using the *SourceMailboxes* parameter.

If the In-Place eDiscovery search you want to modify is running, stop it before using the **Set-MailboxSearch** cmdlet. When restarting a search, any previous search results are removed from the target mailbox.

For more information, see In-Place eDiscovery and In-Place Hold.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "In-Place eDiscovery" and "In-Place Hold" entries in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.EwsStoreObjectIdParameter	The *Identity* parameter specifies the name of the mailbox search.
AllPublicFolderSources	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
AllSourceMailboxes	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Description	Optional	System.String	The *Description* parameter specifies a description for the search.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EndDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *EndDate* parameter specifies an end date for the search.
EstimateOnly	Optional	System.Boolean	The *EstimateOnly* parameter specifies that only an estimate of the number of items that will be returned is provided. Messages aren't copied to the target mailbox.
ExcludeDuplicateMessages	Optional	System.Boolean	The *ExcludeDuplicateMessages* parameter eliminates duplication of messages across mailboxes in an In-Place eDiscovery search.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch suppresses the confirmation prompt displayed before

			modifying a search. When modifying a search, previous search results are removed from the target mailbox, and the search is restarted after modification. The *Force* switch doesn't require a value.
IncludeKeywordStatistics	Optional	System.Management.Automation.SwitchParameter	The *IncludeKeywordStatistics* switch specifies that the command generates and returns keyword statistics for the search.
IncludeUnsearchableItems	Optional	System.Boolean	The *IncludeUnsearchableItems* parameter specifies whether unsearchable items are included in search results. If set to `$true`, unsearchable items are included in the search results. **Caution:** In Exchange 2013, unsearchable items aren't placed on hold for a query-based In-Place Hold. If you need to place unsearchable items on hold, you must create an indefinite hold (a hold without specifying any search parameters, which

			provides functionality similar to litigation hold in Exchange 2010).
InPlaceHoldEnabled	Optional	System.Boolean	The *InPlaceHoldEnabled* parameter specifies whether an In-Place Hold should be placed on items matching the search query. Set the parameter to `$true` to enable In-Place Hold. If the *ItemHoldPeriod* parameter isn't specified, items are held until the hold is removed by deleting the search or removing a mailbox from the search. You can add or remove mailboxes from a mailbox search by modifying the *SourceMailboxes* parameter. If you don't specify a search query, all specified mailboxes are placed on hold. **Important:** If you attempt to place a hold but don't specify mailboxes using the *SourceMailboxes* parameter, the command may succeed but the mailboxes are not placed on In-Place Hold.

ItemHoldPeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *ItemHoldPeriod* parameter specifies the number of days for which to hold mailbox items matching the search query. The duration is calculated from the time the item is received or created in the mailbox.
Language	Optional	System.String	The *Language* parameter specifies a locale for the mailbox search.
LogLevel	Optional	Microsoft.Exchange.Data.Storage.Infoworker.MailboxSearch.LoggingLevel	The *LogLevel* parameter specifies a logging level for the mailbox search. Valid values are: • `Basic` Basic details of the search are kept. • `Full` In addition to details in the Basic logging level, a full list of all messages returned is included. • `Suppress` Logging is suppressed. No logs are kept.
MessageTypes	Optional	Microsoft.Exchange.Data.Search.AqsParser.KindKeyword[]	The *MessageTypes* parameter specifies the message types that should be included in the mailbox search. Valid values include: • `Email` • `Meetings` • `Tasks` • `Notes`

			• `Docs` • `Journals` • `Contacts` • `IM`
Name	Optional	System.String	The *Name* parameter specifies a name for the search. The top-level folder created in the target mailbox, where items returned by the search are copied, is also named after the search name.
PublicFolderSources	Optional	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter[]	This parameter is reserved for internal Microsoft use.
Recipients	Optional	System.String[]	The *Recipients* parameter specifies one or more recipients and is a part of the mailbox search query. Messages addressed to any recipient specified in the *Recipients* parameter are returned.
SearchQuery	Optional	System.String	The *SearchQuery* parameter specifies a search query using Keyword Query Language (KQL). If a query isn't specified, the entire mailbox is copied to the target mailbox. If other search parameters

			such as *Senders*, *Recipients*, *StartDate*, and *EndDate* are specified, these are combined by using the **AND** operator with the *SearchQuery* parameter.
Senders	Optional	System.String[]	The *Senders* parameter specifies one or more senders. Messages sent by the specified senders are returned by the search. Senders can include users, distribution groups, SMTP addresses, or domains. If distribution groups are specified, messages sent by distribution group members are returned in the search results.
SourceMailboxes	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SourceMailboxes* parameter specifies the mailboxes to be searched. If no mailboxes are explicitly specified by using the *SourceMailboxes* parameter, all mailboxes located on Exchange 2013 servers across the entire organization are searched.

StartDate	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *StartDate* parameter specifies a start date for the mailbox search. Messages dated on or after the start date are returned by the search.
StatisticsStartIndex	Optional	System.Int32	The *StatisticsStartIndex* parameter is used by the Exchange Administration Center (EAC) to retrieve keyword statistics in a paged operation.
StatusMailRecipients	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *StatusMailRecipients* parameter specifies one or more recipients to receive a status email message upon completion of the search.
TargetMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *TargetMailbox* parameter specifies the mailbox to which items returned by the search are copied.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply

			any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Start-MailboxSearch

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Start-MailboxSearch** cmdlet to restart or resume a mailbox search that's been stopped.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Start-MailboxSearch -Identity <EwsStoreObjectIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-Resume <SwitchParameter>] [-StatisticsStartIndex <Int32>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example starts the mailbox search ProjectContoso.

```
Start-MailboxSearch -Identity "ProjectContoso"
```

Detailed Description

You can use In-Place eDiscovery to search one or more specified mailboxes or all mailboxes across the Microsoft Exchange Server 2013 organization. A search is created by using the Exchange Administration Center (EAC) or the **New-MailboxSearch** cmdlet.

Caution:
When restarting a search, any previous results returned by the same search and copied to a Discovery mailbox are removed. To preserve previous search results and resume the search from the point it was stopped, use the *Resume* switch.

In Exchange 2013, mailbox searches are also used for In-Place Hold. However, you can't start or stop In-Place Hold using the **Start-MailboxSearch** and **Stop-MailboxSearch** cmdlets.

For more details, see In-Place Hold and In-Place eDiscovery.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.EwsStoreObjectIdParameter	The *Identity* parameter specifies the name of the search. The name is referenced when starting, stopping, or removing the search.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value

			with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch suppresses the confirmation prompt displayed before the command is executed.
Resume	Optional	System.Management.Automation.SwitchParameter	The *Resume* switch resumes a stopped, failed, or partially succeeded search from the point it stopped. If you use the *Resume* switch to resume a search, previous search results aren't removed from the target mailbox.
StatisticsStartIndex	Optional	System.Int32	The *StatisticsStartIndex*

			parameter is used by the Exchange Administration Center (EAC) to retrieve keyword statistics in a paged operation.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Stop-MailboxSearch

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Stop-MailboxSearch** cmdlet to stop a mailbox search that's in progress.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Stop-MailboxSearch -Identity <EwsStoreObjectIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example stops the mailbox search Project Contoso.

```
Stop-MailboxSearch -Identity "Project Contoso"
```

Detailed Description

In Microsoft Exchange Server 2013, mailbox searches are used for In-Place eDiscovery and In-Place Hold. You can start and stop a mailbox search. For more information, see In-Place eDiscovery.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "In-Place eDiscovery" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.EwsStoreObjectIdParameter	The *Identity* parameter specifies the name of the mailbox search.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Start-ManagedFolderAssistant

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Start-ManagedFolderAssistant** cmdlet to immediately start messaging records management (MRM) processing of mailboxes that you specify.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Start-ManagedFolderAssistant -Identity <MailboxOrMailUserIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
EHAHiddenFolderCleanup <SwitchParameter>] [-HoldCleanup <SwitchParameter>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example processes the mailbox for a user with the alias Chris.

```
Start-ManagedFolderAssistant -Identity "Chris"
```

EXAMPLE 2

This example uses the **Get-Mailbox** command to retrieve all the mailboxes that resolve from the ambiguous name resolution (ANR) search on the string "Chr" in the domain DC01 (for example, users such as Chris Ashton, Christian Hess, and Christa Geller), and the results are piped to the **Start-ManagedFolderAssistant** cmdlet for processing.

```
Get-Mailbox -Anr Chr -DomainController DC01 | Start-
ManagedFolderAssistant
```

Detailed Description

The Managed Folder Assistant uses the retention policy settings of users' mailboxes to process retention of items. This mailbox processing occurs automatically. You can use the **Start-ManagedFolderAssistant** cmdlet to immediately start processing the specified mailbox.

Important:
In the Microsoft Exchange Server 2010 release to manufacturing (RTM) and Exchange Server 2007 versions, the *Identity* parameter specifies the Mailbox server to start the assistant and process all mailboxes on that server, and the *Mailbox* parameter specifies the mailbox to process. In Exchange 2010 Service Pack 1 (SP1) and later, the *Mailbox* parameter has been removed, and the *Identity* parameter accepts the mailbox or mail user to process. If you use these parameters in scheduled commands or scripts, we recommend that you review them and make any necessary changes.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxOrMailUserIdParameter	The *Identity* parameter specifies the mailbox to be processed. In cross-premises deployments, you can also specify a mail user who has a mailbox in the cloud.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EHAHiddenFolderCleanup	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
HoldCleanup	Optional	System.Management.Automation.SwitchParameter	The *HoldCleanup* switch instructs the Managed Folder Assistant to clean up duplicate versions of items in the Recoverable Items folder that may have been created when a mailbox is on In-Place

			Hold, litigation hold, or has Single Item Recovery enabled. Removing duplicate items from the Recoverable Items folder reduces the folder size and may help prevent reaching Recoverable Items quota limits. For more details about Recoverable Items quota limits, see Recoverable Items folder.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Stop-ManagedFolderAssistant

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Stop-ManagedFolderAssistant** cmdlet to immediately stop messaging records management (MRM) from processing users' mailboxes on the specified servers.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Stop-ManagedFolderAssistant [-Identity <ServerIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example executes the **Stop-ManagedFolderAssistant** command without parameters. The Managed Folder Assistant is stopped as soon as processing of the current mailbox is completed on the current server.

```
Stop-ManagedFolderAssistant
```

EXAMPLE 2

This example stops the Managed Folder Assistant on the servers ExchSrvr1 and Exchsrvr2.

```
Stop-ManagedFolderAssistant -Identity ExchSrvr1, Exchsrvr2
```

Detailed Description

The Managed Folder Assistant uses the managed folder mailbox policy settings of users to process

mailbox items for retention and journaling as needed. Use the **Stop-ManagedFolderAssistant** cmdlet to stop the Managed Folder Assistant as soon as processing of the current mailbox is completed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the names of servers on which the Managed Folder

			Assistant is to be stopped. If a server isn't specified, the Managed Folder Assistant on the local server is stopped.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MessageClassification

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MessageClassification** cmdlet to view existing message classifications in your organization.

```
Get-MessageClassification [-Identity <MessageClassificationIdParameter>]
[-DomainController <Fqdn>] [-IgnoreDehydratedFlag <SwitchParameter>] [-
IncludeLocales <SwitchParameter>] [-Organization
<OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example lists all message classifications in your organization.

```
Get-MessageClassification
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Message classifications" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

			The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MessageClassificationIdParameter	The *Identity* parameter specifies the name of the message classification instance that you want to view. When you specify a name that includes spaces, you must enclose the name in quotation marks ("), for example, `"Administrative Name"`.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IncludeLocales	Optional	System.Management.Automation.SwitchParameter	The *IncludeLocales* switch specifies whether the command output includes the message classification locale data. When you

			use the *IncludeLocales* switch, the output includes the message classification locale data.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MessageClassification

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MessageClassification** cmdlet to create a message classification instance in your organization.

```
New-MessageClassification [-ClassificationID <Guid>] [-DisplayPrecedence
<Highest | Higher | High | MediumHigh | Medium | MediumLow | Low | Lower |
Lowest>] [-PermissionMenuVisible <$true | $false>] [-
RetainClassificationEnabled <$true | $false>] <COMMON PARAMETERS>

New-MessageClassification -Locale <CultureInfo> <COMMON PARAMETERS>
```

```
COMMON PARAMETERS: -DisplayName <String> -Name <String> -SenderDescription
<String> [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
IgnoreDehydratedFlag <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-RecipientDescription <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the message classification named `MyMessageClassification` with the following properties:

- The display name is `New Message Classification`.
- The sender description is "`This is the description text`".

```
New-MessageClassification -Name MyMessageClassification -
DisplayName "New Message Classification" -SenderDescription
"This is the description text"
```

EXAMPLE 2

This example creates a locale-specific (Spanish - Spain) version of an existing message classification `MyMessageClassification`.

```
New-MessageClassification MyMessageClassification -Locale
es-ES -DisplayName "España Example" -SenderDescription
"Este es el texto de la descripción"
```

Detailed Description

After you create a new message classification, you can specify the message classification as a transport rule predicate. Before Microsoft Outlook and Outlook Web App users can apply the message classification to messages, you must update the end-user systems with the message classification XML file created by the `Export-OutlookClassification.ps1` script file. The `Export-OutlookClassification.ps1` script file is located in the `%ExchangeInstallPath%Scripts` directory.

When you create a message classification, it has no locale. By default, the new message classification is used for all locales. After a default message classification is defined, you can add new locales of the classification by running the **New-MessageClassification** cmdlet and by specifying the default message classification identity that you want to localize.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Message classifications" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
DisplayName	Required	System.String	The *DisplayName* parameter specifies the display name for the message classification instance. The display name is used by Outlook users to select the appropriate message classification before they send a message. **Note:** The message classification XML file must be present on the sender's computer for the display name to be displayed. If the *UserDisplayEnabled* parameter is set to `$true`, the display name is displayed for the recipient, even if no message classification XML file is installed. When you specify a name that includes spaces, you must enclose the name in quotation marks ("), for

			example, `"Display Name"`. The *DisplayName* parameter can contain a maximum of 64 characters.
Locale	Required	System.Globalization.CultureInfo	The *Locale* parameter specifies a locale-specific version of the message classification. You must also pass the *Identity* parameter of the default existing message classification when you create a new locale-specific version. Valid input for the *Locale* parameter is the string names listed in the *Culture Name* column in the Microsoft .NET Class Library class reference available at CultureInfo Class.
Name	Required	System.String	The *Name* parameter specifies the administrative name for the message classification instance. The name is used to administer the message classification instance.

			When you specify a name that includes spaces, you must enclose the name in quotation marks ("), for example, `"Adminstrative Name"`. The *Name* parameter can contain a maximum of 256 characters.
SenderDescription	Required	System.String	The *SenderDescription* parameter specifies the purpose of the message classification to the sender. The value of this parameter is used by Outlook users to select the appropriate message classification before they send a message. Enclose the value in quotation marks ("), for example, `"This is the sender description that explains when to use this message classification"`. The *SenderDescription* parameter can contain a maximum of 1,024 characters.
ClassificationID	Optional	System.Guid	The *ClassificationID* parameter specifies a

			classification ID of an existing message classification that you want to import and use in your Exchange organization. Use this parameter if you're configuring message classifications that span two Exchange forests in the same enterprise.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisplayPrecedence	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ClassificationDisplayPrecedenceLevel	The *DisplayPrecedence* parameter specifies the relative precedence of the message classification to other message classifications that may be applied to a specified message. Although Outlook only lets a user specify a single classification per

			message, transport rules may apply other classifications to a message. The classification with the highest precedence is shown first, and the subsequent classifications, which are those with lesser precedence as defined by this parameter, are appended in the appropriate order thereafter. Valid input for the *DisplayPrecedence* parameter is `Highest`, `Higher`, `High`, `MediumHigh`, `Medium`, `MediumLow`, `Low`, `Lower`, and `Lowest`. The default value is `Medium`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
PermissionMenuVisible	Optional	System.Boolean	The *PermissionMenuVisible* parameter specifies whether the values that you entered for the *DisplayName* and *RecipientDescription* parameters are displayed in Outlook as the user composes a

			message. If you set the *PermissionMenuVisible* parameter to `$false`, users won't be able to assign this message classification to the messages they are composing. However, messages received with this message classification still display the classification information. The default value is `$true`.
RecipientDescription	Optional	System.String	The *RecipientDescription* parameter specifies the purpose of the message classification to the recipient. The value of this parameter is shown to Outlook users when they receive a message that has this message classification. Enclose the value in quotation marks ("), for example, `"This is the recipient description that explains how to treat the message that has been`

			`classified"`. The *RecipientDescription* parameter can contain a maximum of 1,024 characters. If you don't enter a value for this parameter, the description that you enter for the *SenderDescription* parameter is used.
RetainClassificationEnabled	Optional	System.Boolean	The *RetainClassificationEnabled* parameter specifies whether the message classification should persist with the message if the message is forwarded or replied to. The default value is `$true`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of

			those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MessageClassification

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MessageClassification** cmdlet to delete an existing message classification instance from your organization.

```
Remove-MessageClassification -Identity <MessageClassificationIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the message classification named MyMessageClassification.

```
Remove-MessageClassification MyMessageClassification
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Message classifications" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MessageClassificationIdParameter	The *Identity* parameter specifies the name of the message classification instance that you want to remove. When you specify a name that includes spaces, you must enclose the name in quotation marks ("), for example, `"Administrative Name"`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MessageClassification

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MessageClassification** cmdlet to configure an existing message classification instance in your organization.

```
Set-MessageClassification -Identity <MessageClassificationIdParameter> [-
ClassificationID <Guid>] [-Confirm [<SwitchParameter>]] [-DisplayName
<String>] [-DisplayPrecedence <Highest | Higher | High | MediumHigh |
Medium | MediumLow | Low | Lower | Lowest>] [-DomainController <Fqdn>] [-
Name <String>] [-PermissionMenuVisible <$true | $false>] [-
RecipientDescription <String>] [-RetainClassificationEnabled <$true |
$false>] [-SenderDescription <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example makes the following configuration changes to the message classification named MyMessageClassification:

- Changes the display precedence to Low.
- Specifies that the message classification shouldn't persist with the message if the message is

forwarded or replied to.

```
Set-MessageClassification MyMessageClassification -
DisplayPrecedence Low -RetainClassificationEnabled $false
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Message classifications" entry in the Mail flow permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MessageClassificationIdParameter	The *Identity* parameter specifies the name or GUID of the message classification you want to modify.
ClassificationID	Optional	System.Guid	The *ClassificationID* parameter specifies the GUID of an existing message classification that you want to use in your Exchange organization. Use this parameter if you're configuring message classifications to span two Exchange forests in the same organization.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the display name for the message classification instance. The display name appears in the Microsoft Office and is used by Outlook users to select the appropriate message classification before they send a message. When you specify a name that includes spaces, you must enclose the name in quotation marks ("), for example, `"Display Name"`. The *DisplayName* parameter can contain a maximum of 64 characters.
DisplayPrecedence	Optional	Microsoft.Exchange.Dat	The *DisplayPrecedence*

		a.Directory.SystemConfiguration.ClassificationDisplayPrecedenceLevel	parameter specifies the relative precedence of the message classification to other message classifications that may be applied to a specified message. Although Outlook only lets a user specify a single classification for each message, transport rules may apply other classifications to a message. The classification with the highest precedence is shown first, and the subsequent classifications, which are those with lesser precedence as defined by this parameter, are appended in the appropriate order thereafter. Valid input for the *DisplayPrecedence* parameter is `Highest`, `Higher`, `High`, `MediumHigh`, `Medium`, `MediumLow`, `Low`, `Lower`, and `Lowest`.

			The default value is `Medium`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Name	Optional	System.String	The *Name* parameter specifies the administrative name for the message classification instance. The name is used to administer the

			message classification instance. When you specify a name that includes spaces, you must enclose the name in quotation marks ("), for example, `"Administrative Name"`. The *Name* parameter can contain a maximum of 256 characters.
PermissionMenuVisible	Optional	System.Boolean	The *PermissionMenuVisible* parameter specifies whether the values that you entered for the *DisplayName* and *RecipientDescription* parameters are displayed in Outlook as the user composes a message. If you set the *PermissionMenuVisible* parameter to `$false`, users won't be able to assign this message classification to the messages they're composing. However, messages received with this message

			classification still display the classification information. The default value is `$true`.
RecipientDescription	Optional	System.String	The *RecipientDescription* parameter specifies the purpose of the message classification to the recipient. The value of this parameter is shown to Outlook users when they receive a message that has this message classification. Enclose the value in quotation marks ("), for example, `"This is the recipient description that explains how to treat the message that has been classified"`. The *RecipientDescription* parameter can contain a maximum of 1,024 characters. If you don't enter a value for this parameter, the

			description that you enter for *SenderDescription* is used.
RetainClassificationEnabled	Optional	System.Boolean	The *RetainClassificationEnabled* parameter specifies whether the message classification should persist with the message if the message is forwarded or replied to. The default value is `$true`.
SenderDescription	Optional	System.String	The *SenderDescription* parameter specifies the purpose of the message classification to the sender. The value of this parameter is used by Outlook users to select the appropriate message classification before they send a message. Enclose the value in quotation marks ("), for example, `"This is the sender description that explains when to use this message classification"`. The

			SenderDescription parameter can contain a maximum of 1,024 characters.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-OutlookProtectionRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-OutlookProtectionRule** cmdlet to disable an existing Microsoft Outlook protection rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-OutlookProtectionRule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the Outlook protection rule Project Contoso.

```
Disable-OutlookProtectionRule -Identity "Project Contoso"
```

Detailed Description

Outlook protection rules are administrator-created rules applied before a user sends a message using Outlook. Outlook protection rules are used to automatically Information Rights Management (IRM)-protect email messages using a Rights Management Services (RMS) template before the message is sent. However, Outlook protection rules don't inspect message content. To rights-protect messages based on message content, use transport protection rules.

For more information, see Outlook protection rules.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the name of the rule being disabled.
Confirm	Optional	System.Management.A	The *Confirm* switch

		utomation.SwitchParameter	causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-OutlookProtectionRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-OutlookProtectionRule** cmdlet to enable an existing Outlook protection rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-OutlookProtectionRule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the Outlook protection rule Project Contoso.

```
Enable-OutlookProtectionRule -Identity "Project Contoso"
```

Detailed Description

Outlook protection rules are used to automatically Information Rights Management (IRM)-protect messages using a Rights Management Services (RMS) template before the message is sent. However, Outlook protection rules don't inspect message content. To rights-protect messages based on message content, use transport protection rules. For more information, see Outlook protection rules.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the name of the rule being enabled.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-OutlookProtectionRule

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-OutlookProtectionRule** cmdlet to retrieve Microsoft Outlook protection rules configured in an organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-OutlookProtectionRule [-Identity <RuleIdParameter>] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example gets details of all Outlook protection rules configured in the organization.

```
Get-OutlookProtectionRule
```

EXAMPLE 2

This example gets all properties of the Outlook protection rule ProjectContoso.

```
Get-OutlookProtectionRule ProjectContoso | Format-List
```

Note:

The *Identity* parameter is positional. When used after the cmdlet name, the parameter value can be specified without providing the parameter label.

Detailed Description

Outlook protection rules are used to automatically Information Rights Management (IRM)-protect email messages using a Rights Management Services (RMS) template before the message is sent. However, Outlook protection rules don't inspect message content. To IRM-protect messages based on message content, use transport protection rules. For more information, see Outlook protection rules.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter identifies an Outlook protection rule.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-OutlookProtectionRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-OutlookProtectionRule** cmdlet to create a Microsoft Outlook protection rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-OutlookProtectionRule -ApplyRightsProtectionTemplate
<RmsTemplateIdParameter> -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Enabled <$true | $false>] [-Force
<SwitchParameter>] [-FromDepartment <String[]>] [-Organization
<OrganizationIdParameter>] [-Priority <Int32>] [-SentTo
<RecipientIdParameter[]>] [-SentToScope <All | InOrganization>] [-
UserCanOverride <$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example applies the AD RMS template `Template-Contoso` to messages sent to the SMTP address Joe@contoso.com.

```
New-OutlookProtectionRule -Name "Project Contoso" -SentTo
Joe@contoso.com -ApplyRightsProtectionTemplate "Template-
Contoso"
```

Detailed Description

Outlook protection rules are administrator-created rules applied before a user sends a message using Outlook. Outlook inspects message content and protects messages by applying Active Directory Rights Management Services (AD RMS) rights templates.

For more information, see Outlook protection rules.

Caution:
Outlook protection rules created without a condition apply to all messages.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the

"Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
ApplyRightsProtectionTemplate	Required	Microsoft.Exchange.Configuration.Tasks.RmsTemplateIdParameter	The *ApplyRightsProtectionTemplate* parameter specifies the AD RMS template to apply to the message. An AD RMS template can be specified using the template name. Use the **Get-RMSTemplate** cmdlet to retrieve templates from your AD RMS server.
Name	Required	System.String	The *Name* parameter specifies a name for the rule.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Da	This parameter is available only in on-

		ta.Fqdn	premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the Outlook protection rules are enabled. New Outlook protection rules are enabled by default. To create a rule without enabling it, set the *Enabled* parameter to `$false`.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress the confirmation prompt used to warn the administrator when rules are created without any conditions. Rules without any conditions specified apply to all messages. The *Force* switch doesn't require a value.
FromDepartment	Optional	System.String[]	The *FromDepartment*

			parameter specifies the sender's department as a condition. The user's department property is compared with the value specified.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Priority	Optional	System.Int32	The *Priority* parameter specifies whether to set the order of processing of Outlook protection rules. Rules with a lower priority value are executed first.
SentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentTo* parameter specifies one or more recipients as a rule condition. The identity of recipients in the organization or the SMTP address of external recipients can be specified. The *SentTo* parameter doesn't accept wildcard characters. When multiple recipients are specified, messages sent to any of the specified recipients are considered a match.

SentToScope	Optional	Microsoft.Exchange.Management.OutlookProtectionRules.ToUserScope	The *SentToScope* parameter specifies the message scope as a condition. You can use one of the following values: • `InOrganization` This value is for messages sent to recipients within the organization. • `All` This value is for messages sent to all recipients.
UserCanOverride	Optional	System.Boolean	The *UserCanOverride* parameter specifies whether users can override actions taken by Outlook protection rules. By default, users can override actions taken by Outlook protection rules. When set to `$false`, the *UserCanOverride* parameter prevents the user from overriding the rule when sending the message.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-OutlookProtectionRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-OutlookProtectionRule** cmdlet to remove an Outlook protection rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-OutlookProtectionRule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the Outlook protection rule Project Contoso.

```
Remove-OutlookProtectionRule -Identity "Project Contoso"
```

EXAMPLE 2

This example removes all Outlook protection rules in the organization. The **Get-OutlookProtectionRule** cmdlet is used to retrieve all Outlook protection rules in the Microsoft Exchange Server 2013 organization, and the results are pipelined to the **Remove-OutlookProtectionRule** cmdlet to remove them.

```
Get-OutlookProtectionRule | Remove-OutlookProtectionRule
```

Detailed Description

Outlook protection rules use an Active Directory Rights Management Services (AD RMS) rights template to automatically apply Information Rights Management (IRM) protection to messages before they're sent. For more information, see Outlook protection rules.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the name of the rule being removed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-OutlookProtectionRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-OutlookProtectionRule** cmdlet to modify an existing Microsoft Outlook protection rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-OutlookProtectionRule -Identity <RuleIdParameter> [-
ApplyRightsProtectionTemplate <RmsTemplateIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-FromDepartment <String[]>] [-Name <String>] [-Priority <Int32>] [-SentTo
<MultiValuedProperty>] [-SentToScope <All | InOrganization>] [-
UserCanOverride <$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the Outlook protection rule `OPR-DG-Finance` to apply to messages sent to the DG-Finance distribution group.

```
Set-OutlookProtectionRule -Identity "OPR-DG-Finance" -
SentTo "DG-Finance"
```

EXAMPLE 2

This example sets the priority of the Outlook protection rule `OPR-DG-Finance` to 2.

```
Set-OutlookProtectionRule -Identity "OPR-DG-Finance" -
Priority 2
```

Detailed Description

Outlook protection rules are used to automatically rights-protect email messages using a Rights Management Services (RMS) template before the message is sent. However, Outlook protection rules don't inspect message content. To rights-protect messages based on message content, use transport protection rules. For more information, see Outlook protection rules.

Caution:
Not specifying any conditions results in an Outlook protection rule being applied to all messages.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the rule.
ApplyRightsProtectionTemplate	Optional	Microsoft.Exchange.Configuration.Tasks.RmsTemplateIdParameter	The *ApplyRightsProtectionTemplate* parameter specifies an RMS template to be applied to messages matching the conditions.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -

			`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress the confirmation prompt produced by the cmdlet when modifying a rule with no conditions, resulting in such rules being applied to all messages.
FromDepartment	Optional	System.String[]	The *FromDepartment* parameter specifies a department name. The rule is applied to messages where the sender's department attribute matches this value.

Name	Optional	System.String	The *Name* parameter specifies a name for the rule.
Priority	Optional	System.Int32	The *Priority* parameter specifies a priority for the Outlook protection rule. Rule priority values can range from 0 through *n*-1, where *n* is the total number of existing Outlook protection rules. Any existing rules with priority equal to or higher than the priority being set have their priority incremented by 1.
SentTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SentTo* parameter specifies one or more recipients. External recipients can be specified using the SMTP address. Internal recipients can be specified using any of the following values: • Alias • Distinguished name (DN) • ExchangeGUID • LegacyExchangeDN • SmtpAddress • User principal name (UPN)

SentToScope	Optional	Microsoft.Exchange.Management.OutlookProtectionRules.ToUserScope	The *SentToScope* parameter specifies the scope of messages to which the rule applies. Valid values include: • `All` Applies to all messages. • `InOrganization` Applies to messages originating from inside the Exchange organization, where all recipients are also internal. If not specified, the parameter defaults to `All`.
UserCanOverride	Optional	System.Boolean	The *UserCanOverride* parameter specifies whether the Outlook user can override the rule behavior, either by using a different RMS template, or by removing rights protection before sending the message. Valid values include: • `$true` User can override rule action. • `$false` User can't override rule action.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PolicyTipConfig

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-PolicyTipConfig** cmdlet to view the data loss prevention (DLP) Policy Tips in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PolicyTipConfig [-Action <NotifyOnly | RejectOverride | Reject | Url>]
[-Locale <CultureInfo>] [-Original <SwitchParameter>] <COMMON PARAMETERS>


Get-PolicyTipConfig [-Identity <PolicyTipConfigIdParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>]
```

Examples

Example 1

This example returns a summary list of the custom Policy Tips in all languages that have the action value `NotifyOnly`.

```
Get-PolicyTipConfig -Action NotifyOnly
```

Example 2

This example returns a summary list of all built-in French Policy Tips.

```
Get-PolicyTipConfig -Original -Locale fr
```

Example 3

This example returns details about the custom English Policy Tip for the action value `RejectOverride`.

```
Get-PolicyTipConfig en\RejectOverride | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Action	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.PolicyTipMessageConfigAction	The *Action* parameter filters the Policy Tips by action. Valid values for this parameter are: • NotifyOnly • RejectOverride • Reject You can't use the value

			`Url` with the *Action* parameter. Instead, use command: `Get-PolicyTipConfig Url`. You can't use the *Action* parameter with the *Identity* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Management.PolicyNudges.PolicyTipConfigIdParameter	The *Identity* parameter specifies the custom Policy Tip you want to view. You can use any value that uniquely identifies the custom Policy Tip. For example: • <*Locale*>\<*Action*>: Locale is a supported locale code. For example, `en` for English or `fr` for French. For more information about supported locales, see Supported languages

			for system messages. Action is one of the following Policy Tip actions: `NotifyOnly`, `RejectOverride` or `Reject`. • The value `Url` • GUID • Distinguished name (DN) You can't use the *Identity* parameter with the *Action*, *Locale*, or *Original* parameters.
Locale	Optional	System.Globalization.CultureInfo	The *Locale* parameter specifies a locale-specific version of the Policy Tip. Valid values for this parameter are supported locale codes. For example, `en` for English or `fr` for French. For more information about supported locales, see Supported languages for system messages. You can't use the *Locale* parameter with the *Identity* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.Orga	The *Organization* parameter is reserved for

		nizationIdParameter	internal Microsoft use.
Original	Optional	System.Management.Automation.SwitchParameter	The *Original* switch includes built-in Policy Tips in the results. You don't specify a value with the *Original* switch. You can't use the *Original* switch with the *Identity* parameter.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-PolicyTipConfig

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-PolicyTipConfig** cmdlet to create custom Policy Tips in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-PolicyTipConfig -Name <String> -Value <String> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example creates a custom Policy Tip with the following settings:

- Locale: English
- Action: `NotifyOnly`
- Policy Tip text: "This message contains content that is restricted by Contoso company policy."

```
New-PolicyTipConfig -Name en\NotifyOnly -Value "This
message contains content that is restricted by Contoso
company policy."
```

Example 2

This example sets the informational URL in Policy Tips to the value http://www.contoso.com/PolicyTipInformation.

```
New-PolicyTipConfig Url -Value "http://www.contoso.com/
PolicyTipInformation"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the custom Policy Tip you want to modify. Valid input for this parameter is one of the following values: • *<Locale>\<Action>*: Locale is a supported locale code. For example, `en` for English

			or `fr` for French. For more information about supported locales, see Supported languages for system messages. Action is one of the following Policy Tip actions: `NotifyOnly`, `RejectOverride` or `Reject`. • `Url` There can be only one custom Policy Tip with the value `Url` for the *Name* parameter. For the remaining Policy Tip actions, there can be only one custom Policy Tip for each combination of locale and action. For example, there can be only one custom Policy Tip with the *Name* value `en\NotifyOnly`, but you can create additional custom Policy Tips with the values `de\NotifyOnly` and `fr\NotifyOnly` for the *Name* parameter.
Value	Required	System.String	The *Value* parameter specifies the text that's displayed by the Policy Tip. If the value contains

			spaces, enclose the value in quotation marks (").
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-PolicyTipConfig

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-PolicyTipConfig** cmdlet to remove custom data loss prevention (DLP) Policy Tips from your organization. You can't remove built-in Policy Tips.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-PolicyTipConfig -Identity <PolicyTipConfigIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

Example 1

This example removes the custom English Policy Tip for the action value `NotifyOnly`.

```
Remove-PolicyTipConfig en\NotifyOnly
```

Example 2

This example removes all the custom Russian Policy Tips.

```
Get-PolicyTipConfig -Locale ru | Remove-PolicyTipConfig
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.PolicyNudges.PolicyTipConfigIdParameter	The *Identity* parameter specifies the custom Policy Tip you want to remove. You can use any value that uniquely identifies the custom Policy Tip. For example: • <*Locale*>\<*Action*>: Locale is a supported locale code. For example, `en` for English or `fr` for French. For more information about supported locales, see Supported languages for system messages. Action is one of the following Policy Tip

			actions: `NotifyOnly`, `RejectOverride` or `Reject`. The value `Url` • GUID • Distinguished name (DN)
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-PolicyTipConfig

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-PolicyTipConfig** cmdlet to modify custom Policy Tips in your organization. You can't modify built-in Policy Tips.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-PolicyTipConfig -Identity <PolicyTipConfigIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Name <String>] [-Value
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

Example 1

This example modifies the custom English Policy Tip with the action value `NotifyOnly`. The text of this custom Policy Tip is changed to the following value: "This message contains content that is restricted by Contoso company policy."

```
Set-PolicyTipConfig en\NotifyOnly "This message contains content that is restricted by Contoso company policy."
```

Example 2

This example replaces the text of all custom Spanish Policy Tips with the value, "Este mensaje contiene contenido que está restringida por la política de Contoso."

```
Get-PolicyTipConfig -Locale es | Set-PolicyTipConfig -Value "Este mensaje contiene contenido que está restringida por la política de Contoso."
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Data loss prevention (DLP)" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Management.PolicyNudges.PolicyTipConfigIdParameter	The *Identity* parameter specifies the custom Policy Tip you want to modify. You can use any value that uniquely identifies the custom Policy Tip. For example: • *<Locale>\<Action>*: Locale is a supported locale code. For example, `en` for English

			or `fr` for French. For more information about supported locales, see Supported languages for system messages. Action is one of the following Policy Tip actions: `NotifyOnly`, `RejectOverride` or `Reject`. • The value `Url` • GUID • Distinguished name (DN)
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			writes this configuration change to Active Directory.
Name	Optional	System.String	The *Name* parameter specifies the custom Policy Tip you want to modify. Valid input for this parameter is one of the following values: • <*Locale*>\<*Action*>: Locale is a supported locale code. For example, `en` for English or `fr` for French. For more information about supported locales, see Supported languages for system messages. Action is one of the following Policy Tip actions: `NotifyOnly`, `RejectOverride` or `Reject`. • The value `Url` There can be only one custom Policy Tip with the value `Url` for the *Name* parameter. For the remaining Policy Tip actions, there can be only one custom Policy Tip for each combination of

			locale and action. For example, there can be only one custom Policy Tip with the *Name* value `en\NotifyOnly`, but you can create additional custom Policy Tips with the values `de\NotifyOnly` and `fr\NotifyOnly` for the *Name* parameter.
Value	Optional	System.String	The *Value* parameter specifies the text that's displayed by the Policy Tip. If the value contains spaces, enclose the value in quotation marks (").
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-RetentionPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-RetentionPolicy** cmdlet to retrieve the settings for retention policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-RetentionPolicy [-Identity <MailboxPolicyIdParameter>] [-
DomainController <Fqdn>] [-IgnoreDehydratedFlag <SwitchParameter>] [-
Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE1

This example returns all the properties of the retention policy RP Finance. The output is piped to the **Format-List** cmdlet to format the results as a list of properties.

```
Get-RetentionPolicy -Identity "RP Finance" | Format-List
```

Detailed Description

A retention policy is associated with a group of retention policy tags that specify retention settings for items in a mailbox. A policy may contain one default policy tag to move items to an archive mailbox, one default policy tag to delete all items, one default policy tag to delete voicemail items, and multiple personal tags to move or delete items. A mailbox can have only one retention policy applied to it. The **Get-RetentionPolicy** cmdlet displays all policy settings associated with the specified policy.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the policy name.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-RetentionPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-25*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-RetentionPolicy** cmdlet to create a retention policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-RetentionPolicy -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-IgnoreDehydratedFlag <SwitchParameter>] [-
IsDefault <SwitchParameter>] [-IsDefaultArbitrationMailbox
<SwitchParameter>] [-Organization <OrganizationIdParameter>] [-RetentionId
<Guid>] [-RetentionPolicyTagLinks <RetentionPolicyTagIdParameter[]>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the retention policy Business General without associating any retention policy tags.

```
New-RetentionPolicy "Business General"
```

EXAMPLE 2

This example creates the retention policy Business General and uses the *RetentionPolicyTagLinks* parameter to associate two retention policy tags with this policy. You can enter multiple retention policy tags, separated by commas. If a tag name includes a space, enclose the name in quotation marks.

```
New-RetentionPolicy "Business General" -
RetentionPolicyTagLinks "General Business","Legal"
```

Note:
The second retention tag, which is named Legal, is also enclosed in quotation marks for consistency. Values that don't include a space can be enclosed in quotation marks without any change to how the command is interpreted.

Detailed Description

Retention policy tags are associated with a retention policy. When a retention policy is applied to a mailbox, tags associated with the policy are available to the mailbox user.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the policy name.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IsDefault	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use. **Note:** This parameter isn't available in on-premises deployments.
IsDefaultArbitrationMailbox	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *IsDefaultArbitrationMailbox* switch configures this policy as the default retention policy for arbitration mailboxes in your Exchange Online organization. **Note:** This parameter isn't available in on-premises deployments.

Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
RetentionId	Optional	System.Guid	The *RetentionId* parameter specifies the identity of the retention policy to ensure mailboxes moved from an on-premises Exchange deployment to the cloud continue to have the same retention policy applied to them. The *RetentionId* parameter is used in cross-premises deployments. You don't need to specify this parameter in on-premises-only deployments.
RetentionPolicyTagLinks	Optional	Microsoft.Exchange.Configuration.Tasks.RetentionPolicyTagIdParameter[]	The *RetentionPolicyTagLinks* parameter specifies the names of retention policy tags to be associated with this policy.
WhatIf	Optional	System.Management.Automation.SwitchPara	The *WhatIf* switch instructs the command

		meter	to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-RetentionPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-RetentionPolicy** cmdlet to remove a retention policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-RetentionPolicy -Identity <MailboxPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the retention policy Business Critical.

```
Remove-RetentionPolicy -Identity "Business Critical"
```

EXAMPLE 2

This example removes the retention policy Business Critical and suppresses the confirmation prompt.

```
Remove-RetentionPolicy -Identity "Business Critical" -
Confirm:$false
```

EXAMPLE 3

This example removes the retention policy Business Critical, which is assigned to users, and suppresses the confirmation prompt.

```
Remove-RetentionPolicy -Identity "Business Critical" -
Confirm:$false -Force
```

Detailed Description

Retention policies are used to apply message retention settings to folders and items in a mailbox. The **Remove-RetentionPolicy** cmdlet removes an existing retention policy.

Caution:

If you remove a retention policy that's assigned to users and they don't have another retention policy assigned, messages in those mailboxes may never expire. This may be a violation of the organization's messaging retention policies. When you attempt to remove a policy that's assigned to users, Microsoft Exchange displays a confirmation message indicating that the policy is assigned to users. Note that this message is in addition to the confirmation prompt displayed when removing a retention policy.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the retention policy name.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to override the confirmation prompt

			that appears when removing a retention policy that's assigned to users. Removing a policy that's assigned to users results in those users not having any retention policy. You don't have to specify a value with the *Force* switch.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-RetentionPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-RetentionPolicy** cmdlet to change the properties of an existing retention policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-RetentionPolicy -Identity <MailboxPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-IgnoreDehydratedFlag <SwitchParameter>] [-IsDefault <SwitchParameter>]
[-IsDefaultArbitrationMailbox <SwitchParameter>] [-Name <String>] [-
RetentionId <Guid>] [-RetentionPolicyTagLinks
<RetentionPolicyTagIdParameter[]>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the policy MyPolicy to link the retention policy tag MyRetentionPolicyTag with it.

```
Set-RetentionPolicy "MyPolicy" -RetentionPolicyTagLinks
"MyRetentionPolicyTag"
```

Note:
The *Identity* parameter is a positional parameter. Positional parameters can be used without the label (Identity). For more information about positional parameters, see Parameters.

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the name, distinguished name (DN), or GUID of the retention policy.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch overrides the confirmation prompt displayed by the cmdlet

			when you use the *RetentionId* parameter.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IsDefault	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *IsDefault* switch configures this policy as the default retention policy in your Microsoft Exchange Online organization. The default retention policy is applied to users who don't have a retention policy explicitly applied (including `$null`). After the user has a retention policy applied or set to `$null`, the default retention policy is no longer applied. **Note:** This parameter isn't available in on-premises deployments.
IsDefaultArbitrationMailbox	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The

			IsDefaultArbitrationMailbox switch configures this policy as the default retention policy for arbitration mailboxes in your Exchange Online organization. **Note:** This parameter isn't available in on-premises deployments.
Name	Optional	System.String	The *Name* parameter specifies a unique name for the retention policy.
RetentionId	Optional	System.Guid	The *RetentionId* parameter specifies the identity of the retention policy to make sure mailboxes moved between two Exchange organizations continue to have the same retention policy applied to them. For example, in a cross-forest deployment or in a cross-premises deployment, when a mailbox is moved from an on-premises Exchange server to the cloud, or a cloud-based mailbox is moved to an

			on-premises Exchange server, this parameter is used to make sure the same retention policy is applied to the mailbox. **Important:** It's not normally required to specify or modify the *RetentionId* parameter for a retention tag. The parameter is populated automatically when importing retention tags using the Import-RetentionTags.ps1 script.
RetentionPolicyTagLinks	Optional	Microsoft.Exchange.Configuration.Tasks.RetentionPolicyTagIdParameter[]	The *RetentionPolicyTagLinks* parameter specifies the identity of retention policy tags to associate with the retention policy. Mailboxes that get a retention policy applied have retention tags linked with that retention policy.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes

			would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-RetentionPolicyTag

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-RetentionPolicyTag** cmdlet to retrieve settings for a retention tag.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-RetentionPolicyTag [-Mailbox <MailboxIdParameter>] [-OptionalInMailbox
<SwitchParameter>] <COMMON PARAMETERS>


Get-RetentionPolicyTag [-Identity <RetentionPolicyTagIdParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-IgnoreDehydratedFlag
<SwitchParameter>] [-IncludeSystemTags <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-Types <ElcFolderType[]>]
```

Examples

EXAMPLE 1

This example returns all retention tags.

```
Get-RetentionPolicyTag
```

EXAMPLE 2

This example returns system tags in addition to personal and default tags.

```
Get-RetentionPolicyTag -IncludeSystemTags
```

EXAMPLE 3

This example returns the settings for the tag Consolidated Messenger.

```
Get-RetentionPolicyTag "Consolidated Messenger"
```

EXAMPLE 4

This example returns all retention tags of `Inbox` and `All` types and pipes the results to the **Format-Table** command to display the **Name**, **Type**, **RetentionEnabled**, **AgeLimitForRetention**, and **RetentionAction** properties.

```
Get-RetentionPolicyTag -Types Inbox,All | Format-Table
Name,Type,RetentionEnabled,AgeLimitForRetention,RetentionAc
tion -AutoSize
```

Detailed Description

Retention tags are used to apply message retention settings to messages or folders. There are three types of retention tags:

- Retention policy tags
- Default policy tags
- Personal tags

Retention policy tags are applied to default folders such as Inbox and Deleted Items. Personal tags are available to users to tag items and folders. The default policy tag is applied to all items that don't have a tag applied by the user or aren't inherited from the folder they're located in. The **Get-RetentionPolicyTag** cmdlet displays all the settings for the specified tag.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.RetentionPolicyTagIdParameter	The *Identity* parameter specifies the name of the tag.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IncludeSystemTags	Optional	System.Management.Automation.SwitchParameter	The *IncludeSystemTags* switch specifies whether to return any system tags.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter isn't available in this release.
OptionalInMailbox	Optional	System.Management.Automation.SwitchParameter	The *OptionalInMailbox* parameter isn't available in this release.
Organization	Optional	Microsoft.Exchange.Co	The *Organization*

		nfiguration.Tasks.OrganizationIdParameter	parameter is reserved for internal Microsoft use.
Types	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ElcFolderType[]	The *Types* parameter specifies the type of retention tag to return. Valid values include: • `Calendar` • `Contacts` • `DeletedItems` • `Drafts` • `Inbox` • `JunkEmail` • `Journal` • `Notes` • `Outbox` • `SentItems` • `Tasks` • `All` • `RssSubscriptions` • `ConversationHistory` • `Personal` The parameter accepts multiple values separated by a comma.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-RetentionPolicyTag

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-RetentionPolicyTag** cmdlet to create a retention tag.

For more information about retention tags, see Retention tags and retention policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-RetentionPolicyTag [-AddressForJournaling <RecipientIdParameter>] [-
AgeLimitForRetention <EnhancedTimeSpan>] [-JournalingEnabled <$true |
$false>] [-LabelForJournaling <String>] [-MessageClass <String>] [-
MessageFormatForJournaling <UseMsg | UseTnef>] [-RetentionAction
<MoveToDeletedItems | MoveToFolder | DeleteAndAllowRecovery |
PermanentlyDelete | MarkAsPastRetentionLimit | MoveToArchive>] [-
RetentionEnabled <$true | $false>] [-RetentionId <Guid>] <COMMON
PARAMETERS>


New-RetentionPolicyTag [-ManagedFolderToUpgrade <ELCFolderIdParameter>]
<COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-Comment <String>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreDehydratedFlag
<SwitchParameter>] [-IsDefaultAutoGroupPolicyTag <SwitchParameter>] [-
IsDefaultModeratedRecipientsPolicyTag <SwitchParameter>] [-
LocalizedComment <MultiValuedProperty>] [-LocalizedRetentionPolicyTagName
<MultiValuedProperty>] [-MustDisplayCommentEnabled <$true | $false>] [-
Organization <OrganizationIdParameter>] [-SystemTag <$true | $false>] [-
Type <Calendar | Contacts | DeletedItems | Drafts | Inbox | JunkEmail |
Journal | Notes | Outbox | SentItems | Tasks | All | ManagedCustomFolder |
RssSubscriptions | SyncIssues | ConversationHistory | Personal |
RecoverableItems | NonIpmRoot | LegacyArchiveJournals>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the retention policy tag Finance-DeletedItems for the Deleted Items default folder. When applied to a mailbox as a part of a retention policy, the tag permanently deletes items of all types in the Deleted Items folder in 30 days.

```
New-RetentionPolicyTag "Finance-DeletedItems" -Type
DeletedItems -RetentionEnabled $true -AgeLimitForRetention
30 -RetentionAction PermanentlyDelete
```

EXAMPLE 2

This example creates the default policy tag Finance-Default. When applied to a mailbox as part of a retention policy, the tag permanently deletes all items without a retention tag within 365 days. Items of a particular message class such as Voicemail, for which a default tag (a retention tag of type All) exists, aren't impacted.

```
New-RetentionPolicyTag "Finance-Default" -Type All -
RetentionEnabled $true -AgeLimitForRetention 365 -
RetentionAction PermanentlyDelete
```

EXAMPLE 3

This example creates the retention tag Business Critical of type Personal. When applied to mailbox items as part of a retention policy, the items are permanently deleted in approximately seven years.

```
New-RetentionPolicyTag "Business Critical" -Type Personal -
Comment "Use this tag for all business critical mail" -
RetentionEnabled $true -AgeLimitForRetention 2556 -
RetentionAction PermanentlyDelete
```

Detailed Description

Retention tags are used to apply message retention settings to folders and items in a mailbox.

Retention tags support a display of the tag name and an optional comment in localized languages. Language culture codes from the **CultureInfo** class are used for this purpose.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name of the tag.
AddressForJournaling	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	This parameter isn't available in this release.
AgeLimitForRetention	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *AgeLimitForRetention* parameter specifies the age at which retention is enforced on an item. The

			age limit corresponds to the number of days from the date the item was delivered, or the date an item was created if it wasn't delivered. If this parameter isn't present and the *RetentionEnabled* parameter is set to `$true`, an error is returned.
Comment	Optional	System.String	The *Comment* parameter specifies a comment for the tag.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			writes this configuration change to Active Directory.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IsDefaultAutoGroupPolicyTag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IsDefaultModeratedRecipientsPolicyTag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
JournalingEnabled	Optional	System.Boolean	This parameter isn't available in this release.
LabelForJournaling	Optional	System.String	This parameter isn't available in this release.
LocalizedComment	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *LocalizedComment* parameter specifies localized comments and their languages. When the user's language setting matches a language specified for this parameter, Microsoft Outlook and Microsoft Office Outlook Web App display the corresponding localized comment. Comments are specified in the form of *ISO*

			Language Code:Comment, for example, `LocalizedComment EN-US:"This is a localized comment in U.S. English"`.
LocalizedRetentionPolicyTagName	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *LocalizedRetentionPolicyTagName* parameter specifies localized tag names and their languages. When the user's language setting matches a language specified for this parameter, Outlook and Outlook Web App display the corresponding localized tag name. Names are specified in the form of *ISO Language Code:Name*, for example, `LocalizedRetentionPolicyTagName EN-US:"Business Critical"`.
ManagedFolderToUpgrade	Optional	Microsoft.Exchange.Configuration.Tasks.ELCFolderIdParameter	This parameter is available only in on-premises Exchange 2013. The *ManagedFolderToUpgrade* parameter specifies the name of a managed folder to use as a template for a

			retention tag.
MessageClass	Optional	System.String	The *MessageClass* parameter specifies the message type to which the tag applies. If not specified, the default value is set to `*`. With the exception of a default policy tag (DPT) for `voicemail`, Exchange doesn't support retention tags for different message types. Only tags with a *MessageClass* of `*` are supported, and they apply to all message types. To create a DPT for voice mail messages, set the *MessageClass* parameter to `voicemail` and the *Type* parameter to `All`. **Note:** A DPT for voice mail messages applies only to Microsoft Exchange Unified Messaging voice mail messages (identified by the `PR_MESSAGE_CLASS` MAPI property value `IPM.Note.Microsoft.Voicemail*`).
MessageFormatForJournaling	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.Journaling	This parameter isn't available in this release.

		Format	
MustDisplayCommentEnabled	Optional	System.Boolean	The *MustDisplayCommentEnabled* parameter specifies whether the comment can be hidden. The default value is `$true`.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
RetentionAction	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.RetentionActionType	The *RetentionAction* parameter specifies one of the following actions: • `MarkAsPastRetentionLimit` If you specify this action for a retention tag, messages that have the tag applied are marked as past the retention limit. • `MoveToFolder` This action isn't available for retention tags. • `MoveToDeletedItems` This action isn't available for retention tags. • `DeleteAndAllowRecovery` This action deletes a message and allows recovery from the Recoverable Items folder. • `PermanentlyDelete` This action permanently deletes a message. A message that has been

			permanently deleted can't be recovered using the Recoverable Items folder. Permanently deleted messages aren't returned in a Discovery search, unless litigation hold is enabled for the mailbox. • `MoveToArchive` This action moves a message to the user's archive mailbox. You can use this action for retention tags of type `All`, `Personal`, and `RecoverableItems`. If this parameter isn't present and the *RetentionEnabled* parameter is set to `$true`, an error is returned.
RetentionEnabled	Optional	System.Boolean	The *RetentionEnabled* parameter specifies whether the tag is enabled. When set to `$false`, the tag is disabled, and no retention action is taken on messages that have the tag applied. **Note:** Messages with a disabled tag are still considered tagged, so any default policy tags in the user's retention policy aren't applied to such messages. When you set the

			RetentionEnabled parameter to `$false`, the retention period for the tag is shown as **Never**. Users may apply this tag to items they want to indicate should never be deleted or should never be moved to the archive. Enabling the tag later may result in unintentional deletion or archiving of items. To avoid this situation, if a retention policy is disabled temporarily, it may be advisable to change the name of that tag so that users are discouraged from using it, such as `DISABLED_<Original Name>`.
RetentionId	Optional	System.Guid	The *RetentionId* parameter specifies an alternate tag ID to make sure the retention tag found on mailbox items tagged in an on-premises deployment matches the tag when the mailbox is moved to the cloud, or mailbox items tagged in the cloud match the tag

			when the mailbox is moved to an on-premises Exchange server. The parameter is used in cross-premises deployments. You don't need to specify this parameter in on-premises-only deployments.
SystemTag	Optional	System.Boolean	The *SystemTag* parameter specifies that the tag is created for internal Exchange functionality.
Type	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ElcFolderType	The *Type* parameter specifies the type of retention tag being created. Valid values include: • `Calendar` • `Contacts` • `DeletedItems` • `Drafts` • `Inbox` • `JunkEmail` • `Journal` • `Notes` • `Outbox` • `SentItems` • `Tasks` • `All` • `RecoverableItems` • `RssSubscriptions` • `SyncIssues` • `ConversationHistory` • `Personal` **Note:** To create a default policy tag (DPT), specify type `All`. For tags of type

			`RecoverableItems`, the only valid retention action is `MoveToArchive`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-RetentionPolicyTag

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-RetentionPolicyTag** cmdlet to remove a retention tag.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-RetentionPolicyTag -Identity <RetentionPolicyTagIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the retention tag Finance-DeletedItems.

```
Remove-RetentionPolicyTag -Identity "Finance-DeletedItems"
```

Detailed Description

Retention tags are added to a retention policy, which is applied to a mailbox.

Important:
When you use the **Remove-RetentionPolicyTag** cmdlet to remove a retention tag, it removes the tag definition stored in Active Directory. The next time the Managed Folder Assistant runs, it processes all items that have the removed tag applied and restamps them. Depending on the number of mailboxes and messages, this process may result in significant resource consumption on all Mailbox servers that contain mailboxes with a retention policy that includes the removed tag.

For more information about retention tags, see Retention tags and retention policies.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RetentionPolicyTagIdParameter	The *Identity* parameter specifies the name of the retention policy tag.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-RetentionPolicyTag

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-RetentionPolicyTag** cmdlet to modify the properties of a retention tag.

For more information about retention tags, see Retention tags and retention policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-RetentionPolicyTag -Identity <RetentionPolicyTagIdParameter> [-
AddressForJournaling <RecipientIdParameter>] [-AgeLimitForRetention
<EnhancedTimeSpan>] [-JournalingEnabled <$true | $false>] [-
LabelForJournaling <String>] [-MessageClass <String>] [-
MessageFormatForJournaling <UseMsg | UseTnef>] [-RetentionAction
<MoveToDeletedItems | MoveToFolder | DeleteAndAllowRecovery |
PermanentlyDelete | MarkAsPastRetentionLimit | MoveToArchive>] [-
RetentionEnabled <$true | $false>] <COMMON PARAMETERS>


Set-RetentionPolicyTag -Mailbox <MailboxIdParameter> [-OptionalInMailbox
<RetentionPolicyTagIdParameter[]>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Comment <String>] [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Force <SwitchParameter>] [-LegacyManagedFolder
<ELCFolderIdParameter>] [-LocalizedComment <MultiValuedProperty>] [-
LocalizedRetentionPolicyTagName <MultiValuedProperty>] [-
MustDisplayCommentEnabled <$true | $false>] [-Name <String>] [-RetentionId
<Guid>] [-SystemTag <$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the comment for the AllUsers-DeletedItems retention policy tag.

```
Set-RetentionPolicyTag "AllUsers-DeletedItems" -Comment
"Items in the Deleted Items folder will be automatically
deleted in 120 days"
```

EXAMPLE 2

This example makes optional retention tags available to user Terry Adams using the *Mailbox* and *OptionalInMailbox* parameters.

```
Set-RetentionPolicyTag -Mailbox "Terry Adams" -
OptionalInMailbox "ProjectA","ProjectB"
```

Detailed Description

Retention tags are used to apply message retention settings to folders and items in a mailbox.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Messaging records management" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RetentionPolicyTagIdParameter	The *Identity* parameter specifies the name, distinguished name (DN), or GUID of the retention policy tag to be modified.
Mailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies a mailbox for assigning opt-in tags. **Important:** You must use this parameter with the *OptionalInMailbox*

			parameter.
AddressForJournaling	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	This parameter isn't available in this release.
AgeLimitForRetention	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *AgeLimitForRetention* parameter specifies the age at which retention is enforced on an item. The age limit corresponds to the number of days from the date the item was delivered, or the date an item was created if it wasn't delivered. If this parameter isn't present and the *RetentionEnabled* parameter is set to `$true`, an error is returned.
Comment	Optional	System.String	The *Comment* parameter specifies a comment for the retention policy tag.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch overrides the confirmation prompt displayed by the cmdlet when you use the *RetentionId* parameter.
JournalingEnabled	Optional	System.Boolean	This parameter isn't available in this release.
LabelForJournaling	Optional	System.String	This parameter isn't available in this release.
LegacyManagedFolder	Optional	Microsoft.Exchange.Configuration.Tasks.ELCFolderIdParameter	The *LegacyManagedFolder* parameter specifies the name of a managed folder. The retention tag is created by using retention settings from the managed folder and its managed content settings. You can use this

			parameter to create retention tags based on existing managed folders to migrate users from managed folder mailbox policies to retention policies.
LocalizedComment	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *LocalizedComment* parameter specifies the localized comment and language for the retention policy tag. This comment is displayed in Microsoft Outlook based on the user's locale.
LocalizedRetentionPolicyTagName	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *LocalizedRetentionPolicyTagName* parameter specifies a localized name for the retention policy tag. This name is displayed in Outlook based on the user's locale.
MessageClass	Optional	System.String	The *MessageClass* parameter specifies the message type to which the tag applies. If not specified, the default value is set to *. With the exception of a default policy tag (DPT)

			for `voicemail`, Exchange doesn't support retention tags for different message types. Only tags with a *MessageClass* of `*` are supported, and they apply to all message types. To create a DPT for voice mail messages, set the *MessageClass* parameter to `voicemail` and the *Type* parameter to `All`. **Note:** A DPT for voice mail messages applies only to Microsoft Exchange Unified Messaging voice mail messages (identified by the `PR_MESSAGE_CLASS` MAPI property value `IPM.Note.Microsoft.Voicemail*`).
MessageFormatForJournaling	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.JournalingFormat	This parameter isn't available in this release.
MustDisplayCommentEnabled	Optional	System.Boolean	The *MustDisplayCommentEnabled* parameter specifies whether the comment can be hidden. The default value is `$true`.
Name	Optional	System.String	The *Name* parameter specifies the name of the

			retention policy tag.
OptionalInMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.RetentionPolicyTagIdParameter[]	The *OptionalInMailbox* parameter is used with the *Mailbox* parameter to specify opt-in retention tags available to the mailbox.
RetentionAction	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.RetentionActionType	The *RetentionAction* parameter specifies one of the following actions: • `MarkAsPastRetentionLimit` This action isn't available for retention tags. If you specify this action for a retention tag, messages that have the tag applied aren't deleted or marked as past the retention limit. • `MoveToFolder` This action isn't available for retention tags. You can't specify this action for a retention tag. • `MoveToDeletedItems` This action isn't available for retention tags. If you specify this action for a retention tag, messages that have the tag applied aren't deleted or marked as past the retention limit. • `DeleteAndAllowRecovery` This action deletes a message and allows recovery from the Recoverable Items folder.

			• `PermanentlyDelete` This action permanently deletes a message. A message that has been permanently deleted can't be recovered by using the Recoverable Items folder. Permanently deleted messages aren't returned in a Discovery search, unless litigation hold is enabled for the mailbox. • `MoveToArchive` This action moves a message to the user's archive mailbox. If this parameter isn't present and the *RetentionEnabled* parameter is set to `$true`, an error is returned.
RetentionEnabled	Optional	System.Boolean	The *RetentionEnabled* parameter specifies whether the tag is enabled. When set to `$false`, the tag is disabled, and no retention action is taken on messages that have the tag applied. **Note:** Messages with a disabled tag are still considered tagged, so any default policy tags in the user's retention policy aren't applied to such messages.

			When you set the *RetentionEnabled* parameter to `$false`, the retention period for the tag is shown as **Never**. Users may apply this tag to items that they want to indicate should never be deleted or should never be moved to the archive. Enabling the tag later may result in unintentional deletion or archiving of items. To avoid this situation, if a retention policy is disabled temporarily, it may be advisable to change the name of that tag so that users are discouraged from using it, such as `DISABLED_<Original Name>`.
RetentionId	Optional	System.Guid	The *RetentionId* parameter specifies an alternate tag ID to ensure the retention tag found on mailbox items tagged in one Exchange organization matches the tag when the mailbox is moved to another Exchange organization

			(for example, in a cross-forest deployment or in a cross-premises deployment, when a mailbox is moved from an on-premises Exchange server to the cloud, or a cloud-based mailbox is moved to an on-premises Exchange server). **Important:** It's not ordinarily required to specify or modify the *RetentionId* parameter for a retention tag. The parameter is populated automatically by *<scriptname>* when importing retention tags in a cross-forest or cross-premises deployment.
SystemTag	Optional	System.Boolean	The *SystemTag* parameter specifies whether the retention policy tag is created for internal Exchange functionality.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-RMSTemplate

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-RMSTemplate** cmdlet to retrieve the current list of active rights policy templates from the Active Directory Rights Management Services (AD RMS) deployment for the organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-RMSTemplate [-Organization <OrganizationIdParameter>] [-
DomainController <Fqdn>] [-Identity <RmsTemplateIdParameter>] [-ResultSize
<Unlimited>] [-TrustedPublishingDomain
<RmsTrustedPublishingDomainIdParameter>] [-Type <Archived | Distributed |
All>]
```

Examples

EXAMPLE 1

This example retrieves all RMS templates available from the RMS deployment.

```
Get-RMSTemplate -ResultSize unlimited
```

EXAMPLE 2

This example retrieves the Company Confidential RMS template.

```
Get-RMSTemplate -Identity "Company Confidential"
```

Detailed Description

Note:
The **Get-RMSTemplate** cmdlet doesn't return any active rights policy templates if internal licensing isn't enabled. Use the Get-IRMConfiguration cmdlet to check the *InternalLicensingEnabled* parameter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Information Rights Management (IRM) configuration" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.RmsTemplateIdParameter	The *Identity* parameter specifies the name of the RMS template.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.Orga	The *Organization* parameter is reserved for

		nizationIdParameter	internal Microsoft use.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. The default value is `1000`. To return all results, use `unlimited`.
TrustedPublishingDomain	Optional	Microsoft.Exchange.Configuration.Tasks.RmsTrustedPublishingDomainIdParameter	This parameter is available only in the cloud-based service. The *TrustedPublishingDomain* parameter specifies the trusted publishing domain you want to search for RMS templates. You can use any value that uniquely identifies the trusted publishing domain, for example: • Name • Distinguished name (DN) • GUID
Type	Optional	Microsoft.Exchange.Security.RightsManagement.RmsTemplateType	This parameter is available only in the cloud-based service. The *Type* parameter specifies the type of RMS template. Use one of the following values: • `All`

			• Archived • Distributed

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-TransportRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-04-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-TransportRule** cmdlet to disable a specific transport rule for messages that pass through the Transport service on a Mailbox server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-TransportRule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the transport rule Sales-Disclaimer.

```
Disable-TransportRule "Sales-Disclaimer"
```

Detailed Description

You can enable or disable specific transport rules at any time using the **Disable-TransportRule** and Enable-TransportRule cmdlets. To learn more about transport rules, see Transport rules.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the rule you want to disable. Enter either the name or the GUID of the rule. You can omit the parameter label.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-TransportRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-04-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-TransportRule** cmdlet to enable a specific transport rule for messages that pass through the Transport service on a Mailbox server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-TransportRule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Mode <Audit |
AuditAndNotify | Enforce>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the transport rule `Disclaimer-Finance`.

```
Enable-TransportRule "Disclaimer-Finance"
```

EXAMPLE 2

This example enables the transport rule `Require approval of messages to contoso.com` in audit mode, so you can see how the rule would function by analyzing the related entries in message tracking logs.

```
Enable-TransportRule "Require approval of messages to
contoso.com" -Mode Audit
```

Detailed Description

You can turn specific transport rules on or off in your organization at any time using the **Enable-TransportRule** and Disable-TransportRule cmdlets. To learn more about transport rules, see Transport rules.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the rule you want to turn on. Enter either the name or the GUID of the rule. You can omit this parameter label.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Mode	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleMode	The *Mode* parameter specifies in which mode this rule will operate after it's turned on. Valid values for this parameter include: • `Audit` The rule is turned on, and what would have happened if the rule was enforced is logged in message tracking logs. Exchange doesn't take any action that impacts the delivery of the message. • `AuditAndNotify` The rule is turned on, and it operates the same way it would in `Audit` mode, but notifications are also enabled. • `Enforce` The rule is turned on, and all actions specified in the

			rule are taken. The default value is `Enforce`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-TransportRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-07-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-TransportRule** cmdlet to view transport rules configured in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-TransportRule [-Identity <RuleIdParameter>] [-DlpPolicy <String>] [-
DomainController <Fqdn>] [-Filter <String>] [-Organization
<OrganizationIdParameter>] [-ResultSize <Unlimited>] [-State <Enabled |
Disabled>]
```

Examples

EXAMPLE 1

This example returns all transport rules configured in your organization.

```
Get-TransportRule
```

EXAMPLE 2

This example returns only the rule that matches the name "`Block email messages between Sales and Brokerage Groups`". The command is piped to the **Format-List** cmdlet to display the detailed configuration of the specified transport rule.

```
Get-TransportRule "Block email messages between Sales and
Brokerage Groups" | Format-List
```

For more information about pipelining, see Pipelining. For more information about how to work with the output of a command, see Working with command output.

EXAMPLE 3

This example returns the rules that are used to enforce the DLP policy PII (U.S.) in your organization.

```
Get-TransportRule -DlpPolicy "PII (U.S.)"
```

EXAMPLE 4

This example returns all rules in your organization that are used to enforce DLP policies in your organization. The command output is filtered to display only those rules that have a value for their **DlpPolicy** attribute.

```
Get-TransportRule | Where {$_.DlpPolicy -ne $null}
```

Detailed Description

The **Get-TransportRule** cmdlet lets you view the configuration of transport rules that are defined in your organization.

For information about how to configure transport rules in your organization, see Set-TransportRule. To learn more about transport rules, see Transport rules.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
DlpPolicy	Optional	System.String	The *DlpPolicy* parameter specifies the data loss prevention (DLP) policy when you want to view the rules associated with a specific DLP policy. DLP policies in your organization allow you to prevent unintentional disclosure of sensitive information. Each DLP policy is enforced using a set of transport rules. If you want to view the rules that are used to support a specific DLP policy, use this parameter to specify the name of that policy.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Filter	Optional	System.String	The *Filter* parameter specifies an OPath filter that returns all transport rules that match the specified search criteria. This parameter searches the **Description** property, which includes the conditions, exceptions, actions and the associated values of a transport rule. This parameter uses the syntax `-Filter "Description -like '*<text>*'"`. For example, `-Filter "Description -like *192.168.1.1*'"`.

Identity	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the transport rule you want to view. Enter either the name or GUID of the rule. You can omit this parameter label.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all requests that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
State	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleState	The *State* parameter specifies whether to return only the rules that are enabled or the ones that are disabled. The following values are valid for this parameter: • `Enabled` The command returns only the rules that are currently enabled. • `Disabled` The command returns only the rules that are currently disabled.

			If you don't use this parameter, the command returns all rules, both enabled and disabled.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-TransportRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-07-22*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-TransportRule** cmdlet to create transport rules in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-TransportRule -Name <String> [-ActivationDate <DateTime>] [-
ADComparisonAttribute <DisplayName | FirstName | Initials | LastName |
Office | PhoneNumber | OtherPhoneNumber | Email | Street | POBox | City |
State | ZipCode | Country | UserLogonName | HomePhoneNumber |
OtherHomePhoneNumber | PagerNumber | MobileNumber | FaxNumber |
OtherFaxNumber | Notes | Title | Department | Company | Manager |
CustomAttribute1 | CustomAttribute2 | CustomAttribute3 | CustomAttribute4
| CustomAttribute5 | CustomAttribute6 | CustomAttribute7 |
CustomAttribute8 | CustomAttribute9 | CustomAttribute10 |
CustomAttribute11 | CustomAttribute12 | CustomAttribute13 |
CustomAttribute14 | CustomAttribute15>] [-ADComparisonOperator <Equal |
NotEqual>] [-AddManagerAsRecipientType <To | Cc | Bcc | Redirect>] [-
AddToRecipients <RecipientIdParameter[]>] [-AnyOfCcHeader
<RecipientIdParameter[]>] [-AnyOfCcHeaderMemberOf <RecipientIdParameter[]
>] [-AnyOfRecipientAddressContainsWords <Word[]>] [-
AnyOfRecipientAddressMatchesPatterns <Pattern[]>] [-AnyOfToCcHeader
<RecipientIdParameter[]>] [-AnyOfToCcHeaderMemberOf
<RecipientIdParameter[]>] [-AnyOfToHeader <RecipientIdParameter[]>] [-
AnyOfToHeaderMemberOf <RecipientIdParameter[]>] [-ApplyClassification
<String>] [-ApplyHtmlDisclaimerFallbackAction <Wrap | Ignore | Reject>] [-
```

```
ApplyHtmlDisclaimerLocation <Append | Prepend>] [-ApplyHtmlDisclaimerText
<DisclaimerText>] [-ApplyOME <$true | $false>] [-
ApplyRightsProtectionTemplate <RmsTemplateIdParameter>] [-
AttachmentContainsWords <Word[]>] [-AttachmentExtensionMatchesWords
<Word[]>] [-AttachmentHasExecutableContent <$true | $false>] [-
AttachmentIsPasswordProtected <$true | $false>] [-AttachmentIsUnsupported
<$true | $false>] [-AttachmentMatchesPatterns <Pattern[]>] [-
AttachmentNameMatchesPatterns <Pattern[]>] [-
AttachmentProcessingLimitExceeded <$true | $false>] [-
AttachmentPropertyContainsWords <Word[]>] [-AttachmentSizeOver
<ByteQuantifiedSize>] [-BetweenMemberOf1 <RecipientIdParameter[]>] [-
BetweenMemberOf2 <RecipientIdParameter[]>] [-BlindCopyTo
<RecipientIdParameter[]>] [-Comments <String>] [-Confirm
[<SwitchParameter>]] [-ContentCharacterSetContainsWords <Word[]>] [-CopyTo
<RecipientIdParameter[]>] [-DeleteMessage <$true | $false>] [-Disconnect
<$true | $false>] [-DlpPolicy <String>] [-DomainController <Fqdn>] [-
Enabled <$true | $false>] [-ExceptIfADComparisonAttribute <DisplayName |
FirstName | Initials | LastName | Office | PhoneNumber | OtherPhoneNumber
| Email | Street | POBox | City | State | ZipCode | Country |
UserLogonName | HomePhoneNumber | OtherHomePhoneNumber | PagerNumber |
MobileNumber | FaxNumber | OtherFaxNumber | Notes | Title | Department |
Company | Manager | CustomAttribute1 | CustomAttribute2 | CustomAttribute3
| CustomAttribute4 | CustomAttribute5 | CustomAttribute6 |
CustomAttribute7 | CustomAttribute8 | CustomAttribute9 | CustomAttribute10
| CustomAttribute11 | CustomAttribute12 | CustomAttribute13 |
CustomAttribute14 | CustomAttribute15>] [-ExceptIfADComparisonOperator
<Equal | NotEqual>] [-ExceptIfAnyOfCcHeader <RecipientIdParameter[]>] [-
ExceptIfAnyOfCcHeaderMemberOf <RecipientIdParameter[]>] [-
ExceptIfAnyOfRecipientAddressContainsWords <Word[]>] [-
ExceptIfAnyOfRecipientAddressMatchesPatterns <Pattern[]>] [-
ExceptIfAnyOfToCcHeader <RecipientIdParameter[]>] [-
ExceptIfAnyOfToCcHeaderMemberOf <RecipientIdParameter[]>] [-
ExceptIfAnyOfToHeader <RecipientIdParameter[]>] [-
ExceptIfAnyOfToHeaderMemberOf <RecipientIdParameter[]>] [-
ExceptIfAttachmentContainsWords <Word[]>] [-
ExceptIfAttachmentExtensionMatchesWords <Word[]>] [-
ExceptIfAttachmentHasExecutableContent <$true | $false>] [-
ExceptIfAttachmentIsPasswordProtected <$true | $false>] [-
ExceptIfAttachmentIsUnsupported <$true | $false>] [-
ExceptIfAttachmentMatchesPatterns <Pattern[]>] [-
ExceptIfAttachmentNameMatchesPatterns <Pattern[]>] [-
ExceptIfAttachmentProcessingLimitExceeded <$true | $false>] [-
ExceptIfAttachmentPropertyContainsWords <Word[]>] [-
ExceptIfAttachmentSizeOver <ByteQuantifiedSize>] [-
ExceptIfBetweenMemberOf1 <RecipientIdParameter[]>] [-
ExceptIfBetweenMemberOf2 <RecipientIdParameter[]>] [-
ExceptIfContentCharacterSetContainsWords <Word[]>] [-ExceptIfFrom
<RecipientIdParameter[]>] [-ExceptIfFromAddressContainsWords <Word[]>] [-
ExceptIfFromAddressMatchesPatterns <Pattern[]>] [-ExceptIfFromMemberOf
<RecipientIdParameter[]>] [-ExceptIfFromScope <InOrganization |
NotInOrganization>] [-ExceptIfHasClassification <String>] [-
ExceptIfHasNoClassification <$true | $false>] [-ExceptIfHasSenderOverride
<$true | $false>] [-ExceptIfHeaderContainsMessageHeader <HeaderName>] [-
ExceptIfHeaderContainsWords <Word[]>] [-ExceptIfHeaderMatchesMessageHeader
<HeaderName>] [-ExceptIfHeaderMatchesPatterns <Pattern[]>] [-
ExceptIfManagerAddresses <RecipientIdParameter[]>] [-
ExceptIfManagerForEvaluatedUser <Sender | Recipient>] [-
ExceptIfMessageContainsDataClassifications <Hashtable[]>] [-
ExceptIfMessageSizeOver <ByteQuantifiedSize>] [-ExceptIfMessageTypeMatches
<OOF | AutoForward | Encrypted | Calendaring | PermissionControlled |
Voicemail | Signed | ApprovalRequest | ReadReceipt>] [-
ExceptIfRecipientADAttributeContainsWords <Word[]>] [-
ExceptIfRecipientADAttributeMatchesPatterns <Pattern[]>] [-
ExceptIfRecipientAddressContainsWords <Word[]>] [-
ExceptIfRecipientAddressMatchesPatterns <Pattern[]>] [-
ExceptIfRecipientDomainIs <Word[]>] [-ExceptIfRecipientInSenderList
<Word[]>] [-ExceptIfSCLOver <SclValue>] [-
ExceptIfSenderADAttributeContainsWords <Word[]>] [-
ExceptIfSenderADAttributeMatchesPatterns <Pattern[]>] [-
ExceptIfSenderDomainIs <Word[]>] [-ExceptIfSenderInRecipientList <Word[]>]
[-ExceptIfSenderIpRanges <MultiValuedProperty>] [-
ExceptIfSenderManagementRelationship <Manager | DirectReport>] [-
ExceptIfSentTo <RecipientIdParameter[]>] [-ExceptIfSentToMemberOf
<RecipientIdParameter[]>] [-ExceptIfSentToScope <InOrganization |
NotInOrganization | ExternalPartner | ExternalNonPartner>] [-
ExceptIfSubjectContainsWords <Word[]>] [-ExceptIfSubjectMatchesPatterns
<Pattern[]>] [-ExceptIfSubjectOrBodyContainsWords <Word[]>] [-
ExceptIfSubjectOrBodyMatchesPatterns <Pattern[]>] [-ExceptIfWithImportance
```

```
<Low | Normal | High>] [-ExpiryDate <DateTime>] [-From
<RecipientIdParameter[]>] [-FromAddressContainsWords <Word[]>] [-
FromAddressMatchesPatterns <Pattern[]>] [-FromMemberOf
<RecipientIdParameter[]>] [-FromScope <InOrganization |
NotInOrganization>] [-GenerateIncidentReport <RecipientIdParameter>] [-
GenerateNotification <DisclaimerText>] [-HasClassification <String>] [-
HasNoClassification <$true | $false>] [-HasSenderOverride <$true |
$false>] [-HeaderContainsMessageHeader <HeaderName>] [-HeaderContainsWords
<Word[]>] [-HeaderMatchesMessageHeader <HeaderName>] [-
HeaderMatchesPatterns <Pattern[]>] [-IncidentReportContent
<IncidentReportContent[]>] [-IncidentReportOriginalMail
<IncludeOriginalMail | DoNotIncludeOriginalMail>] [-LogEventText
<EventLogText>] [-ManagerAddresses <RecipientIdParameter[]>] [-
ManagerForEvaluatedUser <Sender | Recipient>] [-
MessageContainsDataClassifications <Hashtable[]>] [-MessageSizeOver
<ByteQuantifiedSize>] [-MessageTypeMatches <OOF | AutoForward | Encrypted
| Calendaring | PermissionControlled | Voicemail | Signed |
ApprovalRequest | ReadReceipt>] [-Mode <Audit | AuditAndNotify | Enforce>]
[-ModerateMessageByManager <$true | $false>] [-ModerateMessageByUser
<RecipientIdParameter[]>] [-NotifySender <NotifyOnly | RejectMessage |
RejectUnlessFalsePositiveOverride | RejectUnlessSilentOverride |
RejectUnlessExplicitOverride>] [-Organization <OrganizationIdParameter>]
[-PrependSubject <SubjectPrefix>] [-Priority <Int32>] [-Quarantine <$true
| $false>] [-RecipientADAttributeContainsWords <Word[]>] [-
RecipientADAttributeMatchesPatterns <Pattern[]>] [-
RecipientAddressContainsWords <Word[]>] [-RecipientAddressMatchesPatterns
<Pattern[]>] [-RecipientDomainIs <Word[]>] [-RecipientInSenderList <Word[]
>] [-RedirectMessageTo <RecipientIdParameter[]>] [-
RejectMessageEnhancedStatusCode <RejectEnhancedStatus>] [-
RejectMessageReasonText <DsnText>] [-RemoveHeader <HeaderName>] [-
RemoveOME <$true | $false>] [-RouteMessageOutboundConnector
<OutboundConnectorIdParameter>] [-RouteMessageOutboundRequireTls <$true |
$false>] [-RuleErrorAction <Ignore | Defer>] [-RuleSubType <None | Dlp>]
[-SCLOver <SclValue>] [-SenderADAttributeContainsWords <Word[]>] [-
SenderADAttributeMatchesPatterns <Pattern[]>] [-SenderAddressLocation
<Header | Envelope | HeaderOrEnvelope>] [-SenderDomainIs <Word[]>] [-
SenderInRecipientList <Word[]>] [-SenderIpRanges <MultiValuedProperty>] [-
SenderManagementRelationship <Manager | DirectReport>] [-SentTo
<RecipientIdParameter[]>] [-SentToMemberOf <RecipientIdParameter[]>] [-
SentToScope <InOrganization | NotInOrganization | ExternalPartner |
ExternalNonPartner>] [-SetAuditSeverity <String>] [-SetHeaderName
<HeaderName>] [-SetHeaderValue <HeaderValue>] [-SetSCL <SclValue>] [-
SmtpRejectMessageRejectStatusCode <RejectStatusCode>] [-
SmtpRejectMessageRejectText <RejectText>] [-StopRuleProcessing <$true |
$false>] [-SubjectContainsWords <Word[]>] [-SubjectMatchesPatterns
<Pattern[]>] [-SubjectOrBodyContainsWords <Word[]>] [-
SubjectOrBodyMatchesPatterns <Pattern[]>] [-UseLegacyRegex <$true |
$false>] [-WhatIf [<SwitchParameter>]] [-WithImportance <Low | Normal |
High>]
```

Examples

EXAMPLE 1

This example creates a transport rule with the following condition:

- **Between distribution list and distribution list** The first condition property value is the distribution group `Sales-Group`. The second condition property value is the distribution group `Brokerage-Group`.

The rule also has the following exceptions:

- **With text patterns in the subject** The value for this exception is `Press Release` or `Corporate Communication`.
- **From people** The values for this exception are the users Tony Smith and Pilar Ackerman.

The following action will be applied to any messages that match the "Between distribution list and distribution list" condition, but doesn't match the "with text patterns in the subject" or "from

people" exceptions:

- **Send reject message to sender** The value for this action is: `Email messages sent between the Sales department and the Brokerage department are prohibited.`

```
New-TransportRule "BlockMessagesBetweenSalesAndBrokerage" -
BetweenMemberOf1 "Sales-Group" -BetweenMemberOf2
"Brokerage-Group" -ExceptIfFrom "Tony Smith","Pilar
Ackerman" -ExceptIfSubjectContainsWords "Press
Release","Corporate Communication" -
RejectMessageEnhancedStatusCode "5.7.1" -
RejectMessageReasonText "Email messages sent between the
Sales department and the Brokerage department are
prohibited."
```

Detailed Description

Transport rule conditions and exceptions use one or more conditions along with the corresponding values to test for. For a list of supported transport rule conditions, see Transport rule conditions (predicates).

Transport rules apply actions to messages, most with corresponding action values. For a list of supported transport rule actions, see Transport rule actions.

For detailed information about how to create a transport rule, see Manage Transport Rules. For information about transport rules, see Transport rules.

In on-premises Exchange organizations, Transport rules created on Mailbox servers are stored in Active Directory. All Mailbox servers in the organization have access to the same set of transport rules. On Edge Transport servers, transport rules are saved in the local copy of Active Directory Lightweight Directory Services (AD LDS). Transport rules aren't shared or replicated between Edge Transport servers or between Mailbox servers and Edge Transport servers. Also, Mailbox servers and Edge Transport servers share a set of common conditions and actions, but some conditions and actions are exclusive to each server role.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter

			specifies the display name of the transport rule to be created. The length of the name can't exceed 64 characters.
ActivationDate	Optional	System.DateTime	The *ActivationDate* parameter specifies the date when this rule will become effective. The rule won't take any action on messages until the day you specify for this parameter.
ADComparisonAttribute	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ADAttribute	The *ADComparisonAttribute* parameter specifies an Active Directory attribute to compare between the sender and recipients. When you use this parameter, the specified Active Directory attribute of the sender is compared to the same Active Directory attribute of all the recipients of the message. You can use one of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials**

			• **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** When specifying the *ADComparisonAttribute* parameter, if you don't specify a value for the *ADComparisonOperator* parameter, the default comparison operator

			`Equal` is used. This parameter is used to define a rule condition.
ADComparisonOperator	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Evaluation	The *ADComparisonOperator* parameter specifies a comparison operator for the *ADComparisonAttribute* parameter. Valid values include: • `Equal` • `NotEqual` If you use the *ADComparisonOperator* parameter, you must also use the *ADComparisonAttribute* parameter. This parameter is used to define a rule condition.
AddManagerAsRecipientType	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.AddedRecipientType	The *AddManagerAsRecipientType* parameter specifies how the message is relayed to the manager of the sender or recipient. You can use any of the following values: • `To` The manager is added to the recipients in the To line of the message.

			• `Cc` The manager is added to the recipients in the carbon copy (Cc) line of the message. • `Bcc` The manager is added to the recipients in the blind carbon copy (Bcc) line of the message. • `Redirect` The message is redirected to the manager instead of being delivered to the original recipients. This parameter is used to define a rule action.
AddToRecipients	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AddToRecipients* parameter specifies one or more additional recipients for the message. Separate multiple recipients with commas. The specified recipients are added as To recipients. This parameter is used to define a rule action.
AnyOfCcHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfCcHeader* parameter specifies one or more recipients. The rule is applied if any of these recipients are present as a Cc recipient. This parameter is used to define a rule condition.

AnyOfCcHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfCcHeaderMemberOf* parameter specifies a distribution group. The rule is applied if a member of the specified distribution group is present as a Cc recipient. This parameter is used to define a rule condition.
AnyOfRecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *AnyOfRecipientAddressContainsWords* parameter specifies one or more words to check in a recipient address. The rule is applied if a recipient's address includes any of these words. This parameter is used to define a rule condition.
AnyOfRecipientAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *AnyOfRecipientAddressMatchesPatterns* parameter specifies one or more regular expressions to match in a recipient address. The rule is applied if any of the recipients' addresses matches the pattern you specify.

			This parameter is used to define a rule condition.
AnyOfToCcHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfToCcHeader* parameter specifies one or more recipients. The rule is applied if any of the recipients specified are present as a To or Cc recipient. This parameter is used to define a rule condition.
AnyOfToCcHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfToCcHeaderMemberOf* parameter specifies a distribution group. The rule is applied if a member of the specified distribution group is present as a To or Cc recipient. This parameter is used to define a rule condition.
AnyOfToHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfToHeader* parameter specifies one or more recipients. The rule is applied if any of the specified recipients are present as a To recipient. This parameter is used to define a rule condition.

AnyOfToHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfToHeaderMemberOf* parameter specifies a distribution group. The rule is applied if a member of the specified distribution group is present as a To recipient. This parameter is used to define a rule condition.
ApplyClassification	Optional	System.String	The *ApplyClassification* parameter specifies a message classification to apply to the message. **Note:** The message classification referred to in this parameter is the custom message classification that you can create in your organization. It isn't related to the data loss prevention (DLP) classification. This parameter is used to define a rule action.
ApplyHtmlDisclaimerFallbackAction	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.DisclaimerFallbackAction	The *ApplyHtmlDisclaimerFallbackAction* parameter specifies an action to fall back to if the HTML disclaimer can't be applied to a message. Valid fallback actions include

			the following: • `Wrap` The original message is wrapped as an attachment in a new message and the disclaimer is used as the message body for the new message. • `Ignore` The rule is ignored and the message is delivered without the disclaimer. • `Reject` The message is rejected. **Note:** This parameter is used with the *ApplyHtmlDisclaimerText* parameter. If you use the *ApplyHtmlDisclaimerText* parameter without specifying a value for this parameter, the default fallback action, `Wrap`, is used. This parameter is used to define a rule action.
ApplyHtmlDisclaimerLocation	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.DisclaimerLocation	The *ApplyHtmlDisclaimerLocation* parameter specifies the location within the message where the HTML disclaimer text is inserted. You can use either of the following two values: • `Append` The disclaimer is added to the end of the message body. • `Prepend` The disclaimer is inserted to the beginning of the

			message body. **Note:** This parameter is used with the *ApplyHtmlDisclaimerText* parameter. If you use the *ApplyHtmlDisclaimerText* parameter without specifying a value for this parameter, the default value, Append, is used. This parameter is used to define a rule action.
ApplyHtmlDisclaimerText	Optional	Microsoft.Exchange.Data.DisclaimerText	The *ApplyHtmlDisclaimerText* parameter specifies disclaimer text to be inserted in the message. Disclaimer text can include HTML tags and inline cascading style sheet (CSS) tags. You can add images using the IMG tag. This parameter is used to define a rule action.
ApplyOME	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *ApplyOME* parameter specifies that a message and its attachments will be encrypted if the message matches the conditions of this rule.

			Valid input for this parameter is `$true` or `$false`. The default is `$false`. This parameter is used to define a rule action.
ApplyRightsProtectionTemplate	Optional	Microsoft.Exchange.Configuration.Tasks.RmsTemplateIdParameter	The *ApplyRightsProtectionTemplate* parameter specifies the name of a rights management service (RMS) template to apply to the message. This action adds rights protection to the messages that meet the conditions of this rule. To use this action, an Active Directory Rights Management Services (AD RMS) server should exist in the topology or the organization should be configured to use the ILS service. For more information, see Transport protection rules. This parameter is used to define a rule action.
AttachmentContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *AttachmentContainsWord*

			s parameter specifies one or more words to check in attachments. Only supported attachment types are checked. The rule is applied if any of the attachments contain any of the words you specify. This parameter is used to define a rule condition.
AttachmentExtensionMatchesWords	Optional	Microsoft.Exchange.Data.Word[]	The *AttachmentExtensionMatchesWords* parameter specifies one or more word patterns to check in attachment extensions. The rule is applied if the extensions of any of the attachments match the word patterns you specify. This parameter is used to define a rule condition.
AttachmentHasExecutableContent	Optional	System.Boolean	The *AttachmentHasExecutableContent* parameter specifies whether the rule is applied when any attachments in the message contain executable content. If you set this parameter to `$true`, the rule is applied if

			any of the attachments contains executable content. This parameter is used to define a rule condition.
AttachmentIsPasswordProtected	Optional	System.Boolean	The *AttachmentIsPasswordProtected* parameter specifies whether the attachment is a password protected file whose contents can't be inspected. For example, if a password protected ZIP file is in a message, this condition will be met. The rule is applied if any attachment is password protected. This parameter is used to define a rule condition.
AttachmentIsUnsupported	Optional	System.Boolean	The *AttachmentIsUnsupported* parameter specifies whether the rule is applied when any attachments in the message are of an unsupported type. Unsupported attachments are attachments for which an IFilter isn't installed on the servers. If you set this parameter to `$true`, the

			rule is applied if any of the attachments is an unsupported type. This parameter is used to define a rule condition.
AttachmentMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *AttachmentMatchesPatterns* parameter specifies one or more regular expressions to match in message attachment content. Only supported attachment types are checked for the specified pattern. **Note:** Only the first 150 kilobytes (KB) of the attachment is scanned when trying to match a pattern. This parameter is used to define a rule condition.
AttachmentNameMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *AttachmentNameMatchesPatterns* parameter specifies one or more word patterns to check in attachment names. The rule is applied if the name of any attachment matches the patterns you specify.

			This parameter is used to define a rule condition.
AttachmentProcessingLimitExceeded	Optional	System.Boolean	The *AttachmentProcessingLimitExceeded* parameter specifies whether the scanning of attachments in the message didn't complete because the processing exceeded built-in limits. This condition is used to create rules that work together with other attachment processing rules and gives you the ability to handle messages whose content couldn't be fully scanned. Valid values are `$true` and `$false`. This parameter is used to define a rule condition.
AttachmentPropertyContainsWords	Optional	Microsoft.Exchange.Data.Word[]	This parameter is reserved for internal Microsoft use.
AttachmentSizeOver	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *AttachmentSizeOver* parameter specifies an attachment size. The rule is applied if the size of any of the attachments exceeds the specified size. This parameter is used to

			define a rule condition.
BetweenMemberOf1	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *BetweenMemberOf1* parameter specifies a distribution group and must be used together with the *BetweenMemberOf2* parameter. The rule is applied if the message is sent between members of the distribution groups specified in these parameters. This parameter is used to define a rule condition.
BetweenMemberOf2	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *BetweenMemberOf2* parameter specifies a distribution group and must be used together with the *BetweenMemberOf1* parameter. The rule is applied if the message is sent between members of the distribution groups specified in these parameters. This parameter is used to define a rule condition.
BlindCopyTo	Optional	Microsoft.Exchange.Configuration.Tasks.Reci	The *BlindCopyTo* parameter specifies one or

		pientIdParameter[]	more recipients to add to the message as Bcc recipients. This parameter is used to define a rule action.
Comments	Optional	System.String	The *Comments* parameter specifies informative comments for the transport rule, such as what the rule is used for or how it has changed over time. The length of the comment can't exceed 1024 characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ContentCharacterSetContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ContentCharacterSetContainsWords* parameter specifies one or more character set names to check for in the message. The rule is applied if the message contains any of

			the character sets specified. This parameter is used to define a rule condition.
CopyTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *CopyTo* parameter specifies one or more recipients to add to the message as Cc recipients. This parameter is used to define a rule action.
DeleteMessage	Optional	System.Boolean	The *DeleteMessage* parameter specifies that the rule deletes any message that matches the conditions specified. This parameter is used to define a rule action.
Disconnect	Optional	System.Boolean	The *Disconnect* parameter specifies whether the rules agent disconnects the SMTP session. This parameter is used to define a rule action.
DlpPolicy	Optional	System.String	The *DlpPolicy* parameter specifies the data loss prevention (DLP) Policy associated with this rule. Each DLP policy is enforced using a set of Transport rules. To learn

			more about DLP, see Data loss prevention.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the transport rule should be enabled when it's created. The default value is `$true`.
ExceptIfADComparisonAttribute	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ADAttribute	The *ExceptIfADComparisonAttribute* parameter specifies an Active Directory attribute to compare

			between the sender and recipients. When you use this parameter, the specified Active Directory attribute of the sender is compared to the same Active Directory attribute of all the recipients of the message. You can use one of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber**

			• **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** When specifying the *ExceptIfADComparisonAttribute* parameter, if you don't specify a value for the *ExceptIfADComparisonOperator* parameter, the default comparison operator `Equal` is used.
ExceptIfADComparisonOperator	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Evaluation	The *ExceptIfADComparisonOperator* parameter specifies a comparison operator for the *ExceptIfADComparisonAttribute* parameter. Valid values are: • `Equal` • `NotEqual` If you use the *ExceptIfADComparisonOperator* parameter, you must also use the *ExceptIfADComparisonAttribute* parameter.

ExceptIfAnyOfCcHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfAnyOfCcHeader* parameter specifies one or more recipients. The rule isn't applied if any of these recipients are present as a Cc recipient.
ExceptIfAnyOfCcHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfAnyOfCcHeaderMemberOf* parameter specifies a distribution group. The rule isn't applied if a member of the specified distribution group is present as a Cc recipient.
ExceptIfAnyOfRecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfAnyOfRecipientAddressContainsWords* parameter specifies one or more words to check in a recipient address. The rule isn't applied if a recipient's address includes any of these words.
ExceptIfAnyOfRecipientAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfAnyOfRecipientAddressMatchesPatterns* parameter specifies one or more regular expressions to match in a recipient

			address. The rule isn't applied if any of the recipient addresses matches the pattern you specify.
ExceptIfAnyOfToCcHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfAnyOfToCcHeader* parameter specifies one or more recipients. The rule isn't applied if any of the recipients specified are present as a To or Cc recipient.
ExceptIfAnyOfToCcHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfAnyOfToCcHeaderMemberOf* parameter specifies a distribution group. The rule isn't applied if a member of the specified distribution group is present as a To or Cc recipient.
ExceptIfAnyOfToHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfAnyOfToHeader* parameter specifies one or more recipients. The rule isn't applied if any of the specified recipients are present as a To recipient.
ExceptIfAnyOfToHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.Reci	The *ExceptIfAnyOfToHeaderM*

		pientIdParameter[]	*emberOf* parameter specifies a distribution group. The rule isn't applied if a member of the specified distribution group is present as a To recipient.
ExceptIfAttachmentContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfAttachmentContainsWords* parameter specifies one or more words to check in attachments. Only supported attachment types are checked. The rule isn't applied if any of the attachments contain any of the words you specify.
ExceptIfAttachmentExtensionMatchesWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfAttachmentExtensionMatchesWords* parameter specifies one or more word patterns to check in attachment extensions. The rule isn't applied if the extensions of any of the attachments match the word patterns you specify.
ExceptIfAttachmentHa	Optional	System.Boolean	The

sExecutableContent			*ExceptIfAttachmentHasExecutableContent* parameter specifies whether the rule is applied when any attachments in the message contain executable content. If you set this parameter to `$true`, the rule isn't applied if any of the attachments contains executable content.
ExceptIfAttachmentIsPasswordProtected	Optional	System.Boolean	The *ExceptIfAttachmentIsPasswordProtected* parameter specifies whether the attachment is a password protected file whose contents can't be inspected. For example, if a password protected ZIP file is in a message, this exception will be met. The rule isn't applied if any attachment is password protected.
ExceptIfAttachmentIsUnsupported	Optional	System.Boolean	The *ExceptIfAttachmentIsUnsupported* parameter specifies whether the rule is applied when any attachments in the

			message are of an unsupported type. Unsupported attachments are attachments for which an IFilter isn't installed on your servers. If you set this parameter to `$true` the rule isn't applied if any of the attachments is an unsupported type.
ExceptIfAttachmentMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfAttachmentMatchesPatterns* parameter specifies one or more regular expressions to match in message attachment content. Only supported attachment types are checked for the specified pattern. **Note:** Only the first 150 KB of the attachment is scanned when trying to match a pattern.
ExceptIfAttachmentNameMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfAttachmentNameMatchesPatterns* parameter specifies one or more word patterns to check in attachment names. The rule isn't applied if the name of any

			attachment matches the patterns you specify.
ExceptIfAttachmentProcessingLimitExceeded	Optional	System.Boolean	The *ExceptIfAttachmentProcessingLimitExceeded* parameter specifies whether the scanning of attachments in the message didn't complete because the processing exceeded built-in limits. This condition is used to create rules that work together with other attachment processing rules and gives you the ability to handle messages whose content couldn't be fully scanned. Valid values are `$true` and `$false`.
ExceptIfAttachmentPropertyContainsWords	Optional	Microsoft.Exchange.Data.Word[]	This parameter is reserved for internal Microsoft use.
ExceptIfAttachmentSizeOver	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ExceptIfAttachmentSizeOver* parameter specifies an attachment size. The rule isn't applied if the size of any of the attachments exceeds the specified size.
ExceptIfBetweenMemb	Optional	Microsoft.Exchange.Co	The

erOf1		nfiguration.Tasks.RecipientIdParameter[]	*ExceptIfBetweenMemberOf1* parameter specifies a distribution group and must be used together with the *ExceptIfBetweenMemberOf2* parameter. The rule isn't applied if the message is sent between members of the distribution groups specified in these parameters.
ExceptIfBetweenMemberOf2	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfBetweenMemberOf2* parameter specifies a distribution group and must be used together with the *ExceptIfBetweenMemberOf1* parameter. The rule isn't applied if the message is sent between members of the distribution groups specified in these parameters.
ExceptIfContentCharacterSetContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfContentCharacterSetContainsWords* parameter specifies one or more character set names

			to check for in the message. The rule isn't applied if the message contains any of the character sets specified.
ExceptIfFrom	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfFrom* parameter specifies the sender. The rule isn't applied to messages received from this sender.
ExceptIfFromAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfFromAddressContainsWords* parameters specifies one or more words to check for in the From address. The rule isn't applied if the sender's address includes any of these words.
ExceptIfFromAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfFromAddressMatchesPatterns* parameter specifies one or more regular expressions to match in the sender's address. The rule isn't applied if the sender's address matches the pattern you specify.
ExceptIfFromMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.Reci	The *ExceptIfFromMemberOf*

		pientIdParameter[]	parameter specifies a distribution group. The rule isn't applied if the sender of the message is a member of this distribution group.
ExceptIfFromScope	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.FromUserScope	The *ExceptIfFromScope* parameter specifies whether the sender is inside or outside your organization. Valid values for this parameter are: • `InOrganization` • `NotInOrganization`
ExceptIfHasClassification	Optional	System.String	The *ExceptIfHasClassification* parameter specifies a message classification. The rule isn't applied to messages that have the specified classification. **Note:** The message classification referred to in this parameter is the custom message classification that you can create in your organization. It isn't related to the DLP classification.
ExceptIfHasNoClassification	Optional	System.Boolean	The *ExceptIfHasNoClassification* parameter specifies that the rule isn't applied to messages that don't have

			a message classification.
ExceptIfHasSenderOverride	Optional	System.Boolean	The *ExceptIfHasSenderOverride* parameter specifies the rule to check if the sender has chosen to override a DLP policy. Set this parameter to `$true` to prevent this rule from applying to messages where the sender took action to override a DLP policy restriction.
ExceptIfHeaderContainsMessageHeader	Optional	Microsoft.Exchange.Data.HeaderName	The *ExceptIfHeaderContainsMessageHeader* parameter specifies the SMTP message header to inspect for specific words or patterns. This parameter is used together with the *ExceptIfHeaderContainsWords* parameter.
ExceptIfHeaderContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfHeaderContainsWords* parameter specifies one or more words to look for in the message header specified in the *ExceptIfHeaderContainsMessageHeader* parameter.

			The rule isn't applied to messages where the header value of the specified header matches any of the words specified.
ExceptIfHeaderMatchesMessageHeader	Optional	Microsoft.Exchange.Data.HeaderName	The *ExceptIfHeaderMatchesMessageHeader* parameter specifies an SMTP message header to inspect. This parameter is used together with the *ExceptIfHeaderMatchesPatterns* parameter.
ExceptIfHeaderMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfHeaderMatchesPatterns* parameter specifies a pattern to match in the header specified in the *ExceptIfHeaderMatchesMessageHeader* parameter.
ExceptIfManagerAddresses	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfManagerAddresses* parameter specifies a recipient. The rule isn't applied to messages where the specified recipient is the manager of the sender or the recipient. Whether it's the manager for the sender or the

			recipient is defined in the *ExceptIfManagerForEvaluatedUser* parameter.
ExceptIfManagerForEvaluatedUser	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.EvaluatedUser	The *ExceptIfManagerForEvaluatedUser* parameter specifies whether the sender or the recipient's manager should be evaluated. The specified user's manager attribute is compared with users specified in the *ExceptIfManagerAddresses* parameter. Valid values include: • `Recipient` • `Sender` Use this parameter together with the *ExceptIfManagerAddresses* parameter.
ExceptIfMessageContainsDataClassifications	Optional	System.Collections.Hashtable[]	The *ExceptIfMessageContainsDataClassifications* parameter specifies the sensitive information types to look for in the message body and any of the attachments. For a list of sensitive information types available, see Sensitive information

			types inventory.
ExceptIfMessageSizeOver	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ExceptIfMessageSizeOver* parameter specifies a message size. The rule isn't applied to any messages that exceed the message size you specify for this parameter.
ExceptIfMessageTypeMatches	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.MessageType	The *ExceptIfMessageTypeMatches* parameter specifies a message type. The rule isn't applied to any messages that match the message type you specify. Valid values include: • `OOF` Auto-reply messages configured by the user • `AutoForward` Messages automatically forwarded to an alternative recipient • `Encrypted` Encrypted messages • `Calendaring` Meeting requests and responses • `PermissionControlled` Messages that have specific permissions configured • `Voicemail` Voice mail messages forwarded by Unified Messaging Service • `Signed` Digitally signed

			messages • `ApprovalRequest` Moderation request messages sent to moderators • `ReadReceipt` Read receipts
ExceptIfRecipientADAttributeContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfRecipientADAttributeContainsWords* parameter specifies one or more words to check for in specific Active Directory attributes of the recipient. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber**

			• **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the words. For example, to look for the word `Sales` in the `Department` attribute, set this parameter to `Department:Sales`. If you want to specify multiple values for multiple attributes, separate them with commas. The rule isn't applied if any of the specified attributes have the value specified.
ExceptIfRecipientADAttributeMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfRecipientADAttributeMatchesPatterns* parameter specifies one or more patterns to check for

			in the specified Active Directory attribute of the recipient. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager**

			• **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the pattern. If you want to specify multiple patterns for multiple attributes, separate them with commas. The rule isn't applied if the values of any of the specified attributes match the specified patterns for that attribute.
ExceptIfRecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfRecipientAddressContainsWords* parameter specifies words to check for in the recipient address.
ExceptIfRecipientAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfRecipientAddressMatchesPatterns* parameter specifies one or more text patterns to match in the recipient address.
ExceptIfRecipientDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfRecipientDomainIs* parameter specifies the

			recipient's domain. The rule isn't applied to messages sent to recipients whose email addresses are in the specified domain.
ExceptIfRecipientInSenderList	Optional	Microsoft.Exchange.Data.Word[]	This parameter is available only in the cloud-based service. The *ExceptIfRecipientInSenderList* parameter specifies an exception when a recipient is defined in a supervision list entry on the sender's mailbox. Supervision list entries perform the following functions: • They specify individual exceptions for the user in the closed campus supervision policy. • They identify the user as a victim in the anti-bullying policy. To view the supervision list entries that are configured on a user's mailbox, run the **Get-SupervisionListEntry** command and specify the user's mailbox.

ExceptIfSCLOver	Optional	Microsoft.Exchange.Data.SclValue	The *ExceptIfSCLOver* parameter specifies a spam confidence level (SCL) value. The rule isn't applied to messages with an SCL equal to or higher than the value specified. Valid SCL values are integers from 0 through 9, and -1. The value -1 specifies that the message is from a trusted source.
ExceptIfSenderADAttributeContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfSenderADAttributeContainsWords* parameter specifies one or more words to check for in specific Active Directory attributes of the sender. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City**

			• **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the words. For example, to look for the word `Sales` in the `Department` attribute, set this parameter to `Department:Sales`. If you want to specify multiple values for multiple attributes, separate them with commas. The rule isn't applied if any of the

			specified attributes have the value specified.
ExceptIfSenderADAttributeMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfSenderADAttributeMatchesPatterns* parameter specifies one or more patterns to check for in the specified Active Directory attribute of the sender. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber**

			• **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the pattern. If you want to specify multiple patterns for multiple attributes, separate them with commas. The rule isn't applied if the values of any of the specified attributes match the specified patterns for that attribute.
ExceptIfSenderDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfSenderDomainIs* parameter specifies the sender's domain. The rule isn't applied to messages received from senders whose email addresses are in the specified domain.

ExceptIfSenderInRecipientList	Optional	Microsoft.Exchange.Data.Word[]	This parameter is available only in the cloud-based service. The *ExceptIfSenderInRecipientList* parameter specifies an exception when the sender is defined in a supervision list entry on a recipient's mailbox. Supervision list entries perform the following functions: • They specify individual exceptions for the user in the closed campus supervision policy. • They identify the user as a victim in the anti-bullying policy. To view the supervision list entries that are configured on a user's mailbox, run the **Get-SupervisionListEntry** command and specify the user's mailbox.
ExceptIfSenderIpRanges	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfSenderIpRanges* parameter specifies the IP ranges to compare with the sender's IP address. The rule isn't applied if the

			IP address of the sender falls within one of the IP ranges specified in this parameter.
ExceptIfSenderManagementRelationship	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ManagementRelationship	The *ExceptIfSenderManagementRelationship* parameter specifies a relationship between the sender and the recipient. Valid values are: • `Manager` The rule isn't applied if the sender is the manager of the recipient. • `DirectReport` The rule isn't applied if the sender is a direct report of the recipient.
ExceptIfSentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfSentTo* parameter specifies a recipient. The rule isn't applied to messages sent to the specified recipient.
ExceptIfSentToMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfSentToMemberOf* parameter specifies a distribution group. The rule isn't applied to messages where any recipient is a member of the specified group. **Note:** If the distribution group is

			removed after creation of the rule, no exception is made.
ExceptIfSentToScope	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ToUserScope	The *ExceptIfSentToScope* parameter specifies whether the message is sent to internal, external, or partner recipients. Valid values are: • `InOrganization` The recipients are internal to your organization. • `NotInOrganization` The recipients are outside your organization. • `ExternalPartner` The recipients are in a partner organization. • `ExternalNonPartner` The recipients are external to your organization which isn't a partner organization.
ExceptIfSubjectContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfSubjectContainsWords* parameter specifies words to look for in the message subject. You can specify one or more words or phrases. When specifying more than one word or phrase, each word or phrase should be separated by a comma (,). When specifying a phrase that

			contains one or more spaces, you must enclose the phrase in quotation marks ("), for example: `word1,"Phrase with spaces",word2`.
ExceptIfSubjectMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfSubjectMatchesPatterns* parameter specifies text patterns to check the message subject for.
ExceptIfSubjectOrBodyContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfSubjectOrBodyContainsWords* parameter specifies words to look for in the message subject and body. The rule isn't applied if any of the words or phrases specified is found in the message subject or body. You can specify one or more words or phrases. When specifying more than one word or phrase, each word or phrase should be separated by a comma (,). When specifying a phrase with one or more spaces, you must enclose the phrase in quotation marks ("), for

			example: `Word1,"Phrase with spaces",word2.`
ExceptIfSubjectOrBodyMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfSubjectOrBodyMatchesPatterns* parameter specifies text patterns to look for in the message subject and body. The rule isn't applied if the word specified is found in the message subject or body.
ExceptIfWithImportance	Optional	Microsoft.Exchange.Management.Tasks.Importance	The *ExceptIfWithImportance* parameter specifies message importance. The rule isn't applied to messages matching the specified importance. Valid values are: • `High` • `Low` • `Normal`
ExpiryDate	Optional	System.DateTime	The *ExpiryDate* parameter specifies the date when this rule will stop processing. The rule won't take any action on messages past the date you specify for this parameter.
From	Optional	Microsoft.Exchange.Configuration.Tasks.Reci	The *From* parameter specifies the sender. The

		pientIdParameter[]	rule is applied to messages received from this sender. This parameter is used to define a rule condition.
FromAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *FromAddressContainsWords* parameter specifies one or more words to check for in the From address. The rule is applied if the sender's address includes any of these words. This parameter is used to define a rule condition.
FromAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *FromAddressMatchesPatterns* parameter specifies one or more regular expressions to match in the sender's address. The rule is applied if the sender's address matches the pattern you specify. This parameter is used to define a rule condition.
FromMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *FromMemberOf* parameter specifies a distribution group. The rule is applied if the sender of the message is a

			member of this distribution group. This parameter is used to define a rule condition.
FromScope	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.FromUserScope	The *FromScope* parameter specifies whether the sender is inside or outside your organization. Valid values for this parameter are: • `InOrganization` • `NotInOrganization` This parameter is used to define a rule condition.
GenerateIncidentReport	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *GenerateIncidentReport* parameter specifies the recipient to whom incident reports will be sent. An incident report is generated for messages that violate a DLP policy in your organization. This parameter is used to define a rule action.
GenerateNotification	Optional	Microsoft.Exchange.Data.DisclaimerText	This parameter is reserved for internal Microsoft use.
HasClassification	Optional	System.String	The *HasClassification* parameter specifies a message classification. The rule is applied to

			messages that have the specified classification. **Note:** The message classification referred to in this parameter is the custom message classification that you can create in your organization. It isn't related to the DLP classification. This parameter is used to define a rule condition.
HasNoClassification	Optional	System.Boolean	The *HasNoClassification* parameter specifies whether the rule is applied to messages that don't have a message classification. If you set this parameter to `$true`, the rule is applied to all messages that don't have a message classification. If you set this parameter to `$false`, the rule is applied to all messages that have one or more message classifications. This parameter is used to define a rule condition.
HasSenderOverride	Optional	System.Boolean	The *HasSenderOverride* parameter specifies the

			rule to check if the sender has chosen to override a DLP policy. Set this parameter to `$true` to apply this rule to messages where the sender took action to override a DLP policy restriction. This parameter is used to define a rule condition.
HeaderContainsMessageHeader	Optional	Microsoft.Exchange.Data.HeaderName	The *HeaderContainsMessageHeader* parameter specifies the SMTP message header to inspect for specific words or patterns. This parameter is used together with the *HeaderContainsWords* parameter. This parameter is used to define a rule condition.
HeaderContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *HeaderContainsWords* parameter specifies one or more words to look for in the message header specified in the *HeaderContainsMessageHeader* parameter. The rule is applied to messages

			where the header value of the specified header matches any of the words specified. This parameter is used to define a rule condition.
HeaderMatchesMessageHeader	Optional	Microsoft.Exchange.Data.HeaderName	The *HeaderMatchesMessageHeader* parameter specifies an SMTP message header to inspect. This parameter is used together with the *HeaderMatchesPatterns* parameter. This parameter is used to define a rule condition.
HeaderMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *HeaderMatchesPatterns* parameter specifies a pattern to match in the header specified in the *HeaderMatchesMessageHeader* parameter. This parameter is used to define a rule condition.
IncidentReportContent	Optional	Microsoft.Exchange.MessagingPolicies.Rules.IncidentReportContent[]	The *IncidentReportContent* parameter specifies the message properties that are included in the incident reports. This parameter is used

			together with the *GenerateIncidentReport* parameter. The valid values are: • `Sender` Includes the sender of the message. • `Recipients` Includes the recipients in the To: box of the message. Only the first 10 recipients are displayed in the incident report. If there are more than 10 recipients, the remaining number of recipients will be displayed. • `Subject` Includes the message subject. • `CC` Includes the recipients in the Cc: box of the message. Only the first 10 recipients are displayed in the incident report. If there are more than 10 recipients, the remaining number of recipients will be displayed. • `BCC` Includes the recipients in the Bcc: box of the message. Only the first 10 recipients are displayed in the incident report. If there are more than 10 recipients, the remaining number of recipients will be displayed. • `Severity` Includes the audit severity of the rule that was triggered. If the message was processed

			by more than one rule, the highest severity is displayed. • `Override` Includes the override if the sender has chosen to override a PolicyTip. If the sender has provided a justification, the first 100 characters of the justification is also included. • `RuleDetections` Includes the list of rules that the message triggered. • `FalsePositive` Includes the false positive if the sender marked the message as a false positive for a PolicyTip. • `DataClassifications` Includes the list of sensitive information types detected in the message. • `IdMatch` Includes the sensitive information type detected, the exact matched content from the message, and the 150 characters before and after the matched sensitive information. • `AttachOriginalMail` Includes the entire original message. **Note:** The message ID is always included in the incident report. This parameter is used to

			define a rule action.
IncidentReportOriginalMail	Optional	Microsoft.Exchange.MessagingPolicies.Rules.IncidentReportOriginalMail	The *IncidentReportOriginalMail* parameter specifies whether to include the original message with the incident report. This parameter is used together with the *GenerateIncidentReport* parameter. Valid values are: • `IncludeOriginalMail` • `DoNotIncludeOriginalMail` The default value is `DoNotIncludeOriginalMail`. This parameter is used to define a rule action. **Important:** The functionality of this parameter is now managed by the *IncidentReportContent* parameter, and this parameter will be deprecated in the future. Adding the value `AttachOriginalMail` to the *IncidentReportContent* parameter is equivalent to setting this parameter to `IncludeOriginalMail` value. Even though this parameter is still functional, we recommend you use the *IncidentReportContent* parameter instead.

LogEventText	Optional	Microsoft.Exchange.Data.EventLogText	The *LogEventText* parameter specifies a message string to add to the event log entry for this rule. This parameter is used to define a rule action.
ManagerAddresses	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ManagerAddresses* parameter specifies a recipient. The rule is applied to messages where the specified recipient is the manager of the sender or the recipient. Whether it's the manager for the sender or the recipient is defined in the *ManagerForEvaluatedUser* parameter. This parameter is used to define a rule condition.
ManagerForEvaluatedUser	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.EvaluatedUser	The *ManagerForEvaluatedUser* parameter specifies whether the sender or the recipient's manager should be evaluated. The specified user's manager attribute is compared with users specified in the *ManagerAddresses* parameter. Valid values

			include: • `Recipient` • `Sender` Use this parameter together with the *ManagerAddresses* parameter. This parameter is used to define a rule condition.
MessageContainsDataClassifications	Optional	System.Collections.Hashtable[]	The *MessageContainsDataClassifications* parameter specifies the sensitive information types to look for in the message body and any of the attachments. For a list of sensitive information types available, see Sensitive information types inventory. This parameter is used to define a rule condition.
MessageSizeOver	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *MessageSizeOver* parameter specifies a message size. The rule is applied to all messages that exceed the message size you specify for this parameter. This parameter is used to define a rule condition.

MessageTypeMatches	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.MessageType	The *MessageTypeMatches* parameter specifies a message type. The rule is applied to all messages that match the message type you specify. Valid values include: • `OOF` Auto-reply messages configured by the user • `AutoForward` Messages automatically forwarded to an alternative recipient • `Encrypted` Encrypted messages • `Calendaring` Meeting requests and responses • `PermissionControlled` Messages that have specific permissions configured • `Voicemail` Voice mail messages forwarded by Unified Messaging service • `Signed` Digitally signed messages • `ApprovalRequest` Moderations request messages sent to moderators • `ReadReceipt` Read receipts This parameter is used to define a rule condition.
Mode	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleMode	The *Mode* parameter specifies in which mode this rule will operate. Valid

			values include: • `Audit` The rule is turned on, and what would have happened if the rule was enforced is logged in message tracking logs. Exchange doesn't take any action that impacts the delivery of the message. • `AuditAndNotify` The rule is turned on, and it operates the same way it would in `Audit` mode, but notifications are also enabled. • `Enforce` The rule is turned on, and all actions specified in the rule are taken. The default value `Enforce`.
ModerateMessageByManager	Optional	System.Boolean	The *ModerateMessageByManager* parameter specifies whether the message should be forwarded to the sender's manager for approval. To enable moderation by the sender's manager, set the value to `$true`. This parameter is used to define a rule action.
ModerateMessageByUser	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ModerateMessageByUser* parameter specifies a

			recipient to forward the message to for approval. This parameter is used to define a rule action.
NotifySender	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.NotifySenderType	The *NotifySender* parameter specifies how the sender of a message that goes against a DLP policy is notified. Valid values are: • `NotifyOnly` Sender is notified, but the message is delivered normally. • `RejectMessage` Message is rejected, and the sender is notified. • `RejectUnlessFalsePositiveOverride` Message is rejected unless it's marked as a false positive by the sender. • `RejectUnlessSilentOverride` Message is rejected unless the sender has chosen to override the policy restriction. • `RejectUnlessExplicitOverride` This is the same as `RejectUnlessSilentOverride`, but the sender can also provide a justification for overriding the policy restriction. If you specify any value

			other than `NotifyOnly`, you can provide a specific rejection status code and reason using the *RejectMessageEnhancedStatusCode* and *RejectMessageReasonText* parameters. This action is used together with the *MessageContainsDataClassifications* condition. If you use this parameter, you must also specify the sensitive information types you want to check against using the *MessageContainsDataClassifications* parameter. This parameter is used to define a rule action.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
PrependSubject	Optional	Microsoft.Exchange.Data.SubjectPrefix	The *PrependSubject* parameter specifies a word or phrase to add to the beginning of the message subject. **Note:** The rule will add the text as you specify in this

			parameter without adding spaces or other characters to separate it from the original subject. Consider ending the value you specify in this parameter with a colon (:) and a space, or at least a space, to separate it from the original subject. This parameter is used to define a rule action.
Priority	Optional	System.Int32	The *Priority* parameter specifies the priority for this transport rule. Rules with a lower priority value are processed first. If you modify the priority of the rule, the position of the rule in the rule list changes to match the priority that you specified, and the Transport Rules agent increments all rules with a higher priority value. The value of this parameter must be greater than or equal to 0, and must be one less than the total number of transport rules in your organization. For example, if you configured 8 transport rules, you can set this parameter to any

			value from 0 through 7.
Quarantine	Optional	System.Boolean	The *Quarantine* parameter specifies whether the rules agent delivers the message to the quarantine mailbox specified in the Content Filtering configuration. This parameter is used to define a rule action.
RecipientADAttributeContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *RecipientADAttributeContainsWords* parameter specifies one or more words to check for in specific Active Directory attributes of the recipient. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State**

			• **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the words. For example, to look for the word `Sales` in the `Department` attribute, set this parameter to `Department:Sales`. If you want to specify multiple values for multiple attributes, separate them with commas. The rule will be applied if any of the specified attributes have

			the value specified. This parameter is used to define a rule condition.
RecipientADAttributeMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *RecipientADAttributeMatchesPatterns* parameter specifies one or more patterns to check for in the specified Active Directory attribute of the recipient. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber**

			• **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the pattern. If you want to specify multiple patterns for multiple attributes, separate them with commas. The rule will be applied if the values of any of the specified attributes match the specified patterns for that attribute. This parameter is used to define a rule condition.
RecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *RecipientAddressContainsWords* parameter specifies one or more words to check for in the recipient's email address.

			This parameter is used to define a rule condition.
RecipientAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *RecipientAddressMatchesPatterns* parameter specifies a pattern to check the recipient address for. This parameter is used to define a rule condition.
RecipientDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *RecipientDomainIs* parameter specifies the recipient's domain. The rule is applied to messages sent to recipients whose email addresses are in the specified domain. This parameter is used to define a rule condition.
RecipientInSenderList	Optional	Microsoft.Exchange.Data.Word[]	This parameter is available only in the cloud-based service. The *RecipientInSenderList* parameter specifies the condition when a recipient is defined in a supervision list entry on the sender's mailbox. Supervision list entries perform the following functions:

			• They specify individual exceptions for the user in the closed campus supervision policy. • They identify the user as a victim in the anti-bullying policy. To view the supervision list entries that are configured on a user's mailbox, run the **Get-SupervisionListEntry** command and specify the user's mailbox. This parameter is used to define a rule condition.
RedirectMessageTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *RedirectMessageTo* parameter specifies that the rule redirects the message to the specified recipient. This parameter is used to define a rule action.
RejectMessageEnhancedStatusCode	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RejectEnhancedStatus	The *RejectMessageEnhancedStatusCode* parameter specifies an enhanced status code to provide when rejecting messages. Valid values are `5.7.1` or between `5.7.10` and `5.7.999`.

			Note: The transport rule can add a custom rejection message. To further customize the delivery status notification (DSN), you need to create a custom DSN message using the **New-SystemMessage** cmdlet. If an enhanced status code isn't specified, and only the *RejectMessageReasonText* parameter is used, the enhanced status code `5.7.1` is used. This parameter is used to define a rule action.
RejectMessageReasonText	Optional	Microsoft.Exchange.Data.DsnText	The *RejectMessageReasonText* parameter specifies a reason that explains why the message was rejected. **Note:** The transport rule can add a custom rejection message. To further customize the DSN, you need to create a custom DSN message using the **New-SystemMessage** cmdlet. If a *RejectMessageReasonText* parameter value isn't specified, and an

			enhanced status code is specified by using the *RejectMessageEnhancedStatusCode* parameter, the default reason text "`Delivery not authorized, message refused`" is used. This parameter is used to define a rule action.
RemoveHeader	Optional	Microsoft.Exchange.Data.HeaderName	The *RemoveHeader* parameter specifies an SMTP header name to be removed from the message. This parameter is used to define a rule action.
RemoveOME	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *RemoveOME* parameter specifies that a message and its attachments will be decrypted if the message matches the conditions of this rule. Valid input for this parameter is `$true` or `$false`. The default is `$false`.

			This parameter is used to define a rule action.
RouteMessageOutboundConnector	Optional	Microsoft.Exchange.Configuration.Tasks.OutboundConnectorIdParameter	This parameter is available only in the cloud-based service. The *RouteMessageOutboundConnector* parameter specifies the connector to use for routing this message. This parameter is used to define a rule action.
RouteMessageOutboundRequireTls	Optional	System.Boolean	The *RouteMessageOutboundRequireTls* parameter specifies that Transport Layer Security (TLS) encryption is required when routing this message outside your organization. Set this parameter to `$true` to require TLS. This parameter is used to define a rule action.
RuleErrorAction	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleErrorAction	The *RuleErrorAction* parameter specifies how the message should be handled if the rule processing can't be

			completed. Valid values are: • `Ignore` The message is sent without completing the rule processing. • `Defer` The message is deferred so the rules engine can attempt to process the message again. The default value is `Ignore`.
RuleSubType	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleSubType	The *RuleSubType* parameter specifies the type of this transport rule. Valid values are: • `Dlp` Specifies that this rule is associated with a DLP policy. • `None` Specifies that this is a regular transport rule.
SCLOver	Optional	Microsoft.Exchange.Data.SclValue	The *SCLOver* parameter specifies an SCL value. The rule is applied to messages with an SCL equal to or higher than the value specified. Valid SCL values are integers from 0 through 9, and -1. The value -1 specifies that the message is from a trusted source. This parameter is used to define a rule condition.

SenderADAttributeContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *SenderADAttributeContainsWords* parameter specifies one or more words to check for in specific Active Directory attributes of the sender. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes**

			• **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the words. For example, to look for the word `Sales` in the `Department` attribute, set this parameter to `Department:Sales`. If you want to specify multiple values for multiple attributes, separate them with commas. The rule will be applied if any of the specified attributes have the value specified. This parameter is used to define a rule condition.
SenderADAttributeMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *SenderADAttributeMatchesPatterns* parameter specifies one or more patterns to check for in the specified Active Directory attribute of the sender.

			You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an

			Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the pattern. If you want to specify multiple patterns for multiple attributes, separate them with commas. The rule will be applied if the values of any of the specified attributes match the specified patterns for that attribute. This parameter is used to define a rule condition.
SenderAddressLocation	Optional	Microsoft.Exchange.MessagingPolicies.Rules.SenderAddressLocation	The *SenderAddressLocation* parameter specifies the fields to look in when evaluating sender addresses. Prior to Exchange Server 2013 Cumulative Update 1, Transport rules only processed message headers when evaluating senders. With the addition of the *SenderAddressLocation* parameter, you can configure the rules to also examine the message

			envelope (the sender information sent with the MAIL FROM command in the SMTP transmission) when evaluating senders. Valid values are: • `Header` Only message headers are examined when evaluating senders. • `Envelope` Only the message envelope is examined when evaluating senders. • `HeaderOrEnvelope` Both message headers and the message envelope are examined when evaluating senders. The default value is `Header`. **Note:** By configuring this parameter you can evaluate the message envelope for the following conditions and exceptions: • From • FromAddressContainsWords • FromAddressMatchesPatterns • FromMemberOf • SenderDomainIs
SenderDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *SenderDomainIs* parameter specifies the sender's domain. The rule is applied to messages

			received from senders whose email addresses are in the specified domain. This parameter is used to define a rule condition.
SenderInRecipientList	Optional	Microsoft.Exchange.Data.Word[]	This parameter is available only in the cloud-based service. The *SenderInRecipientList* parameter specifies the condition when the sender is defined in a supervision list entry on a recipient's mailbox. Supervision list entries perform the following functions: • They specify individual exceptions for the user in the closed campus supervision policy. • They identify the user as a victim in the anti-bullying policy. To view the supervision list entries that are configured on a user's mailbox, run the **Get-SupervisionListEntry** command and specify the user's mailbox.

			This parameter is used to define a rule condition.
SenderIpRanges	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SenderIpRanges* parameter specifies the IP ranges to compare with the sender's IP address. The rule is applied if the IP address of the sender falls within one of the IP ranges specified in this parameter. This parameter is used to define a rule condition.
SenderManagementRelationship	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ManagementRelationship	The *SenderManagementRelationship* parameter specifies a relationship between the sender and the recipient. Valid values are: • `Manager` The rule is applied if the sender is the manager of the recipient. • `DirectReport` The rule is applied if the sender is a direct report of the recipient. This parameter is used to define a rule condition.
SentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentTo* parameter specifies a recipient. The rule is applied to messages sent to the

			specified recipient. This parameter is used to define a rule condition.
SentToMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentToMemberOf* parameter specifies a distribution group. The rule is applied to messages where any recipient is a member of the specified group. **Note:** If the distribution group is removed after creation of the rule, no action is taken. This parameter is used to define a rule condition.
SentToScope	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ToUserScope	The *SentToScope* parameter specifies whether the message is sent to internal, external, or partner recipients. Valid values are: • `InOrganization` The recipients are internal to your organization. • `NotInOrganization` The recipients are outside your organization. • `ExternalPartner` The recipients are in a partner organization. • `ExternalNonPartner` The recipients are external to your

			organization, which isn't a partner organization. This parameter is used to define a rule condition.
SetAuditSeverity	Optional	System.String	The *SetAuditSeverity* parameter specifies the severity level assigned to an incident report that's generated and the corresponding entry logged in the message tracking logs. You can specify one of the following values: • `DoNotAudit` No audit entry is logged. • `Low` The audit entry is assigned low severity. • `Medium` The audit entry is assigned medium severity. • `High` The audit entry is assigned high severity. This parameter is used to define a rule action.
SetHeaderName	Optional	Microsoft.Exchange.Data.HeaderName	The *SetHeaderName* parameter specifies the SMTP header name to add or modify when the rule is applied. When the *SetHeaderName* parameter is used, you must also use the *SetHeaderValue*

			parameter to specify a value for the header. This parameter is used to define a rule action.
SetHeaderValue	Optional	Microsoft.Exchange.Data.HeaderValue	The *SetHeaderValue* parameter specifies a value for the header specified in the *SetHeaderName* parameter. This parameter is used to define a rule action.
SetSCL	Optional	Microsoft.Exchange.Data.SclValue	The *SetSCL* parameter modifies the SCL value of the message to the value specified. The SCL value can be a number from 0 through 9, or -1. The value -1 specifies that the message is from a trusted source. This parameter is used to define a rule action.
SmtpRejectMessageRejectStatusCode	Optional	Microsoft.Exchange.Data.RejectStatusCode	The *SmtpRejectMessageRejectStatusCode* parameter specifies an enhanced status code to provide when rejecting a message. This parameter is used to

			define a rule action.
SmtpRejectMessageRejectText	Optional	Microsoft.Exchange.Data.RejectText	The *SmtpRejectMessageRejectText* parameter specifies a text string to add to the rejection message. You must use this parameter with the *SmtpRejectMessageRejectStatusCode* parameter. This parameter is used to define a rule action.
StopRuleProcessing	Optional	System.Boolean	The *StopRuleProcessing* parameter specifies whether the processing of subsequent rules should be stopped for this message. This parameter is used to define a rule action.
SubjectContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *SubjectContainsWords* parameter specifies words to look for in the message subject. You can specify one or more words or phrases. When specifying more than one word or phrase, each word or phrase should be separated by a comma (,). When

			specifying a phrase that contains one or more spaces, you must enclose the phrase in quotation marks ("), for example: `word1,"Phrase with spaces",word2`. This parameter is used to define a rule condition.
SubjectMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *SubjectMatchesPatterns* parameter specifies text patterns to check for in the message subject for. This parameter is used to define a rule condition.
SubjectOrBodyContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *SubjectOrBodyContainsWords* parameter specifies words to look for in the message subject or body. The rule is applied if any of the words or phrases specified is found in the message subject or body. You can specify one or more words or phrases. When specifying more than one word or phrase, each word or phrase should be separated by a comma (,). When

			specifying a phrase with one or more spaces, you must enclose the phrase in quotation marks ("), for example: `word1,"Phrase with spaces",word2`. This parameter is used to define a rule condition.
SubjectOrBodyMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *SubjectOrBodyMatchesPatterns* parameter specifies text patterns to look for in the message subject or body. The rule is applied if the word specified is found in the message subject or body. This parameter is used to define a rule condition.
UseLegacyRegex	Optional	System.Boolean	The *UseLegacyRegex* parameter specifies that the new rule uses the regular expressions compatible with Exchange Server 2010.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WithImportance	Optional	Microsoft.Exchange.Management.Tasks.Importance	The *WithImportance* parameter specifies message importance. The rule is applied to messages matching the specified importance. Valid values are: • `High` • `Low` • `Normal` This parameter is used to define a rule condition.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-TransportRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-04-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-TransportRule** cmdlet to remove a transport rule from your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-TransportRule -Identity <RuleIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes a single transport rule.

```
Remove-TransportRule "Redirect messages from
kim@contoso.com to legal@contoso.com"
```

Detailed Description

On Mailbox servers, the cmdlet removes the rule from Active Directory. On an Edge Transport server, the cmdlet removes the rule from the local Active Directory Lightweight Directory Services (AD LDS) instance.

To temporarily disable a transport rule without removing it, use the **Disable-TransportRule** cmdlet instead.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter specifies the transport rule you want to remove. Enter either the name or the GUID of the rule.

			You can omit the parameter label.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance

			of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-TransportRule

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-07-22*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-TransportRule** cmdlet to modify existing transport rules in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-TransportRule -Identity <RuleIdParameter> [-ActivationDate <DateTime>]
[-ADComparisonAttribute <DisplayName | FirstName | Initials | LastName |
Office | PhoneNumber | OtherPhoneNumber | Email | Street | POBox | City |
State | ZipCode | Country | UserLogonName | HomePhoneNumber |
OtherHomePhoneNumber | PagerNumber | MobileNumber | FaxNumber |
OtherFaxNumber | Notes | Title | Department | Company | Manager |
CustomAttribute1 | CustomAttribute2 | CustomAttribute3 | CustomAttribute4
| CustomAttribute5 | CustomAttribute6 | CustomAttribute7 |
CustomAttribute8 | CustomAttribute9 | CustomAttribute10 |
CustomAttribute11 | CustomAttribute12 | CustomAttribute13 |
CustomAttribute14 | CustomAttribute15>] [-ADComparisonOperator <Equal |
NotEqual>] [-AddManagerAsRecipientType <To | Cc | Bcc | Redirect>] [-
AddToRecipients <RecipientIdParameter[]>] [-AnyOfCcHeader
<RecipientIdParameter[]>] [-AnyOfCcHeaderMemberOf <RecipientIdParameter[]
>] [-AnyOfRecipientAddressContainsWords <Word[]>] [-
AnyOfRecipientAddressMatchesPatterns <Pattern[]>] [-AnyOfToCcHeader
<RecipientIdParameter[]>] [-AnyOfToCcHeaderMemberOf
<RecipientIdParameter[]>] [-AnyOfToHeader <RecipientIdParameter[]>] [-
AnyOfToHeaderMemberOf <RecipientIdParameter[]>] [-ApplyClassification
<String>] [-ApplyHtmlDisclaimerFallbackAction <Wrap | Ignore | Reject>] [-
ApplyHtmlDisclaimerLocation <Append | Prepend>] [-ApplyHtmlDisclaimerText
<DisclaimerText>] [-ApplyOME <$true | $false>] [-
ApplyRightsProtectionTemplate <RmsTemplateIdParameter>] [-
AttachmentContainsWords <Word[]>] [-AttachmentExtensionMatchesWords
<Word[]>] [-AttachmentHasExecutableContent <$true | $false>] [-
AttachmentIsPasswordProtected <$true | $false>] [-AttachmentIsUnsupported
<$true | $false>] [-AttachmentMatchesPatterns <Pattern[]>] [-
AttachmentNameMatchesPatterns <Pattern[]>] [-
AttachmentProcessingLimitExceeded <$true | $false>] [-
AttachmentPropertyContainsWords <Word[]>] [-AttachmentSizeOver
<ByteQuantifiedSize>] [-BetweenMemberOf1 <RecipientIdParameter[]>] [-
BetweenMemberOf2 <RecipientIdParameter[]>] [-BlindCopyTo
<RecipientIdParameter[]>] [-Comments <String>] [-Confirm
[<SwitchParameter>]] [-ContentCharacterSetContainsWords <Word[]>] [-CopyTo
<RecipientIdParameter[]>] [-DeleteMessage <$true | $false>] [-Disconnect
<$true | $false>] [-DlpPolicy <String>] [-DomainController <Fqdn>] [-
ExceptIfADComparisonAttribute <DisplayName | FirstName | Initials |
LastName | Office | PhoneNumber | OtherPhoneNumber | Email | Street |
POBox | City | State | ZipCode | Country | UserLogonName | HomePhoneNumber
| OtherHomePhoneNumber | PagerNumber | MobileNumber | FaxNumber |
OtherFaxNumber | Notes | Title | Department | Company | Manager |
CustomAttribute1 | CustomAttribute2 | CustomAttribute3 | CustomAttribute4
| CustomAttribute5 | CustomAttribute6 | CustomAttribute7 |
CustomAttribute8 | CustomAttribute9 | CustomAttribute10 |
CustomAttribute11 | CustomAttribute12 | CustomAttribute13 |
CustomAttribute14 | CustomAttribute15>] [-ExceptIfADComparisonOperator
<Equal | NotEqual>] [-ExceptIfAnyOfCcHeader <RecipientIdParameter[]>] [-
ExceptIfAnyOfCcHeaderMemberOf <RecipientIdParameter[]>] [-
ExceptIfAnyOfRecipientAddressContainsWords <Word[]>] [-
ExceptIfAnyOfRecipientAddressMatchesPatterns <Pattern[]>] [-
ExceptIfAnyOfToCcHeader <RecipientIdParameter[]>] [-
ExceptIfAnyOfToCcHeaderMemberOf <RecipientIdParameter[]>] [-
ExceptIfAnyOfToHeader <RecipientIdParameter[]>] [-
ExceptIfAnyOfToHeaderMemberOf <RecipientIdParameter[]>] [-
ExceptIfAttachmentContainsWords <Word[]>] [-
ExceptIfAttachmentExtensionMatchesWords <Word[]>] [-
ExceptIfAttachmentHasExecutableContent <$true | $false>] [-
ExceptIfAttachmentIsPasswordProtected <$true | $false>] [-
ExceptIfAttachmentIsUnsupported <$true | $false>] [-
ExceptIfAttachmentMatchesPatterns <Pattern[]>] [-
ExceptIfAttachmentNameMatchesPatterns <Pattern[]>] [-
ExceptIfAttachmentProcessingLimitExceeded <$true | $false>] [-
ExceptIfAttachmentPropertyContainsWords <Word[]>] [-
ExceptIfAttachmentSizeOver <ByteQuantifiedSize>] [-
ExceptIfBetweenMemberOf1 <RecipientIdParameter[]>] [-
ExceptIfBetweenMemberOf2 <RecipientIdParameter[]>] [-
ExceptIfContentCharacterSetContainsWords <Word[]>] [-ExceptIfFrom
```

<RecipientIdParameter[]>] [-ExceptIfFromAddressContainsWords <Word[]>] [-
ExceptIfFromAddressMatchesPatterns <Pattern[]>] [-ExceptIfFromMemberOf
<RecipientIdParameter[]>] [-ExceptIfFromScope <InOrganization |
NotInOrganization>] [-ExceptIfHasClassification <String>] [-
ExceptIfHasNoClassification <$true | $false>] [-ExceptIfHasSenderOverride
<$true | $false>] [-ExceptIfHeaderContainsMessageHeader <HeaderName>] [-
ExceptIfHeaderContainsWords <Word[]>] [-ExceptIfHeaderMatchesMessageHeader
<HeaderName>] [-ExceptIfHeaderMatchesPatterns <Pattern[]>] [-
ExceptIfManagerAddresses <RecipientIdParameter[]>] [-
ExceptIfManagerForEvaluatedUser <Sender | Recipient>] [-
ExceptIfMessageContainsDataClassifications <Hashtable[]>] [-
ExceptIfMessageSizeOver <ByteQuantifiedSize>] [-ExceptIfMessageTypeMatches
<OOF | AutoForward | Encrypted | Calendaring | PermissionControlled |
Voicemail | Signed | ApprovalRequest | ReadReceipt>] [-
ExceptIfRecipientADAttributeContainsWords <Word[]>] [-
ExceptIfRecipientADAttributeMatchesPatterns <Pattern[]>] [-
ExceptIfRecipientAddressContainsWords <Word[]>] [-
ExceptIfRecipientAddressMatchesPatterns <Pattern[]>] [-
ExceptIfRecipientDomainIs <Word[]>] [-ExceptIfRecipientInSenderList
<Word[]>] [-ExceptIfSCLOver <SclValue>] [-
ExceptIfSenderADAttributeContainsWords <Word[]>] [-
ExceptIfSenderADAttributeMatchesPatterns <Pattern[]>] [-
ExceptIfSenderDomainIs <Word[]>] [-ExceptIfSenderInRecipientList <Word[]>]
[-ExceptIfSenderIpRanges <MultiValuedProperty>] [-
ExceptIfSenderManagementRelationship <Manager | DirectReport>] [-
ExceptIfSentTo <RecipientIdParameter[]>] [-ExceptIfSentToMemberOf
<RecipientIdParameter[]>] [-ExceptIfSentToScope <InOrganization |
NotInOrganization | ExternalPartner | ExternalNonPartner>] [-
ExceptIfSubjectContainsWords <Word[]>] [-ExceptIfSubjectMatchesPatterns
<Pattern[]>] [-ExceptIfSubjectOrBodyContainsWords <Word[]>] [-
ExceptIfSubjectOrBodyMatchesPatterns <Pattern[]>] [-ExceptIfWithImportance
<Low | Normal | High>] [-ExpiryDate <DateTime>] [-From
<RecipientIdParameter[]>] [-FromAddressContainsWords <Word[]>] [-
FromAddressMatchesPatterns <Pattern[]>] [-FromMemberOf
<RecipientIdParameter[]>] [-FromScope <InOrganization |
NotInOrganization>] [-GenerateIncidentReport <RecipientIdParameter>] [-
GenerateNotification <DisclaimerText>] [-HasClassification <String>] [-
HasNoClassification <$true | $false>] [-HasSenderOverride <$true |
$false>] [-HeaderContainsMessageHeader <HeaderName>] [-HeaderContainsWords
<Word[]>] [-HeaderMatchesMessageHeader <HeaderName>] [-
HeaderMatchesPatterns <Pattern[]>] [-IncidentReportContent
<IncidentReportContent[]>] [-IncidentReportOriginalMail
<IncludeOriginalMail | DoNotIncludeOriginalMail>] [-LogEventText
<EventLogText>] [-ManagerAddresses <RecipientIdParameter[]>] [-
ManagerForEvaluatedUser <Sender | Recipient>] [-
MessageContainsDataClassifications <Hashtable[]>] [-MessageSizeOver
<ByteQuantifiedSize>] [-MessageTypeMatches <OOF | AutoForward | Encrypted
| Calendaring | PermissionControlled | Voicemail | Signed |
ApprovalRequest | ReadReceipt>] [-Mode <Audit | AuditAndNotify | Enforce>]
[-ModerateMessageByManager <$true | $false>] [-ModerateMessageByUser
<RecipientIdParameter[]>] [-Name <String>] [-NotifySender <NotifyOnly |
RejectMessage | RejectUnlessFalsePositiveOverride |
RejectUnlessSilentOverride | RejectUnlessExplicitOverride>] [-
PrependSubject <SubjectPrefix>] [-Priority <Int32>] [-Quarantine <$true |
$false>] [-RecipientADAttributeContainsWords <Word[]>] [-
RecipientADAttributeMatchesPatterns <Pattern[]>] [-
RecipientAddressContainsWords <Word[]>] [-RecipientAddressMatchesPatterns
<Pattern[]>] [-RecipientDomainIs <Word[]>] [-RecipientInSenderList <Word[]
>] [-RedirectMessageTo <RecipientIdParameter[]>] [-
RejectMessageEnhancedStatusCode <RejectEnhancedStatus>] [-
RejectMessageReasonText <DsnText>] [-RemoveHeader <HeaderName>] [-
RemoveOME <$true | $false>] [-RouteMessageOutboundConnector
<OutboundConnectorIdParameter>] [-RouteMessageOutboundRequireTls <$true |
$false>] [-RuleErrorAction <Ignore | Defer>] [-RuleSubType <None | Dlp>]
[-SCLOver <SclValue>] [-SenderADAttributeContainsWords <Word[]>] [-
SenderADAttributeMatchesPatterns <Pattern[]>] [-SenderAddressLocation
<Header | Envelope | HeaderOrEnvelope>] [-SenderDomainIs <Word[]>] [-
SenderInRecipientList <Word[]>] [-SenderIpRanges <MultiValuedProperty>] [-
SenderManagementRelationship <Manager | DirectReport>] [-SentTo
<RecipientIdParameter[]>] [-SentToMemberOf <RecipientIdParameter[]>] [-
SentToScope <InOrganization | NotInOrganization | ExternalPartner |
ExternalNonPartner>] [-SetAuditSeverity <String>] [-SetHeaderName
<HeaderName>] [-SetHeaderValue <HeaderValue>] [-SetSCL <SclValue>] [-
SmtpRejectMessageRejectStatusCode <RejectStatusCode>] [-
SmtpRejectMessageRejectText <RejectText>] [-StopRuleProcessing <$true |
$false>] [-SubjectContainsWords <Word[]>] [-SubjectMatchesPatterns
<Pattern[]>] [-SubjectOrBodyContainsWords <Word[]>] [-
SubjectOrBodyMatchesPatterns <Pattern[]>] [-WhatIf [<SwitchParameter>]] [-

```
WithImportance <Low | Normal | High>]
```

Examples

EXAMPLE 1

This example modifies the `Sales Team Disclaimer` transport rule. Modifying the value of one condition doesn't affect other conditions, exceptions, or actions used in the rule.

This example sets the *FromMemberOf* parameter to a value of `Sales-Group`, which specifies that the rule is applied if the sender of the message is a member of the Sales-Group distribution group.

```
Set-TransportRule "Sales Team Disclaimer" -FromMemberOf "Sales-Group"
```

Detailed Description

Transport rule conditions and exceptions include corresponding values to test for. For a list of supported transport rule conditions, see Transport rule conditions (predicates).

Transport rules apply actions to messages, some with corresponding action values. For a list of supported transport rule actions, see Transport rule actions.

For detailed information about how to create a transport rule, see Manage Transport Rules. To learn more about transport rules, see Transport rules.

In on-premises Exchange organizations, Transport rules created on Mailbox servers are stored in Active Directory. All Mailbox servers in the organization have access to the same set of transport rules. On Edge Transport servers, transport rules are saved in the local copy of Active Directory Lightweight Directory Services (AD LDS). Transport rules aren't shared or replicated between Edge Transport servers or between Mailbox servers and Edge Transport servers. Also, Mailbox servers and Edge Transport servers share a set of common conditions and actions, but some conditions and actions are exclusive to each server role.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.MessagingPolicies.Rules	The *Identity* parameter specifies the transport rule

		.Tasks.RuleIdParameter	you want to modify. Use the name or GUID of the transport rule.
ActivationDate	Optional	System.DateTime	The *ActivationDate* parameter specifies the date when this rule will become effective. The rule won't take any action on messages until the day you specify for this parameter.
ADComparisonAttribute	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ADAttribute	The *ADComparisonAttribute* parameter specifies an Active Directory attribute to compare between the sender and recipients. When you use this parameter, the specified Active Directory attribute of the sender is compared to the same Active Directory attribute of all the recipients of the message. You can use one of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office**

			• **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** When specifying the *ADComparisonAttribute* parameter, if you don't specify a value for the *ADComparisonOperator* parameter, the default comparison operator `Equal` is used. This parameter is used to

			define a rule condition.
ADComparisonOperator	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Evaluation	The *ADComparisonOperator* parameter specifies a comparison operator for the *ADComparisonAttribute* parameter. Valid values include: • `Equal` • `NotEqual` If you use the *ADComparisonOperator* parameter, you must also use the *ADComparisonAttribute* parameter. This parameter is used to define a rule condition.
AddManagerAsRecipientType	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.AddedRecipientType	The *AddManagerAsRecipientType* parameter specifies how the message is relayed to the manager of the sender or recipient. You can use any of the following values: • `To` The manager is added to the recipients in the To line of the message. • `Cc` The manager is added to the recipients in the carbon copy (Cc)

			line of the message. • `Bcc` The manager is added to the recipients in the blind carbon copy (Bcc) line of the message. • `Redirect` The message is redirected to the manager instead of being delivered to the original recipients. This parameter is used to define a rule action.
AddToRecipients	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AddToRecipients* parameter specifies one or more additional recipients for the message. Separate multiple recipients with commas. The specified recipients are added as To recipients. This parameter is used to define a rule action.
AnyOfCcHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfCcHeader* parameter specifies one or more recipients. The rule is applied if any of these recipients are present as a Cc recipient. This parameter is used to define a rule condition.
AnyOfCcHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.Reci	The *AnyOfCcHeaderMemberOf* parameter specifies a

		pientIdParameter[]	distribution group. The rule is applied if a member of the specified distribution group is present as a Cc recipient. This parameter is used to define a rule condition.
AnyOfRecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *AnyOfRecipientAddressContainsWords* parameter specifies one or more words to check in a recipient address. The rule is applied if a recipient's address includes any of these words. This parameter is used to define a rule condition.
AnyOfRecipientAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *AnyOfRecipientAddressMatchesPatterns* parameter specifies one or more regular expressions to match in a recipient address. The rule is applied if any of the recipients' addresses matches the pattern you specify. This parameter is used to define a rule condition.

AnyOfToCcHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfToCcHeader* parameter specifies one or more recipients. The rule is applied if any of the recipients specified are present as a To or Cc recipient. This parameter is used to define a rule condition.
AnyOfToCcHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfToCcHeaderMemberOf* parameter specifies a distribution group. The rule is applied if a member of the specified distribution group is present as a To or Cc recipient. This parameter is used to define a rule condition.
AnyOfToHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfToHeader* parameter specifies one or more recipients. The rule is applied if any of the specified recipients are present as a To recipient. This parameter is used to define a rule condition.
AnyOfToHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *AnyOfToHeaderMemberOf* parameter specifies a

			distribution group. The rule is applied if a member of the specified distribution group is present as a To recipient. This parameter is used to define a rule condition.
ApplyClassification	Optional	System.String	The *ApplyClassification* parameter specifies a message classification to apply to the message. **Note:** The message classification referred to in this parameter is the custom message classification that you can create in your organization. It isn't related to the data loss prevention (DLP) classification. This parameter is used to define a rule action.
ApplyHtmlDisclaimerFallbackAction	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.DisclaimerFallbackAction	The *ApplyHtmlDisclaimerFallbackAction* parameter specifies an action to fall back to if the HTML disclaimer can't be applied to a message. Valid fallback actions include the following: • `Wrap` The original message is wrapped as an attachment in a new

			message, and the disclaimer is used as the message body for the new message. • `Ignore` The rule is ignored, and the message is delivered without the disclaimer. • `Reject` The message is rejected. **Note:** This parameter is used with the *ApplyHtmlDisclaimerText* parameter. If you use the *ApplyHtmlDisclaimerText* parameter without specifying a value for this parameter, the default fallback action, `Wrap`, is used. This parameter is used to define a rule action.
ApplyHtmlDisclaimerLocation	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.DisclaimerLocation	The *ApplyHtmlDisclaimerLocation* parameter specifies the location within the message where the HTML disclaimer text is inserted. You can use either of the following two values: • `Append` The disclaimer is added to the end of the message body. • `Prepend` The disclaimer is inserted to the beginning of the message body. **Note:** This parameter is used with the

			ApplyHtmlDisclaimerText parameter. If you use the *ApplyHtmlDisclaimerText* parameter without specifying a value for this parameter, the default value, `Append`, is used. This parameter is used to define a rule action.
ApplyHtmlDisclaimerText	Optional	Microsoft.Exchange.Data.DisclaimerText	The *ApplyHtmlDisclaimerText* parameter specifies disclaimer text to be inserted in the message. Disclaimer text can include HTML tags and inline cascading style sheet (CSS) tags. You can add images using the IMG tag. This parameter is used to define a rule action.
ApplyOME	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *ApplyOME* parameter specifies that a message and its attachments will be encrypted if the message matches the conditions of this rule. Valid input for this parameter is `$true` or `$false`. The default is

			`$false`. This parameter is used to define a rule action.
ApplyRightsProtectionTemplate	Optional	Microsoft.Exchange.Configuration.Tasks.RmsTemplateIdParameter	The *ApplyRightsProtectionTemplate* parameter specifies the name of a rights management service (RMS) template to apply to the message. This action adds rights protection to the messages that meet the conditions of this rule. To use this action, an Active Directory Rights Management Services (AD RMS) server should exist in the topology or the organization should be configured to use the ILS service. For more information, see Transport protection rules. This parameter is used to define a rule action.
AttachmentContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *AttachmentContainsWords* parameter specifies one or more words to check in attachments. Only

			supported attachment types are checked. The rule is applied if any of the attachments contain any of the words you specify. This parameter is used to define a rule condition.
AttachmentExtensionMatchesWords	Optional	Microsoft.Exchange.Data.Word[]	The *AttachmentExtensionMatchesWords* parameter specifies one or more word patterns to check in attachment extensions. The rule is applied if the extensions of any of the attachments match the word patterns you specify. This parameter is used to define a rule condition.
AttachmentHasExecutableContent	Optional	System.Boolean	The *AttachmentHasExecutableContent* parameter specifies whether the rule is applied when any attachments in the message contain executable content. If you set this parameter to `$true`, the rule is applied if any of the attachments contains executable content.

			This parameter is used to define a rule condition.
AttachmentIsPasswordProtected	Optional	System.Boolean	The *AttachmentIsPasswordProtected* parameter specifies whether the attachment is a password protected file whose contents can't be inspected. For example, if a password protected ZIP file is in a message, this condition will be met. The rule is applied if any attachment is password protected. This parameter is used to define a rule condition.
AttachmentIsUnsupported	Optional	System.Boolean	The *AttachmentIsUnsupported* parameter specifies whether the rule is applied when any attachments in the message are of an unsupported type. Unsupported attachments are attachments for which an IFilter isn't installed on the servers. If you set this parameter to `$true`, the rule is applied if any of the attachments is an unsupported type.

			This parameter is used to define a rule condition.
AttachmentMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *AttachmentMatchesPatterns* parameter specifies one or more regular expressions to match in message attachment content. Only supported attachment types are checked for the specified pattern. **Note:** Only the first 150 kilobytes (KB) of the attachment is scanned when trying to match a pattern. This parameter is used to define a rule condition.
AttachmentNameMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *AttachmentNameMatchesPatterns* parameter specifies one or more word patterns to check in attachment names. The rule is applied if the name of any attachment matches the patterns you specify. This parameter is used to define a rule condition.

AttachmentProcessingLimitExceeded	Optional	System.Boolean	The *AttachmentProcessingLimitExceeded* parameter specifies whether the scanning of attachments in the message didn't complete because the processing exceeded built-in limits. This condition is used to create rules that work together with other attachment processing rules and gives you the ability to handle messages whose content couldn't be fully scanned. Valid values are `$true` and `$false`. This parameter is used to define a rule condition.
AttachmentPropertyContainsWords	Optional	Microsoft.Exchange.Data.Word[]	This parameter is reserved for internal Microsoft use.
AttachmentSizeOver	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *AttachmentSizeOver* parameter specifies an attachment size. The rule is applied if the size of any of the attachments exceeds the specified size. This parameter is used to define a rule condition.
BetweenMemberOf1	Optional	Microsoft.Exchange.Co	The *BetweenMemberOf1*

		nfiguration.Tasks.RecipientIdParameter[]	parameter specifies a distribution group and must be used together with the *BetweenMemberOf2* parameter. The rule is applied if the message is sent between members of the distribution groups specified in these parameters. This parameter is used to define a rule condition.
BetweenMemberOf2	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *BetweenMemberOf2* parameter specifies a distribution group and must be used together with the *BetweenMemberOf1* parameter. The rule is applied if the message is sent between members of the distribution groups specified in these parameters. This parameter is used to define a rule condition.
BlindCopyTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *BlindCopyTo* parameter specifies one or more recipients to add to the message as Bcc recipients.

			This parameter is used to define a rule action.
Comments	Optional	System.String	The *Comments* parameter specifies informative comments for the transport rule, such as what the rule is used for or how it has changed over time. The length of the comment can't exceed 1024 characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ContentCharacterSetContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ContentCharacterSetContainsWords* parameter specifies one or more character set names to check for in the message. The rule is applied if the message contains any of the character sets specified. This parameter is used to

			define a rule condition.
CopyTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *CopyTo* parameter specifies one or more recipients to add to the message as Cc recipients. This parameter is used to define a rule action.
DeleteMessage	Optional	System.Boolean	The *DeleteMessage* parameter specifies that the rule deletes any message that matches the conditions specified. This parameter is used to define a rule action.
Disconnect	Optional	System.Boolean	The *Disconnect* parameter specifies whether the rules agent disconnects the SMTP session. This parameter is used to define a rule action.
DlpPolicy	Optional	System.String	The *DlpPolicy* parameter specifies the data loss prevention (DLP) policy associated with this rule. Each DLP policy is enforced using a set of transport rules. To learn more about DLP, see Data loss prevention.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
ExceptIfADComparisonAttribute	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ADAttribute	The *ExceptIfADComparisonAttribute* parameter specifies an Active Directory attribute to compare between the sender and recipients. When you use this parameter, the specified Active Directory attribute of the sender is compared to the same Active Directory attribute of all the recipients of the

			message. You can use one of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** When specifying the

			ExceptIfADComparisonAttribute parameter, if you don't specify a value for the *ExceptIfADComparisonOperator* parameter, the default comparison operator `Equal` is used.
ExceptIfADComparisonOperator	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Evaluation	The *ExceptIfADComparisonOperator* parameter specifies a comparison operator for the *ExceptIfADComparisonAttribute* parameter. Valid values are: • `Equal` • `NotEqual` If you use the *ExceptIfADComparisonOperator* parameter, you must also use the *ExceptIfADComparisonAttribute* parameter.
ExceptIfAnyOfCcHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfAnyOfCcHeader* parameter specifies one or more recipients. The rule isn't applied if any of these recipients are present as a Cc recipient.
ExceptIfAnyOfCcHead	Optional	Microsoft.Exchange.Co	The

erMemberOf		nfiguration.Tasks.RecipientIdParameter[]	*ExceptIfAnyOfCcHeaderMemberOf* parameter specifies a distribution group. The rule isn't applied if a member of the specified distribution group is present as a Cc recipient.
ExceptIfAnyOfRecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfAnyOfRecipientAddressContainsWords* parameter specifies one or more words to check in a recipient address. The rule isn't applied if a recipient's address includes any of these words.
ExceptIfAnyOfRecipientAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfAnyOfRecipientAddressMatchesPatterns* parameter specifies one or more regular expressions to match in a recipient address. The rule isn't applied if any of the recipient addresses matches the pattern you specify.
ExceptIfAnyOfToCcHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfAnyOfToCcHeader* parameter specifies one or

			more recipients. The rule isn't applied if any of the recipients specified are present as a To or Cc recipient.
ExceptIfAnyOfToCcHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfAnyOfToCcHeaderMemberOf* parameter specifies a distribution group. The rule isn't applied if a member of the specified distribution group is present as a To or Cc recipient.
ExceptIfAnyOfToHeader	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfAnyOfToHeader* parameter specifies one or more recipients. The rule isn't applied if any of the specified recipients are present as a To recipient.
ExceptIfAnyOfToHeaderMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfAnyOfToHeaderMemberOf* parameter specifies a distribution group. The rule isn't applied if a member of the specified distribution group is present as a To recipient.
ExceptIfAttachmentCo	Optional	Microsoft.Exchange.Da	The

ntainsWords		ta.Word[]	*ExceptIfAttachmentContainsWords* parameter specifies one or more words to check in attachments. Only supported attachment types are checked. The rule isn't applied if any of the attachments contain any of the words you specify.
ExceptIfAttachmentExtensionMatchesWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfAttachmentExtensionMatchesWords* parameter specifies one or more word patterns to check in attachment extensions. The rule isn't applied if the extensions of any of the attachments match the word patterns you specify.
ExceptIfAttachmentHasExecutableContent	Optional	System.Boolean	The *ExceptIfAttachmentHasExecutableContent* parameter specifies whether the rule is applied when any attachments in the message contain executable content. If you set this parameter to `$true`, the rule isn't

			applied if any of the attachments contains executable content.
ExceptIfAttachmentIsPasswordProtected	Optional	System.Boolean	The *ExceptIfAttachmentIsPasswordProtected* parameter specifies whether the attachment is a password protected file whose contents can't be inspected. For example, if a password protected ZIP file is in a message, this exception will be met. The rule isn't applied if any attachment is password protected.
ExceptIfAttachmentIsUnsupported	Optional	System.Boolean	The *ExceptIfAttachmentIsUnsupported* parameter specifies whether the rule is applied when any attachments in the message are of an unsupported type. Unsupported attachments are attachments for which an IFilter isn't installed on your servers. If you set this parameter to `$true` the rule isn't applied if any of the attachments is an

			unsupported type.
ExceptIfAttachmentMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfAttachmentMatchesPatterns* parameter specifies one or more regular expressions to match in message attachment content. Only supported attachment types are checked for the specified pattern. **Note:** Only the first 150 KB of the attachment is scanned when trying to match a pattern.
ExceptIfAttachmentNameMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfAttachmentNameMatchesPatterns* parameter specifies one or more word patterns to check in attachment names. The rule isn't applied if the name of any attachment matches the patterns you specify.
ExceptIfAttachmentProcessingLimitExceeded	Optional	System.Boolean	The *ExceptIfAttachmentProcessingLimitExceeded* parameter specifies whether the scanning of attachments in the message didn't complete

			because the processing exceeded built-in limits. This condition is used to create rules that work together with other attachment processing rules and gives you the ability to handle messages whose content couldn't be fully scanned. Valid values are `$true` and `$false`.
ExceptIfAttachmentPropertyContainsWords	Optional	Microsoft.Exchange.Data.Word[]	This parameter is reserved for internal Microsoft use.
ExceptIfAttachmentSizeOver	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ExceptIfAttachmentSizeOver* parameter specifies an attachment size. The rule isn't applied if the size of any of the attachments exceeds the specified size.
ExceptIfBetweenMemberOf1	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfBetweenMemberOf1* parameter specifies a distribution group and must be used together with the *ExceptIfBetweenMemberOf2* parameter. The rule isn't applied if the message is sent between

			members of the distribution groups specified in these parameters.
ExceptIfBetweenMemberOf2	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfBetweenMemberOf2* parameter specifies a distribution group and must be used together with the *ExceptIfBetweenMemberOf1* parameter. The rule isn't applied if the message is sent between members of the distribution groups specified in these parameters.
ExceptIfContentCharacterSetContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfContentCharacterSetContainsWords* parameter specifies one or more character set names to check for in the message. The rule isn't applied if the message contains any of the character sets specified.
ExceptIfFrom	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfFrom* parameter specifies the sender. The rule isn't

			applied to messages received from this sender.
ExceptIfFromAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfFromAddressContainsWords* parameters specifies one or more words to check for in the From address. The rule isn't applied if the sender's address includes any of these words.
ExceptIfFromAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfFromAddressMatchesPatterns* parameter specifies one or more regular expressions to match in the sender's address. The rule isn't applied if the sender's address matches the pattern you specify.
ExceptIfFromMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfFromMemberOf* parameter specifies a distribution group. The rule isn't applied if the sender of the message is a member of this distribution group.
ExceptIfFromScope	Optional	Microsoft.Exchange.MessagingPolicies.Rules	The *ExceptIfFromScope* parameter specifies

		.Tasks.FromUserScope	whether the sender is inside or outside your organization. Valid values for this parameter are: • `InOrganization` • `NotInOrganization`
ExceptIfHasClassification	Optional	System.String	The *ExceptIfHasClassification* parameter specifies a message classification. The rule isn't applied to messages that have the specified classification. **Note:** The message classification referred to in this parameter is the custom message classification that you can create in your organization. It isn't related to the DLP classification.
ExceptIfHasNoClassification	Optional	System.Boolean	The *ExceptIfHasNoClassification* parameter specifies that the rule isn't applied to messages that don't have a message classification.
ExceptIfHasSenderOverride	Optional	System.Boolean	The *ExceptIfHasSenderOverride* parameter specifies the rule to check if the sender has chosen to override a DLP policy. Set this parameter to `$true` to

			prevent this rule from applying to messages where the sender took action to override a DLP policy restriction.
ExceptIfHeaderContainsMessageHeader	Optional	Microsoft.Exchange.Data.HeaderName	The *ExceptIfHeaderContainsMessageHeader* parameter specifies the SMTP message header to inspect for specific words or patterns. This parameter is used together with the *ExceptIfHeaderContainsWords* parameter.
ExceptIfHeaderContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfHeaderContainsWords* parameter specifies one or more words to look for in the message header specified in the *ExceptIfHeaderContainsMessageHeader* parameter. The rule isn't applied to messages where the header value of the specified header matches any of the words specified.
ExceptIfHeaderMatchesMessageHeader	Optional	Microsoft.Exchange.Data.HeaderName	The *ExceptIfHeaderMatchesMessageHeader* parameter

			specifies an SMTP message header to inspect. This parameter is used together with the *ExceptIfHeaderMatchesPatterns* parameter.
ExceptIfHeaderMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfHeaderMatchesPatterns* parameter specifies a pattern to match in the header specified in the *ExceptIfHeaderMatchesMessageHeader* parameter.
ExceptIfManagerAddresses	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfManagerAddresses* parameter specifies a recipient. The rule isn't applied to messages where the specified recipient is the manager of the sender or the recipient. Whether it's the manager for the sender or the recipient is defined in the *ExceptIfManagerForEvaluatedUser* parameter.
ExceptIfManagerForEvaluatedUser	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.EvaluatedUser	The *ExceptIfManagerForEvaluatedUser* parameter specifies whether the sender or the recipient's manager should be

			evaluated. The specified user's manager attribute is compared with users specified in the *ExceptIfManagerAddresses* parameter. Valid values include: • `Recipient` • `Sender` Use this parameter together with the *ExceptIfManagerAddresses* parameter.
ExceptIfMessageContainsDataClassifications	Optional	System.Collections.Hashtable[]	The *ExceptIfMessageContainsDataClassifications* parameter specifies the sensitive information types to look for in the message body and any of the attachments. For a list of sensitive information types available, see Sensitive information types inventory.
ExceptIfMessageSizeOver	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *ExceptIfMessageSizeOver* parameter specifies a message size. The rule isn't applied to any messages that exceed the message size you specify for this parameter.

ExceptIfMessageTypeMatches	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.MessageType	The *ExceptIfMessageTypeMatches* parameter specifies a message type. The rule isn't applied to any messages that match the message type you specify. Valid values include: • `OOF` Auto-reply messages configured by the user • `AutoForward` Messages automatically forwarded to an alternative recipient • `Encrypted` Encrypted messages • `Calendaring` Meeting requests and responses • `PermissionControlled` Messages that have specific permissions configured • `Voicemail` Voice mail messages forwarded by Unified Messaging service • `Signed` Digitally signed messages • `ApprovalRequest` Moderation request messages sent to moderators • `ReadReceipt` Read receipts
ExceptIfRecipientADAttributeContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfRecipientADAttributeContainsWords* parameter specifies one or

			more words to check for in specific Active Directory attributes of the recipient. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager**

			• **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the words. For example, to look for the word `Sales` in the `Department` attribute, set this parameter to `Department:Sales`. If you want to specify multiple values for multiple attributes, separate them with commas. The rule isn't applied if any of the specified attributes have the value specified.
ExceptIfRecipientADAttributeMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfRecipientADAttributeMatchesPatterns* parameter specifies one or more patterns to check for in the specified Active Directory attribute of the recipient. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials**

			• **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the pattern. If you want to specify multiple patterns

			for multiple attributes, separate them with commas. The rule isn't applied if the values of any of the specified attributes match the specified patterns for that attribute.
ExceptIfRecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfRecipientAddressContainsWords* parameter specifies words to check for in the recipient address.
ExceptIfRecipientAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfRecipientAddressMatchesPatterns* parameter specifies one or more text patterns to match in the recipient address.
ExceptIfRecipientDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfRecipientDomainIs* parameter specifies the recipient's domain. The rule isn't applied to messages sent to recipients whose email addresses are in the specified domain.
ExceptIfRecipientInSenderList	Optional	Microsoft.Exchange.Data.Word[]	This parameter is available only in the cloud-based service.

			The *ExceptIfRecipientInSenderList* parameter specifies an exception when a recipient is defined in a supervision list entry on the sender's mailbox. Supervision list entries perform the following functions: • They specify individual exceptions for the user in the closed campus supervision policy. • They identify the user as a victim in the anti-bullying policy. To view the supervision list entries that are configured on a user's mailbox, run the **Get-SupervisionListEntry** command and specify the user's mailbox.
ExceptIfSCLOver	Optional	Microsoft.Exchange.Data.SclValue	The *ExceptIfSCLOver* parameter specifies a spam confidence level (SCL) value. The rule isn't applied to messages with an SCL equal to or higher than the value specified. Valid SCL values are integers from 0 through 9,

			and -1. The value -1 specifies that the message is from a trusted source.
ExceptIfSenderADAttributeContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfSenderADAttributeContainsWords* parameter specifies one or more words to check for in specific Active Directory attributes of the sender. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber**

			• **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the words. For example, to look for the word `Sales` in the `Department` attribute, set this parameter to `Department:Sales`. If you want to specify multiple values for multiple attributes, separate them with commas. The rule isn't applied if any of the specified attributes have the value specified.
ExceptIfSenderADAttributeMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfSenderADAttributeMatchesPatterns* parameter specifies one or more patterns to check for in the specified Active Directory attribute of the

			sender. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15**

			To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the pattern. If you want to specify multiple patterns for multiple attributes, separate them with commas. The rule isn't applied if the values of any of the specified attributes match the specified patterns for that attribute.
ExceptIfSenderDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfSenderDomainIs* parameter specifies the sender's domain. The rule isn't applied to messages received from senders whose email addresses are in the specified domain.
ExceptIfSenderInRecipientList	Optional	Microsoft.Exchange.Data.Word[]	This parameter is available only in the cloud-based service. The *ExceptIfSenderInRecipientList* parameter specifies an exception when the sender is defined in a supervision list entry on a recipient's

			mailbox. Supervision list entries perform the following functions: • They specify individual exceptions for the user in the closed campus supervision policy. • They identify the user as a victim in the anti-bullying policy. To view the supervision list entries that are configured on a user's mailbox, run the **Get-SupervisionListEntry** command and specify the user's mailbox.
ExceptIfSenderIpRanges	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExceptIfSenderIpRanges* parameter specifies the IP ranges to compare with the sender's IP address. The rule isn't applied if the IP address of the sender falls within one of the IP ranges specified in this parameter.
ExceptIfSenderManagementRelationship	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ManagementRelationship	The *ExceptIfSenderManagementRelationship* parameter specifies a relationship between the sender and

			the recipient. Valid values are: • `Manager` The rule isn't applied if the sender is the manager of the recipient. • `DirectReport` The rule isn't applied if the sender is a direct report of the recipient.
ExceptIfSentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfSentTo* parameter specifies a recipient. The rule isn't applied to messages sent to the specified recipient.
ExceptIfSentToMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ExceptIfSentToMemberOf* parameter specifies a distribution group. The rule isn't applied to messages where any recipient is a member of the specified group. **Note:** If the distribution group is removed after creation of the rule, no exception is made.
ExceptIfSentToScope	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ToUserScope	The *ExceptIfSentToScope* parameter specifies whether the message is sent to internal, external, or partner recipients. Valid values are: • `InOrganization` The

			recipients are internal to your organization. • `NotInOrganization` The recipients are outside your organization. • `ExternalPartner` The recipients are in a partner organization. • `ExternalNonPartner` The recipients are external to your organization which isn't a partner organization.
ExceptIfSubjectContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfSubjectContainsWords* parameter specifies words to look for in the message subject. You can specify one or more words or phrases. When specifying more than one word or phrase, each word or phrase should be separated by a comma (,). When specifying a phrase that contains one or more spaces, you must enclose the phrase in quotation marks ("), for example: `word1,"Phrase with spaces",word2`.
ExceptIfSubjectMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfSubjectMatchesPatterns* parameter specifies

			text patterns to check the message subject for.
ExceptIfSubjectOrBodyContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *ExceptIfSubjectOrBodyContainsWords* parameter specifies words to look for in the message subject and body. The rule isn't applied if any of the words or phrases specified is found in the message subject or body. You can specify one or more words or phrases. When specifying more than one word or phrase, each word or phrase should be separated by a comma (,). When specifying a phrase with one or more spaces, you must enclose the phrase in quotation marks ("), for example: `word1,"Phrase with spaces",word2`.
ExceptIfSubjectOrBodyMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *ExceptIfSubjectOrBodyMatchesPatterns* parameter specifies text patterns to look for in the message subject and body. The rule

			isn't applied if the word specified is found in the message subject or body.
ExceptIfWithImportance	Optional	Microsoft.Exchange.Management.Tasks.Importance	The *ExceptIfWithImportance* parameter specifies message importance. The rule isn't applied to messages matching the specified importance. Valid values are: • `High` • `Low` • `Normal`
ExpiryDate	Optional	System.DateTime	The *ExpiryDate* parameter specifies the date when this rule will stop processing. The rule won't take any action on messages past the date you specify for this parameter.
From	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *From* parameter specifies the sender. The rule is applied to messages received from this sender. This parameter is used to define a rule condition.
FromAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *FromAddressContainsWords* parameter specifies one

			or more words to check for in the From address. The rule is applied if the sender's address includes any of these words. This parameter is used to define a rule condition.
FromAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *FromAddressMatchesPatterns* parameter specifies one or more regular expressions to match in the sender's address. The rule is applied if the sender's address matches the pattern you specify. This parameter is used to define a rule condition.
FromMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *FromMemberOf* parameter specifies a distribution group. The rule is applied if the sender of the message is a member of this distribution group. This parameter is used to define a rule condition.
FromScope	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.FromUserScope	The *FromScope* parameter specifies whether the sender is inside or outside your organization. Valid

			values for this parameter are: • `InOrganization` • `NotInOrganization` This parameter is used to define a rule condition.
GenerateIncidentReport	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *GenerateIncidentReport* parameter specifies the recipient to whom incident reports will be sent. An incident report is generated for messages that violate a DLP policy in your organization. This parameter is used to define a rule action.
GenerateNotification	Optional	Microsoft.Exchange.Data.DisclaimerText	This parameter is reserved for internal Microsoft use.
HasClassification	Optional	System.String	The *HasClassification* parameter specifies a message classification. The rule is applied to messages that have the specified classification. **Note:** The message classification referred to in this parameter is the custom message classification that you can create in your organization. It isn't related to the DLP classification.

			This parameter is used to define a rule condition.
HasNoClassification	Optional	System.Boolean	The *HasNoClassification* parameter specifies whether the rule is applied to messages that don't have a message classification. If you set this parameter to `$true`, the rule is applied to all messages that don't have a message classification. If you set this parameter to `$false`, the rule is applied to all messages that have one or more message classifications. This parameter is used to define a rule condition.
HasSenderOverride	Optional	System.Boolean	The *HasSenderOverride* parameter specifies the rule to check if the sender has chosen to override a DLP policy. Set this parameter to `$true` to apply this rule to messages where the sender took action to override a DLP policy restriction.

			This parameter is used to define a rule condition.
HeaderContainsMessageHeader	Optional	Microsoft.Exchange.Data.HeaderName	The *HeaderContainsMessageHeader* parameter specifies the SMTP message header to inspect for specific words or patterns. This parameter is used together with the *HeaderContainsWords* parameter. This parameter is used to define a rule condition.
HeaderContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *HeaderContainsWords* parameter specifies one or more words to look for in the message header specified in the *HeaderContainsMessageHeader* parameter. The rule is applied to messages where the header value of the specified header matches any of the words specified. This parameter is used to define a rule condition.
HeaderMatchesMessageHeader	Optional	Microsoft.Exchange.Data.HeaderName	The *HeaderMatchesMessageH*

			eader parameter specifies an SMTP message header to inspect. This parameter is used together with the *HeaderMatchesPatterns* parameter. This parameter is used to define a rule condition.
HeaderMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *HeaderMatchesPatterns* parameter specifies a pattern to match in the header specified in the *HeaderMatchesMessageHeader* parameter. This parameter is used to define a rule condition.
IncidentReportContent	Optional	Microsoft.Exchange.MessagingPolicies.Rules.IncidentReportContent[]	The *IncidentReportContent* parameter specifies the message properties that are included in incident reports. This parameter is used together with the *GenerateIncidentReport* parameter. The valid values are: • `Sender` Includes the sender of the message. • `Recipients` Includes the recipients in the To: box of the message. Only the first 10 recipients are displayed

			in the incident report. If there are more than 10 recipients, the remaining number of recipients will be displayed. • `Subject` Includes the message subject. • `cc` Includes the recipients in the Cc: box of the message. Only the first 10 recipients are displayed in the incident report. If there are more than 10 recipients, the remaining number of recipients will be displayed. • `Bcc` Includes the recipients in the Bcc: box of the message. Only the first 10 recipients are displayed in the incident report. If there are more than 10 recipients, the remaining number of recipients will be displayed. • `Severity` Includes the audit severity of the rule that was triggered. If the message was processed by more than one rule, the highest severity is displayed. • `Override` Includes the override if the sender has chosen to override a PolicyTip. If the sender has provided a justification, the first 100 characters of the justification is also included. • `RuleDetections`

			Includes the list of rules that the message triggered. • `FalsePositive` Includes the false positive if the sender marked the message as a false positive for a PolicyTip. • `DataClassifications` Includes the list of sensitive information types detected in the message. • `IdMatch` Includes the sensitive information type that is detected, the exact matched content from the message, and the 150 characters before and after the matched sensitive information. • `AttachOriginalMail` Includes the entire original message. **Note:** The message ID is always included in the incident report. This parameter is used to define a rule action.
IncidentReportOriginalMail	Optional	Microsoft.Exchange.MessagingPolicies.Rules.IncidentReportOriginalMail	The *IncidentReportOriginalMail* parameter specifies whether to include the original message with the incident report. This parameter is used together with the

			GenerateIncidentReport parameter. Valid values are: • `IncludeOriginalMail` • `DoNotIncludeOriginalMail` The default value is `DoNotIncludeOriginalMail`. This parameter is used to define a rule action. **Important:** The functionality of this parameter is now managed by the *IncidentReportContent* parameter, and this parameter will be deprecated in the future. Adding the value `AttachOriginalMail` to the *IncidentReportContent* parameter is equivalent to setting this parameter to `IncludeOriginalMail` value. Even though this parameter is still functional, we recommend you use the *IncidentReportContent* parameter instead.
LogEventText	Optional	Microsoft.Exchange.Data.EventLogText	The *LogEventText* parameter specifies a message string to add to the event log entry for this rule. This parameter is used to define a rule action.
ManagerAddresses	Optional	Microsoft.Exchange.Co	The *ManagerAddresses*

		nfiguration.Tasks.RecipientIdParameter[]	parameter specifies a recipient. The rule is applied to messages where the specified recipient is the manager of the sender or the recipient. Whether it's the manager for the sender or the recipient is defined in the *ManagerForEvaluatedUser* parameter. This parameter is used to define a rule condition.
ManagerForEvaluatedUser	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.EvaluatedUser	The *ManagerForEvaluatedUser* parameter specifies whether the sender or the recipient's manager should be evaluated. The specified user's manager attribute is compared with users specified in the *ManagerAddresses* parameter. Valid values include: • `Recipient` • `Sender` Use this parameter together with the *ManagerAddresses* parameter. This parameter is used to define a rule condition.

MessageContainsDataClassifications	Optional	System.Collections.Hashtable[]	The *MessageContainsDataClassifications* parameter specifies the sensitive information types to look for in the message body and any of the attachments. For a list of sensitive information types available, see Sensitive information types inventory. This parameter is used to define a rule condition.
MessageSizeOver	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *MessageSizeOver* parameter specifies a message size. The rule is applied to all messages that exceed the message size you specify for this parameter. This parameter is used to define a rule condition.
MessageTypeMatches	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.MessageType	The *MessageTypeMatches* parameter specifies a message type. The rule is applied to all messages that match the message type you specify. Valid values include: • OOF Auto-reply messages configured by

			the user • `AutoForward` Messages automatically forwarded to an alternative recipient • `Encrypted` Encrypted messages • `Calendaring` Meeting requests and responses • `PermissionControlled` Messages that have specific permissions configured • `Voicemail` Voice mail messages forwarded by Unified Messaging service • `Signed` Digitally signed messages • `ApprovalRequest` Moderation request messages sent to moderators • `ReadReceipt` Read receipts This parameter is used to define a rule condition.
Mode	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleMode	The *Mode* parameter specifies in which mode this rule will operate. Valid values include: • `Audit` The rule is turned on, and what would have happened if the rule was enforced is logged in message tracking logs. Exchange doesn't take any action that impacts the delivery of the message. • `AuditAndNotify` The

			rule is turned on, and it operates the same way it would in `Audit` mode, but notifications are also enabled. • `Enforce` The rule is turned on, and all actions specified in the rule are taken. The default value is `Enforce`.
ModerateMessageByManager	Optional	System.Boolean	The *ModerateMessageByManager* parameter specifies whether the message should be forwarded to the sender's manager for approval. To enable moderation by the sender's manager, set the value to `$true`. This parameter is used to define a rule action.
ModerateMessageByUser	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *ModerateMessageByUser* parameter specifies a recipient to forward the message to for approval. This parameter is used to define a rule action.
Name	Optional	System.String	The *Name* parameter specifies the display name of the transport rule to be

			created. The length of the name can't exceed 64 characters.
NotifySender	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.NotifySenderType	The *NotifySender* parameter specifies how the sender of a message that goes against a DLP policy is notified. Valid values are: • `NotifyOnly` Sender is notified, but the message is delivered normally. • `RejectMessage` Message is rejected, and the sender is notified. • `RejectUnlessFalsePositiveOverride` Message is rejected unless it's marked as a false positive by the sender. • `RejectUnlessSilentOverride` Message is rejected unless the sender has chosen to override the policy restriction. • `RejectUnlessExplicitOverride` This is the same as `RejectUnlessSilentOverride`, but the sender can also provide a justification for overriding the policy restriction. If you specify any value other than `NotifyOnly`, you can provide a specific

			rejection status code and reason using the *RejectMessageEnhancedStatusCode* and *RejectMessageReasonText* parameters. This action is used together with the *MessageContainsDataClassifications* condition. If you use this parameter, you must also specify the sensitive information types you want to check against using the *MessageContainsDataClassifications* parameter. This parameter is used to define a rule action.
PrependSubject	Optional	Microsoft.Exchange.Data.SubjectPrefix	The *PrependSubject* parameter specifies a word or phrase to add to the beginning of the message subject. **Note:** The rule will add the text as you specify in this parameter without adding spaces or other characters to separate it from the original subject. Consider ending the value you specify in this parameter with a colon (:) and a space, or at least a space,

			to separate it from the original subject. This parameter is used to define a rule action.
Priority	Optional	System.Int32	The *Priority* parameter specifies the priority for this transport rule. Rules with a lower priority value are processed first. If you modify the priority of the rule, the position of the rule in the rule list changes to match the priority that you specified, and the Transport Rules agent increments all rules with a higher priority value. The value of this parameter must be greater than or equal to 0, and must be one less than the total number of transport rules in your organization. For example, if you configured 8 transport rules, you can set this parameter to any value from 0 through 7.
Quarantine	Optional	System.Boolean	The *Quarantine* parameter specifies whether the rules agent delivers the message to the quarantine

			mailbox specified in the Content Filtering configuration. This parameter is used to define a rule action.
RecipientADAttributeContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *RecipientADAttributeContainsWords* parameter specifies one or more words to check for in specific Active Directory attributes of the recipient. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNu**

			mber • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the words. For example, to look for the word `Sales` in the `Department` attribute, set this parameter to `Department:Sales`. If you want to specify multiple values for multiple attributes, separate them with commas. The rule will be applied if any of the specified attributes have the value specified. This parameter is used to define a rule condition.
RecipientADAttribute	Optional	Microsoft.Exchange.M	The

MatchesPatterns		essagingPolicies.Rules.Tasks.Pattern[]	*RecipientADAttributeMatchesPatterns* parameter specifies one or more patterns to check for in the specified Active Directory attribute of the recipient. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title**

			• **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the pattern. If you want to specify multiple patterns for multiple attributes, separate them with commas. The rule will be applied if the values of any of the specified attributes match the specified patterns for that attribute. This parameter is used to define a rule condition.
RecipientAddressContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *RecipientAddressContainsWords* parameter specifies one or more words to check for in the recipient's email address. This parameter is used to define a rule condition.
RecipientAddressMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules	The *RecipientAddressMatchesPatterns* parameter

		.Tasks.Pattern[]	specifies a pattern to check the recipient address for. This parameter is used to define a rule condition.
RecipientDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *RecipientDomainIs* parameter specifies the recipient's domain. The rule is applied to messages sent to recipients whose email addresses are in the specified domain. This parameter is used to define a rule condition.
RecipientInSenderList	Optional	Microsoft.Exchange.Data.Word[]	This parameter is available only in the cloud-based service. The *RecipientInSenderList* parameter specifies the condition when a recipient is defined in a supervision list entry on the sender's mailbox. Supervision list entries perform the following functions: • They specify individual exceptions for the user in the closed campus supervision policy. • They identify the user as

			a victim in the anti-bullying policy. To view the supervision list entries that are configured on a user's mailbox, run the **Get-SupervisionListEntry** command and specify the user's mailbox. This parameter is used to define a rule condition.
RedirectMessageTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *RedirectMessageTo* parameter specifies that the rule redirects the message to the specified recipient. This parameter is used to define a rule action.
RejectMessageEnhancedStatusCode	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RejectEnhancedStatus	The *RejectMessageEnhancedStatusCode* parameter specifies an enhanced status code to provide when rejecting messages. Valid values are `5.7.1` or between `5.7.10` and `5.7.999`. **Note:** The transport rule can add a custom rejection message. To further customize the delivery status notification (DSN),

			you need to create a custom DSN message using the **New-SystemMessage** cmdlet. If an enhanced status code isn't specified, and only the *RejectMessageReasonText* parameter is used, the enhanced status code `5.7.1` is used. This parameter is used to define a rule action.
RejectMessageReasonText	Optional	Microsoft.Exchange.Data.DsnText	The *RejectMessageReasonText* parameter specifies a reason that explains why the message was rejected. **Note:** The transport rule can add a custom rejection message. To further customize the DSN, you need to create a custom DSN message using the **New-SystemMessage** cmdlet. If a *RejectMessageReasonText* parameter value isn't specified, and an enhanced status code is specified by using the *RejectMessageEnhancedStatusCode* parameter, the

			default reason text "`Delivery not authorized, message refused`" is used. This parameter is used to define a rule action.
RemoveHeader	Optional	Microsoft.Exchange.Data.HeaderName	The *RemoveHeader* parameter specifies an SMTP header name to be removed from the message. This parameter is used to define a rule action.
RemoveOME	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *RemoveOME* parameter specifies that a message and its attachments will be decrypted if they match the identified transport rule. The default is $false.
RouteMessageOutboundConnector	Optional	Microsoft.Exchange.Configuration.Tasks.OutboundConnectorIdParameter	This parameter is available only in the cloud-based service. The *RouteMessageOutboundConnector* parameter specifies the connector to

			use for routing this message. This parameter is used to define a rule action.
RouteMessageOutboundRequireTls	Optional	System.Boolean	The *RouteMessageOutboundRequireTls* parameter specifies that Transport Layer Security (TLS) encryption is required when routing this message outside your organization. Set this parameter to `$true` to require TLS. This parameter is used to define a rule action.
RuleErrorAction	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleErrorAction	The *RuleErrorAction* parameter specifies how the message should be handled if the rule processing can't be completed. Valid values are: • `Ignore` The message is sent without completing the rule processing. • `Defer` The message is deferred so the rules engine can attempt to process the message again. The default value is

			`Ignore`.
RuleSubType	Optional	Microsoft.Exchange.MessagingPolicies.Rules.RuleSubType	The *RuleSubType* parameter specifies the type of this transport rule. Valid values are: • `Dlp` Specifies that this rule is associated with a DLP policy. • `None` Specifies that this is a regular transport rule.
SCLOver	Optional	Microsoft.Exchange.Data.SclValue	The *SCLOver* parameter specifies an SCL value. The rule is applied to messages with an SCL equal to or higher than the value specified. Valid SCL values are integers from 0 through 9, and -1. The value -1 specifies that the message is from a trusted source. This parameter is used to define a rule condition.
SenderADAttributeContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *SenderADAttributeContainsWords* parameter specifies one or more words to check for in specific Active Directory attributes of the sender. You can check against any of the following Active

			Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber** • **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active

			Directory attribute followed by a colon and the words. For example, to look for the word `Sales` in the `Department` attribute, set this parameter to `Department:Sales`. If you want to specify multiple values for multiple attributes, separate them with commas. The rule will be applied if any of the specified attributes have the value specified. This parameter is used to define a rule condition.
SenderADAttributeMatchesPatterns	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.Pattern[]	The *SenderADAttributeMatchesPatterns* parameter specifies one or more patterns to check for in the specified Active Directory attribute of the sender. You can check against any of the following Active Directory attributes: • **DisplayName** • **FirstName** • **Initials** • **LastName** • **Office** • **PhoneNumber**

			• **OtherPhoneNumber** • **Email** • **Street** • **POBox** • **City** • **State** • **ZipCode** • **Country** • **UserLogonName** • **HomePhoneNumber** • **OtherHomePhoneNumber** • **PagerNumber** • **MobileNumber** • **FaxNumber** • **OtherFaxNumber** • **Notes** • **Title** • **Department** • **Company** • **Manager** • **CustomAttribute1 - CustomAttribute15** To specify a value for an Active Directory attribute, use the name of the Active Directory attribute followed by a colon and the pattern. If you want to specify multiple patterns for multiple attributes, separate them with commas. The rule will be

			applied if the values of any of the specified attributes match the specified patterns for that attribute. This parameter is used to define a rule condition.
SenderAddressLocation	Optional	Microsoft.Exchange.MessagingPolicies.Rules.SenderAddressLocation	The *SenderAddressLocation* parameter specifies the fields to look in when evaluating sender addresses. Prior to Exchange Server 2013 Cumulative Update 1, transport rules only processed message headers when evaluating senders. With the addition of the *SenderAddressLocation* parameter, you can configure the rules to also examine the message envelope (the sender information sent with the MAIL FROM command in the SMTP transmission) when evaluating senders. Valid values are: • `Header` Only message headers are examined when evaluating senders.

			• `Envelope` Only the message envelope is examined when evaluating senders. • `HeaderOrEnvelope` Both message headers and the message envelope are examined when evaluating senders. The default value is `Header`. **Note:** By configuring this parameter, you can evaluate the message envelope for the following conditions and exceptions): • From • FromAddressContainsWords • FromAddressMatchesPatterns • FromMemberOf • SenderDomainIs
SenderDomainIs	Optional	Microsoft.Exchange.Data.Word[]	The *SenderDomainIs* parameter specifies the sender's domain. The rule is applied to messages received from senders whose email addresses are in the specified domain. This parameter is used to define a rule condition.
SenderInRecipientList	Optional	Microsoft.Exchange.Da	This parameter is available only in the cloud-based

		ta.Word[]	service. The *SenderInRecipientList* parameter specifies the condition when the sender is defined in a supervision list entry on a recipient's mailbox. Supervision list entries perform the following functions: • They specify individual exceptions for the user in the closed campus supervision policy. • They identify the user as a victim in the anti-bullying policy. To view the supervision list entries that are configured on a user's mailbox, run the **Get-SupervisionListEntry** command and specify the user's mailbox. This parameter is used to define a rule condition.
SenderIpRanges	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SenderIpRanges* parameter specifies the IP ranges to compare with the sender's IP address. The rule is applied if the IP address of the sender falls within one of the IP ranges

			specified in this parameter. This parameter is used to define a rule condition.
SenderManagementRelationship	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ManagementRelationship	The *SenderManagementRelationship* parameter specifies a relationship between the sender and the recipient. Valid values are: • `Manager` The rule is applied if the sender is the manager of the recipient. • `DirectReport` The rule is applied if the sender is a direct report of the recipient. This parameter is used to define a rule condition.
SentTo	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentTo* parameter specifies a recipient. The rule is applied to messages sent to the specified recipient. This parameter is used to define a rule condition.
SentToMemberOf	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *SentToMemberOf* parameter specifies a distribution group. The rule is applied to messages where any recipient is a member of

			the specified group. **Note:** If the distribution group is removed after creation of the rule, no action is taken. This parameter is used to define a rule condition.
SentToScope	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.ToUserScope	The *SentToScope* parameter specifies whether the message is sent to internal, external, or partner recipients. Valid values are: • `InOrganization` The recipients are internal to your organization. • `NotInOrganization` The recipients are outside your organization. • `ExternalPartner` The recipients are in a partner organization. • `ExternalNonPartner` The recipients are external to your organization, which isn't a partner organization. This parameter is used to define a rule condition.
SetAuditSeverity	Optional	System.String	The *SetAuditSeverity* parameter specifies the severity level assigned to an incident report that's generated and the

			corresponding entry logged in the message tracking logs. You can specify one of the following values: • `DoNotAudit` No audit entry is logged. • `Low` The audit entry is assigned low severity. • `Medium` The audit entry is assigned medium severity. • `High` The audit entry is assigned high severity. This parameter is used to define a rule action.
SetHeaderName	Optional	Microsoft.Exchange.Data.HeaderName	The *SetHeaderName* parameter specifies the SMTP header name to add or modify when the rule is applied. When the *SetHeaderName* parameter is used, you must also use the *SetHeaderValue* parameter to specify a value for the header. This parameter is used to define a rule action.
SetHeaderValue	Optional	Microsoft.Exchange.Data.HeaderValue	The *SetHeaderValue* parameter specifies a value for the header specified in the *SetHeaderName*

			parameter. This parameter is used to define a rule action.
SetSCL	Optional	Microsoft.Exchange.Data.SclValue	The *SetSCL* parameter modifies the SCL value of the message to the value specified. The SCL value can be a number from 0 through 9, or -1. The value -1 specifies that the message is from a trusted source. This parameter is used to define a rule action.
SmtpRejectMessageRejectStatusCode	Optional	Microsoft.Exchange.Data.RejectStatusCode	The *SmtpRejectMessageRejectStatusCode* parameter specifies an enhanced status code to provide when rejecting a message. This parameter is used to define a rule action.
SmtpRejectMessageRejectText	Optional	Microsoft.Exchange.Data.RejectText	The *SmtpRejectMessageRejectText* parameter specifies a text string to add to the rejection message. You must use this parameter with the *SmtpRejectMessageRejectStatusCode* parameter.

			This parameter is used to define a rule action.
StopRuleProcessing	Optional	System.Boolean	The *StopRuleProcessing* parameter specifies whether the processing of subsequent rules should be stopped for this message. This parameter is used to define a rule action.
SubjectContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *SubjectContainsWords* parameter specifies words to look for in the message subject. You can specify one or more words or phrases. When specifying more than one word or phrase, each word or phrase should be separated by a comma (,). When specifying a phrase that contains one or more spaces, you must enclose the phrase in quotation marks ("), for example: `word1,"Phrase with spaces",word2`. This parameter is used to define a rule condition.
SubjectMatchesPatter	Optional	Microsoft.Exchange.M	The

ns		essagingPolicies.Rules.Tasks.Pattern[]	*SubjectMatchesPatterns* parameter specifies text patterns to check for in the message subject. This parameter is used to define a rule condition.
SubjectOrBodyContainsWords	Optional	Microsoft.Exchange.Data.Word[]	The *SubjectOrBodyContainsWords* parameter specifies words to look for in the message subject or body. The rule is applied if any of the words or phrases specified is found in the message subject or body. You can specify one or more words or phrases. When specifying more than one word or phrase, each word or phrase should be separated by a comma (,). When specifying a phrase with one or more spaces, you must enclose the phrase in quotation marks ("), for example: `word1,"Phrase with spaces",word2`. This parameter is used to define a rule condition.
SubjectOrBodyMatche	Optional	Microsoft.Exchange.M	The

sPatterns		essagingPolicies.Rules.Tasks.Pattern[]	*SubjectOrBodyMatchesPatterns* parameter specifies text patterns to look for in the message subject or body. The rule is applied if the word specified is found in the message subject or body. This parameter is used to define a rule condition.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WithImportance	Optional	Microsoft.Exchange.Management.Tasks.Importance	The *WithImportance* parameter specifies message importance. The rule is applied to messages matching the specified importance. Valid values are: • `High` • `Low` • `Normal` This parameter is used to

			define a rule condition.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-TransportRuleAction

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-07*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-TransportRuleAction** cmdlet to retrieve a list of all available transport rule actions that can be used when creating a transport rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-TransportRuleAction [-Name <String>]
```

Examples

EXAMPLE 1

This example returns all available transport rule actions.

```
Get-TransportRuleAction
```

EXAMPLE 2

This example retrieves detailed information about a single transport rule action. The command is piped to the **Format-List** command to display all properties of the transport rule action.

```
Get-TransportRuleAction -Name DeleteMessage | Format-List
```

For more information about pipelining, see Pipelining. For more information about how to work with the output of a command, see Working with command output.

Detailed Description

The **Get-TransportRuleAction** cmdlet displays a list of available actions you can use in transport.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Optional	System.String	The *Name* parameter specifies the rule action to display. For a list of values that can be used with this parameter on the server role that you're administering, see Transport rule actions.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-TransportRuleCollection

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Export-TransportRuleCollection** cmdlet to export the transport rules in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Export-TransportRuleCollection [-Identity <RuleIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example exports transport rules. Rule data is first exported to the variable `$file`, and then written to the XML file `Rules.xml` in the `C:\MyDocs` folder.

```
$file = Export-TransportRuleCollection
Set-Content -Path "C:\MyDocs\Rules.xml" -Value
$file.FileData -Encoding Byte
```

Detailed Description

The **Export-TransportRuleCollection** cmdlet can be used to export the transport rule collection in your organization. The format of the exported transport rule collection has changed in Exchange Server 2013. The new format can only be imported into Exchange Server 2013: it can't be imported into older versions of Exchange.

Exporting the rules collection is a two-step process. You first export the rules collection to a variable, and then use the **Set-Content** cmdlet to write the data to an XML file. For more information, see Set-Content.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory

			Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	This parameter has been deprecated and is no longer used.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Import-TransportRuleCollection

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Import-TransportRuleCollection** cmdlet to import a transport rule collection. You can import a rule collection you previously exported as a backup, or import rules that you've exported from an older version of Exchange.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Import-TransportRuleCollection -FileData <Byte[]> [-Identity
<RuleIdParameter>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Force <SwitchParameter>] [-MigrationSource <None | Fope | Ehe>]
[-Organization <OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example imports a transport rule collection from the XML file ExportedRules.xml.

```
[Byte[]]$Data = Get-Content -Path "C:\TransportRules
\ExportedRules.xml" -Encoding Byte -ReadCount 0
Import-TransportRuleCollection -FileData $Data
```

Detailed Description

Caution:

Importing a transport rule collection from an XML file removes or overwrites all pre-existing transport rules that were defined in your organization. Make sure that you have a backup of your current transport rule collection before you import and overwrite your current transport rules.

Importing file data is a two-step process. First you must load the data to a variable using the **Get-Content** cmdlet, and then use that variable to transmit the data to the cmdlet.

For information about how to export a transport rule collection to an XML file, see Export-TransportRuleCollection.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description
FileData	Required	System.Byte[]	The *FileData* parameter specifies the variable name that contains the content of the XML file. The content is retrieved using the **Get-Content** cmdlet.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* parameter specifies that the command will override any errors or warnings encountered during the import operation.
Identity	Optional	Microsoft.Exchange.MessagingPolicies.Rules.Tasks.RuleIdParameter	The *Identity* parameter is no longer used and will be deprecated.
MigrationSource	Optional	Microsoft.Exchange.Data.Directory.Transport.MigrationSourceType	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-TransportRulePredicate

Exchange Management Shell > Exchange 2013 cmdlets > Policy and compliance cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-07*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-TransportRulePredicate** cmdlet to retrieve a list of all available rule conditions (predicates) that can be used when creating a transport rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-TransportRulePredicate [-Name <String>]
```

Examples

EXAMPLE 1

This example returns all available conditions for transport rules.

```
Get-TransportRulePredicate
```

EXAMPLE 2

This example retrieves the single condition `SubjectMatches`. The command is piped to the **Format-List** command to display detailed transport rule condition information.

```
Get-TransportRulePredicate -Name SubjectMatches | Format-List
```

For more information about pipelining, see Pipelining. For more information about how to work with the output of a command, see Working with command output.

EXAMPLE 3

This example returns the list of all transport rule conditions related to message attachments by filtering the output..

```
Get-TransportRulePredicate | Where {$_.Name -like '*Attachment*'}
```

For more information about how to work with the output of a command, see Working with command output

Detailed Description

The **Get-TransportRulePredicate** cmdlet displays a list of available rule conditions that you can use in transport rules.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Transport rules" entry in the Messaging policy and compliance permissions topic.

Parameters

Parameter	Required	Type	Description

Name	Optional	System.String	The *Name* parameter specifies the condition to display. For a list of values that can be used with this parameter on the server role that you're administering, see Transport rule conditions (predicates).

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Security cmdlets

Exchange Server 2013 > Exchange Management Shell > Exchange 2013 cmdlets >

Topic Last Modified: *2014-04-09*

Exchange certificate cmdlets

Enable-ExchangeCertificate

Export-ExchangeCertificate

Get-ExchangeCertificate

Import-ExchangeCertificate

New-ExchangeCertificate

Remove-ExchangeCertificate

Get-SmimeConfig

Set-SmimeConfig

Partner application cmdlets

Get-AuthConfig

Set-AuthConfig

Get-AuthServer

New-AuthServer

Remove-AuthServer

Set-AuthServer

Test-OAuthConnectivity

Get-PartnerApplication

New-PartnerApplication

Remove-PartnerApplication

Set-PartnerApplication

Get-AuthConfig

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-AuthConfig** cmdlet to get the authorization configuration for partner applications.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-AuthConfig [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves the authorization configuration for the Exchange organization and pipes the results to the **Format-List** command.

```
Get-AuthConfig | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AuthConfig

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-AuthConfig** cmdlet to modify the authorization configuration for your Exchange

organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-AuthConfig [-Realm <String>] [-ServiceName <String>] <COMMON
PARAMETERS>


Set-AuthConfig -CertificateThumbprint <String> [-Force <SwitchParameter>]
[-Server <ServerIdParameter>] [-SkipImmediateCertificateDeployment
<SwitchParameter>] <COMMON PARAMETERS>


Set-AuthConfig [-Force <SwitchParameter>] [-NewCertificateEffectiveDate
<DateTime>] [-NewCertificateThumbprint <String>] [-Server
<ServerIdParameter>] [-SkipImmediateCertificateDeployment
<SwitchParameter>] <COMMON PARAMETERS>


Set-AuthConfig [-ClearPreviousCertificate <SwitchParameter>] [-Force
<SwitchParameter>] [-PublishCertificate <SwitchParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example specifies a new certificate and a date when the certificate will become effective.

```
Set-AuthConfig -NewCertificateThumbprint
DB821B4FCA2A5DA9593B9DE00C86BC5EA35D0FC0 -
NewCertificateEffectiveDate 4/17/2013
```

EXAMPLE 2

This example immediately rolls over the certificate configured as the next certificate and makes it the current certificate. You must have installed a certificate marked as the next certificate.

```
Set-AuthConfig -PublishCertificate
```

Detailed Description

The **Set-AuthConfig** parameter defines Microsoft Exchange as a partner application for server-to-server authentication with other partner applications such as Microsoft SharePoint 2013 and Microsoft Lync 2013, including the certificate used for signing tokens. It's generally not required for this configuration to be modified except in some cases where you must use a different certificate instead of the self-signed certificate created by Exchange Setup or to use a new certificate after the old one has expired.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
CertificateThumbprint	Required	System.String	The *CertificateThumbprint* parameter specifies the thumbprint of the certificate to be used by Exchange for server-to-server authentication.
ClearPreviousCertificate	Optional	System.Management.Automation.SwitchParameter	The *ClearPreviousCertificate* switch clears the certificate saved as the previous certificate in the authorization configuration.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
NewCertificateEffectiveDate	Optional	System.DateTime	The *NewCertificateEffectiveDate* parameter specifies a date when the certificate configured as the next certificate should be used.
NewCertificateThumbprint	Optional	System.String	The *NewCertificateThumbprint* parameter specifies the thumbprint of the new

			certificate to be used as the next certificate in the authorization configuration.
PublishCertificate	Optional	System.Management.Automation.SwitchParameter	The *PublishCertificate* switch specifies that the specified certificate be immediately rolled over as the current certificate. The certificate is immediately deployed to all Client Access servers.
Realm	Optional	System.String	The *Realm* parameter specifies a security realm for partner applications. If a service or user presents a token from a domain that's not an accepted domain in the Exchange organization, the token must contain the specified realm to gain access to resources.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter isn't available in this release.
ServiceName	Optional	System.String	The *ServiceName* parameter identifies Microsoft Exchange to other partner applications such as SharePoint 2013.

			Caution: Exchange Setup configures the *ServiceName* parameter with a specific constant value. You shouldn't change this parameter. Changing the *ServiceName* parameter can result in server-to-server authentication with partner applications not functioning.
SkipImmediateCertificateDeployment	Optional	System.Management.Automation.SwitchParameter	The *SkipImmediateCertificateDeployment* switch specifies that the certificate shouldn't be used immediately. We recommend that you don't use this parameter in a production environment.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AuthServer

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-AuthServer** cmdlet to retrieve settings of authorization servers configured in the Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-AuthServer [-Identity <AuthServerIdParameter>] [-DomainController
<Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves all settings for all authorization servers in the Exchange organization.

```
Get-AuthServer | Format-List *
```

Detailed Description

An authorization server is a server or service that issues tokens trusted by Microsoft Exchange Server 2013 for access by partner applications.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner application - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.AuthServerIdParameter	The *Identity* parameter specifies the identity of an authorization server.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-AuthServer

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-AuthServer** cmdlet to create an authorization server object in Microsoft Exchange Server 2013 and specify its *AuthMetadataUrl*. Exchange 2013 honors tokens issued by the authorization server for access by a partner application.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-AuthServer -AuthMetadataUrl <String> [-TrustAnySSLCertificate
<SwitchParameter>] <COMMON PARAMETERS>


New-AuthServer -IssuerIdentifier <String> -TokenIssuingEndpoint <String> -
Type <Unknown | MicrosoftACS | Facebook | LinkedIn | ADFS | AzureAD> [-
ApplicationIdentifier <String>] [-AppSecret <String>] [-
AuthorizationEndpoint <String>] <COMMON PARAMETERS>


New-AuthServer -AuthMetadataUrl <String> -Type <Unknown | MicrosoftACS |
Facebook | LinkedIn | ADFS | AzureAD> [-TrustAnySSLCertificate
<SwitchParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Enabled <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This command creates an authorization server.

```
New-AuthServer HRAppAuth -AuthMetadataUrl http://
hrappauth.contoso.com/metadata/json/1
```

Detailed Description

In Exchange 2013, partner applications authorized by Exchange can access their resources after they're authenticated using server-to-server authentication. A partner application can authenticate by using self-issued tokens trusted by Exchange or by using an authorization server trusted by Exchange.

The **New-AuthServer** cmdlet creates a trusted authorization server object in Exchange 2013, which allows it to trust tokens issued by the authorization server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner

applications - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
AuthMetadataUrl	Required	System.String	The *AuthMetadataUrl* parameter specifies the URL for the Microsoft Office 365 authorization server for your cloud-based organization. For details, see the Office 365 documentation.
IssuerIdentifier	Required	System.String	This parameter is reserved for internal Microsoft use.
Name	Required	System.String	The *Name* parameter specifies a name for the authorization server.
TokenIssuingEndpoint	Required	System.String	This parameter is reserved for internal Microsoft use.
Type	Required	Microsoft.Exchange.Data.Directory.SystemConfiguration.AuthServerType	This parameter is reserved for internal Microsoft use.
ApplicationIdentifier	Optional	System.String	This parameter is reserved for internal Microsoft use.
AppSecret	Optional	System.String	This parameter is

			reserved for internal Microsoft use.
AuthorizationEndpoint	Optional	System.String	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the authorization server is enabled. Set the parameter to `$false` to prevent authorization tokens issued by this

			authorization server from being accepted.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-AuthServer

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-AuthServer** cmdlet to remove an authorization server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-AuthServer -Identity <AuthServerIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the authorization server AMC.

```
Remove-AuthServer AMC
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AuthServerIdParameter	The *Identity* parameter specifies the identity of the authorization server.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the

			confirmation prompt, use the syntax -`confirm:$false`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AuthServer

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-AuthServer** cmdlet to configure an authorization server that partner applications can use to obtain tokens recognized by Microsoft Exchange.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-AuthServer [-AuthMetadataUrl <String>] [-TrustAnySSLCertificate
<SwitchParameter>] <COMMON PARAMETERS>


Set-AuthServer [-AuthMetadataUrl <String>] [-
IsDefaultAuthorizationEndpoint <$true | $false>] [-TrustAnySSLCertificate
<SwitchParameter>] <COMMON PARAMETERS>


Set-AuthServer [-RefreshAuthMetadata <SwitchParameter>] <COMMON
PARAMETERS>


Set-AuthServer [-ApplicationIdentifier <String>] [-AppSecret <String>] [-
IssuerIdentifier <String>] [-TokenIssuingEndpoint <String>] <COMMON
PARAMETERS>


COMMON PARAMETERS: -Identity <AuthServerIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-Name <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This command disables the authorization server ACS.

```
Set-AuthServer ACS -Enabled $false
```

Detailed Description

In Exchange 2013, partner applications authorized by Exchange can access their resources after they're authenticated using server-to-server authentication. A partner application can authenticate

by using self-issued tokens trusted by Exchange or by using an authorization server trusted by Exchange. You can use the **New-AuthServer** cmdlet to create a trusted authorization server object in Exchange 2013, which allows it to trust tokens issued by the authorization server.

Use the **Set-AuthServer** cmdlet to enable or disable the authorization server, change the *AuthMetadataUrl* parameter, or refresh authorization metadata.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AuthServerIdParameter	The *Identity* parameter specifies the identity of authorization server.
ApplicationIdentifier	Optional	System.String	This parameter is reserved for internal Microsoft use.
AppSecret	Optional	System.String	This parameter is reserved for internal Microsoft use.
AuthMetadataUrl	Optional	System.String	The *AuthMetadataUrl* parameter specifies the URL of the authorization server. This can be the AuthMetadataUrl of your Microsoft Exchange Online organization.
Confirm	Optional	System.Management.Automation.SwitchPara	The *Confirm* switch causes the command to

		meter	pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the authorization server is enabled. Only enabled authorization servers can issue and accept tokens. Disabling the authorization server prevents any partner applications configured to use the authorization server from getting a token.
IsDefaultAuthorization	Optional	System.Boolean	The *IsDefaultAuthorizationE*

Endpoint			*ndpoint* parameter specifies whether this server is the default authorization endpoint. This server's authorization URL is advertised to calling partner applications, and applications need to get their OAuth access tokens from this authorization server. Valid input for this parameter is `$true` or `$false`. The default value is `$false`.
IssuerIdentifier	Optional	System.String	This parameter is reserved for internal Microsoft use.
Name	Optional	System.String	The *Name* parameter specifies a name for the authorization server.
RefreshAuthMetadata	Optional	System.Management.Automation.SwitchParameter	The *RefreshAuthMetadata* switch specifies whether Exchange should refresh the auth metadata from the specified URL.
TokenIssuingEndpoint	Optional	System.String	This parameter is reserved for internal

			Microsoft use.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	The *TrustAnySSLCertificate* switch specifies whether Exchange should accept certificates from an untrusted certification authority. We don't recommend using this switch in a production environment.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-ExchangeCertificate

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-ExchangeCertificate** cmdlet to enable an existing certificate in the local certificate store for Exchange services such as Internet Information Services (IIS), SMTP, POP, IMAP, and Unified Messaging (UM).

Important:
There are many factors to consider when you configure certificates for Transport Layer Security (TLS) and Secure Sockets Layer (SSL) services. You must understand how these factors may affect your overall configuration. Don't use the **Enable-ExchangeCertificate** cmdlet to enable a wildcard certificate for POP and IMAP services. To enable a wildcard certificate, you must use the **Set-ImapSettings** or **Set-PopSettings** cmdlets with the fully qualified domain name (FQDN) of the service. Don't use the **Enable-ExchangeCertificate** cmdlet to enable a certificate for federation. Certificates used for federation trusts are managed by using the **New-FederationTrust** and **Set-FederationTrust** cmdlets.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-ExchangeCertificate -Thumbprint <String> [-Server
<ServerIdParameter>] <COMMON PARAMETERS>


Enable-ExchangeCertificate [-Identity <ExchangeCertificateIdParameter>]
<COMMON PARAMETERS>


COMMON PARAMETERS: -Services <None | IMAP | POP | UM | IIS | SMTP |
Federation | UMCallRouter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-DoNotRequireSsl <SwitchParameter>] [-Force
<SwitchParameter>] [-NetworkServiceAllowed <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables a certificate for POP, IMAP, SMTP, and IIS services.

```
Enable-ExchangeCertificate -Thumbprint
5113ae0233a72fccb75b1d0198628675333d010e -Services
POP,IMAP,SMTP,IIS
```

Detailed Description

The **Enable-ExchangeCertificate** cmdlet enables certificates by updating the metadata stored with the certificate. To enable an existing certificate to work with additional Exchange services, use the **Enable-ExchangeCertificate** cmdlet and specify the additional services.

Important:
The **Enable-ExchangeCertificate** cmdlet is additive. When you specify a subset of services for which a certificate is enabled, the services that aren't specified aren't removed from the **Services** property. If you don't want to use an existing enabled certificate for Exchange services, you must enable another certificate, and then remove the certificate you don't want to use.

Different services have different certificate requirements. For example, some services may only require a server name in the **Subject Name** or **Subject Alternative Name** fields of a certificate, whereas other services may require an FQDN. Make sure that the certificate name can support the uses required by the services you enable it for.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Certificate management" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Services	Required	Microsoft.Exchange.Management.SystemConfigurationTasks.AllowedServices	The *Services* parameter specifies the services that use the certificate. Valid entries include one or more of the following: • `IIS` • `IMAP` • `POP` • `SMTP` • `UM` • `UMCallRouter` • `Federation` • `None` To enable a certificate for multiple services, separate each value with a comma, for example:

			`-Services IMAP,POP,IIS` You can't use the **Enable-ExchangeCertificate** cmdlet to enable a certificate for federation. Creating or modifying a federation trust enables or modifies how certificates are used for federation.
Thumbprint	Required	System.String	The *Thumbprint* parameter specifies the certificate that you're enabling. Each certificate contains a thumbprint, which is the digest of the certificate data. To view the thumbprint of a certificate, use the Get-ExchangeCertificate cmdlet.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
DoNotRequireSsl	Optional	System.Management.Automation.SwitchPara	The *DoNotRequireSsl* switch specifies whether to leave IIS settings unchanged when IIS is one of the enabled services. If IIS is one of the

		meter	enabled services, the cmdlet changes the default website settings to require SSL. Set the *DoNotRequireSsl* switch to `$true` to override this behavior and leave IIS settings unchanged.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to override the confirmation prompt and set the new certificate as the default certificate for TLS for internal SMTP communication. By default, when you enable a certificate for SMTP, the command prompts for confirmation.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ExchangeCertificateIdParameter	The *Identity* parameter specifies the certificate ID.
NetworkServiceAllowed	Optional	System.Management.Automation.SwitchParameter	The *NetworkServiceAllowed* switch specifies that the Network Service be allowed permissions to access the certificate specified, without enabling the certificate for SMTP.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server name on which you want to enable the certificate.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-ExchangeCertificate

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Export-ExchangeCertificate** cmdlet to export an existing certificate from the certificate store on the local computer. You can export a certificate with its private key or a certificate request file.

```
Export-ExchangeCertificate -Thumbprint <String> [-Server
<ServerIdParameter>] <COMMON PARAMETERS>


Export-ExchangeCertificate [-Identity <ExchangeCertificateIdParameter>]
<COMMON PARAMETERS>


COMMON PARAMETERS: [-BinaryEncoded <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-FileName <String>] [-
Password <SecureString>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example exports a certificate specified by its thumbprint, along with the private key, to a file named htcert.pfx in the certificates directory on a Hub Transport server. The exported certificate is DER-encoded. A password is required when exporting a certificate with its private key.

The following command uses the **Export-ExchangeCertificate** cmdlet to export certificate data to the variable *$file*.

```
$file = Export-ExchangeCertificate -Thumbprint
5113ae0233a72fccb75b1d0198628675333d010e -
BinaryEncoded:$true -Password (Get-Credential).password
```

The following command uses the **Set-Content** cmdlet to write data stored in the variable *$file* to the file htcert.pfx.

```
Set-Content -Path "c:\certificates\htcert.pfx" -Value
$file.FileData -Encoding Byte
```

Detailed Description

The **Export-ExchangeCertificate** cmdlet creates either of the following files:

- **PKCS #10 file** If the thumbprint specified in the command points to a certificate request, the **Export-ExchangeCertificate** cmdlet creates a PKCS #10 file. A thumbprint is the digest of the certificate data. PKCS #10 is the Certification Request Syntax standard specified by RFC 2314. For more information, see PKCS #10: Certification Request Syntax.
- **PKCS #12 file** If the thumbprint specified in the command points to an actual certificate, the **Export-ExchangeCertificate** cmdlet creates a PKCS #12 file. PKCS #12 is the Personal Information Exchange Syntax standard specified by RSA Laboratories. For more information, see PKCS #12: Personal Information Exchange Syntax Standard.

Important:
When you use the **Export-ExchangeCertificate** cmdlet, you must export certificate data to a variable, as shown in "Examples" later in this topic, and then use the **Set-Content** cmdlet to write the data to a file. For more information about the Set-Content cmdlet, see Set-Content.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Certificate management" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Thumbprint	Required	System.String	The *Thumbprint* parameter specifies the thumbprint of the certificate that you're exporting. Each certificate contains a

			thumbprint, which is the digest of the certificate data. It can be retrieved by using the Get-ExchangeCertificate cmdlet.
BinaryEncoded	Optional	System.Management.Automation.SwitchParameter	The *BinaryEncoded* parameter specifies how the exported file is encoded. By default, this command creates a Base64-encoded file. To create a DER-encoded file, set this parameter to `$true`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
FileName	Optional	System.String	The *FileName* parameter specifies the name of the file that will contain the exported certificate.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ExchangeCertificateIdParameter	The *Identity* parameter specifies the certificate ID.
Password	Optional	System.Security.SecureString	The *Password* parameter specifies the password for the private key that's exported with this command. Use the **Get-Credential** cmdlet to store the password variable.

			The **Get-Credential** cmdlet will prompt you for a user name and password, but only the password field is used to export or import the certificate. Therefore, you don't have to use a real domain name or user name in the **Name** field. For implementation details, see "Examples" later in this topic.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server name from which you want to export the certificate.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ExchangeCertificate

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ExchangeCertificate** cmdlet to view certificates in the local certificate store.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ExchangeCertificate [-Server <ServerIdParameter>] [-Thumbprint
<String>] <COMMON PARAMETERS>


Get-ExchangeCertificate [-Identity <ExchangeCertificateIdParameter>]
<COMMON PARAMETERS>


Get-ExchangeCertificate [-Instance <X509Certificate2>] [-Server
<ServerIdParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-DomainController <Fqdn>] [-DomainName
<MultiValuedProperty>]
```

Examples

EXAMPLE 1

This example returns all certificates stored on the Client Access server named ClientAccess01.

```
Get-ExchangeCertificate -Server ClientAccess01
```

EXAMPLE 2

This example returns the properties of a specified certificate in a formatted list.

Note:
The *Thumbprint* parameter is a positional parameter so you can provide only the thumbprint value without the *Thumbprint* parameter name.

```
Get-ExchangeCertificate
0271A7F1CA9AD8A27152CCAE044F968F068B14B8 | Format-List *
```

EXAMPLE 3

This example shows which certificate Exchange will select for the domain name mail.contoso.com. A Send or Receive connector selects the certificate to use based on the fully qualified domain name (FQDN) of the connector. If you have multiple certificates with the same FQDN, you can see which certificate Exchange will select by using the *DomainName* parameter to specify the FQDN. The first certificate returned is the certificate Exchange will select.

```
Get-ExchangeCertificate -DomainName mail.contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Certificate management" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge

			Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
DomainName	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *DomainName* parameter specifies whether to return all certificates that contain the specified domain name in the **Subject Name** or the **Subject Alternative Name** fields.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ExchangeCertificateIdParameter	The *Identity* parameter specifies the certificate ID.
Instance	Optional	System.Security.Cryptography.X509Certificates.X509Certificate2	The *Instance* parameter is no longer used and will be deprecated.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server from which you want to get the certificate. You can use any value that uniquely identifies the Exchange server.

			If you run **Get-ExchangeCertificate** on a Client Access server, and you don't use the *Server* parameter to specify the local server, the command returns the results from a Mailbox server.
Thumbprint	Optional	System.String	The *Thumbprint* parameter specifies a certificate thumbprint. Each certificate contains a thumbprint, which is the digest of the certificate data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Import-ExchangeCertificate

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Import-ExchangeCertificate** cmdlet to import a certificate or chain of certificates.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Import-ExchangeCertificate -Instance <String[]> <COMMON PARAMETERS>

Import-ExchangeCertificate -FileData <Byte[]> <COMMON PARAMETERS>

Import-ExchangeCertificate -FileName <String> <COMMON PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-FriendlyName <String>] [-Password <SecureString>] [-
PrivateKeyExportable <$true | $false>] [-Server <ServerIdParameter>] [-
WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example imports an existing certificate and private key from the PKCS #12 file ExportedCert.pfx.

```
Import-ExchangeCertificate -FileData ([Byte[]]$(Get-Content
-Path c:\certificates\ExportedCert.pfx -Encoding byte -
ReadCount 0)) -Password:(Get-Credential).password
```

EXAMPLE 2

This example imports a chain of certificates from the PKCS #7 file IssuedCert.p7b.

```
Import-ExchangeCertificate -FileData ([Byte[]]$(Get-Content
-Path c:\certificates\IssuedCert.p7b -Encoding byte -
ReadCount 0))
```

Detailed Description

You can use the **Import-ExchangeCertificate** cmdlet for the following purposes:

- To import a certificate or chain of certificates from a PKCS #7 file that has been issued by a certification authority (CA). PKCS #7 is the Cryptographic Message Syntax Standard, a syntax used for digitally signing or encrypting data using public key cryptography, including certificates.
- To import an existing certificate and private key from a PKCS #12 (.pfx or .p12) file to the certificate store on the local computer. PKCS #12 is the Personal Information Exchange Syntax Standard, a file format used to store certificates with corresponding private keys protected with a password. The standard is specified by RSA Laboratories. For more information, see the PKCS #12: Personal Information Exchange Syntax Standard website.

Important:

There are many factors to consider when you configure certificates for Transport Layer Security (TLS) and Secure Sockets Layer (SSL) services. You must understand how these factors may affect your overall configuration.

Note:

In Microsoft Exchange Server 2013, to import data from a file, you must use the **Get-Content** cmdlet to retrieve file data and use the *FileData* parameter to specify the retrieved data. This can be done in a two-step process, or in a single step. Examples shown in this cmdlet use the single-step approach.

The certificate may be published in Active Directory for the purposes of direct trust by using mutual TLS if the following conditions are true:

- The certificate is marked as an SMTP TLS certificate.
- The Subject Name on the certificate matches the fully qualified domain name (FQDN) of the local computer.

The certificate may be published in Active Directory by Edge Subscription if the following conditions are true:

- You import the certificate to an Edge Transport server.
- The certificate has an FQDN that matches the server FQDN.

The **Import-ExchangeCertificate** cmdlet imports either a certificate that's issued from an outstanding request or a PKCS #12 file.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Certificate management" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
FileData	Required	System.Byte[]	The *FileData* parameter specifies the content retrieved from the certificate file using the **Get-Content** cmdlet. For implementation details, see the Examples section.
FileName	Required	System.String	The *FileName*

			parameter specifies the name of the file that contains the certificate you want to import.
Instance	Required	System.String[]	The *Instance* parameter specifies whether to pass a whole object to the command to be processed. This parameter is mainly used in scripts where a whole object must be passed to the command.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change

			to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
FriendlyName	Optional	System.String	The *FriendlyName* parameter specifies a friendly name for the resulting certificate. The friendly name must be less than 64 characters. The default friendly name is Microsoft Exchange.
Password	Optional	System.Security.SecureString	The *Password* parameter specifies the password for the private key that's imported with this command. Use the **Get-Credential** cmdlet to store the password variable. The **Get-Credential** cmdlet prompts you for

			a user name and password, but only the password field is used to import the certificate. You don't have to use a real domain name or user name in the **Name** field. For implementation details, see the Examples section.
PrivateKeyExportable	Optional	System.Boolean	The *PrivateKeyExportable* parameter specifies whether the private key of the certificate can be exported.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server name to which you want to import the certificate.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without

			having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ExchangeCertificate

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-ExchangeCertificate** cmdlet to create a self-signed certificate, renew an existing self-signed certificate, or generate a new certificate request for obtaining a certificate from a certification authority (CA).

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-ExchangeCertificate [-BinaryEncoded <SwitchParameter>] [-
GenerateRequest <SwitchParameter>] [-RequestFile <String>] <COMMON
PARAMETERS>


New-ExchangeCertificate [-Services <None | IMAP | POP | UM | IIS | SMTP |
Federation | UMCallRouter>] <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-DomainName <MultiValuedProperty>] [-Force <SwitchParameter>] [-
FriendlyName <String>] [-IncludeAcceptedDomains <SwitchParameter>] [-
IncludeAutoDiscover <SwitchParameter>] [-IncludeServerFQDN
<SwitchParameter>] [-IncludeServerNetBIOSName <SwitchParameter>] [-
Instance <X509Certificate2>] [-KeySize <Int32>] [-PrivateKeyExportable
<$true | $false>] [-Server <ServerIdParameter>] [-SubjectKeyIdentifier
```

```
<String>] [-SubjectName <X500DistinguishedName>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example runs the **New-ExchangeCertificate** cmdlet without parameters and generates a self-signed certificate. The certificate has the fully qualified domain name (FQDN) of the local computer as the subject name. The Network Services local security group is also provided read access to the private key associated with the certificate. In addition, the certificate is published to Active Directory so that Exchange direct trust can validate the authenticity of the server for mutual TLS.

```
New-ExchangeCertificate
```

EXAMPLE 2

This example outputs the certificate request in Base64 format to the command-line console. You must send the certificate request to a CA within the organization, a trusted CA outside the organization, or a commercial CA. You can do this by pasting the certificate request output in an email message or in the appropriate field on the certificate request web page of the CA. You can also save the certificate request to a file using a text editor such as Notepad.

The certificate that results has the following attributes associated with it:

- Subject name: c=<ES>,o=<Woodgrove Bank>,cn=mail1.woodgrovebank.com
- Subject alternate names: woodgrovebank.com and example.com
- An exportable private key

```
New-ExchangeCertificate -GenerateRequest -SubjectName
"c=US, o=Woodgrove Bank, cn=mail1.woodgrovebank.com" -
DomainName woodgrovebank.com, example.com -
PrivateKeyExportable $true
```

EXAMPLE 3

This example is a variation of the certificate request generated in EXAMPLE 2. Instead of manually pasting the certificate request output produced by the cmdlet, the **Set-Content** cmdlet is used to write the request to a file.

The certificate that results has the following attributes associated with it:

- Subject name: c=<ES>,o=<Woodgrove Bank>,cn=mail1.woodgrovebank.com
- Subject alternate names: woodgrovebank.com and example.com
- An exportable private key

In the first step, the **New-ExchangeCertificate** cmdlet is used to generated the certificate request and save the output in a variable named *$Data*.

```
$Data = New-ExchangeCertificate -GenerateRequest -
SubjectName "c=US, o=Woodgrove Bank,
cn=mail1.woodgrovebank.com" -DomainName woodgrovebank.com,
example.com -PrivateKeyExportable $true
```

In the second step, the **Set-Content** cmdlet is used to write data from the variable to the certificate request file MyCertRequest.req in the Docs folder.

```
Set-Content -path "C:\Docs\MyCertRequest.req" -Value $Data
```

EXAMPLE 4

This example creates a DER-encoded certificate request file. The *BinaryEncoded* parameter is used to generate a DER-encoded certificate request. The **Set-Content** cmdlet is used with the *Encoding* parameter to write the request to a file.

The certificate that results will have the following attributes associated with it:

- Subject name: c=ES,o=Woodgrove Bank,cn=mail1.woodgrovebank.com
- Subject alternate names: woodgrovebank.com and example.com
- An exportable private key

In the first step, the **New-ExchangeCertificate** cmdlet is used to generate the certificate request in DER-encoded format and save the output in a variable named *$Data*.

```
$Data = New-ExchangeCertificate -GenerateRequest -
SubjectName "c=ES, o=Woodgrove Bank,
cn=mail1.woodgrovebank.com" -DomainName woodgrovebank.com,
example.com -BinaryEncoded -PrivateKeyExportable $true
```

In the second step, the **Set-Content** cmdlet is used to write data from the variable to the certificate request file MyCertRequest.req in the Docs folder.

```
Set-Content -path "C:\Docs\MyCertRequest.req" -Value
$Data.FileData -Encoding Byte
```

EXAMPLE 5

This example shows how to renew a self-signed certificate with a specific thumbprint value. You can obtain the thumbprint value in one of two ways.

- Select the certificate in the Exchange Administration Center, and then select **Edit** to view properties of the certificate. The thumbprint value is shown in the **Exchange Certificate** window.
- Run the **Get-ExchangeCertificate** cmdlet to return a list of all certificates installed on the server with their thumbprint values.

```
Get-ExchangeCertificate -Thumbprint
```

```
c4248cd7065c87cb942d60f7293feb7d533a4afc | New-
ExchangeCertificate
```

Detailed Description

Microsoft Exchange Server 2013 uses certificates for SSL and TLS encryption. The **New-ExchangeCertificate** cmdlet uses many parameters of type *SwitchParameter*. For more information about how to use this parameter type, see "Switch Parameters" in Parameters.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Certificate management" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
BinaryEncoded	Optional	System.Management.Automation.SwitchParameter	The *BinaryEncoded* switch specifies how the certificate request is encoded. By default, this cmdlet creates a Base64-encoded request. Use this switch to create a DER-encoded request. **Note:** The *BinaryEncoded* switch is available only if you use the *GenerateRequest* switch.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You

			don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
DomainName	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *DomainName* parameter specifies one or more FQDNs or server names to be populated in the **Subject Alternative Name** field of the resulting certificate request or self-signed certificate. Domain names are restricted to the

			characters a through z, 0 through 9, and the hyphen (-). Each domain name can't be longer than 255 characters. To enter multiple domain or server names, you must enter the names separated by commas. **Note:** If this parameter isn't specified, and you don't use the *IncludeAcceptedDomains*, *IncludeAutoDiscover*, *IncludeServerFQDN*, and *IncludeServerNetBIOSName* switches, the server's **hostname** and **FQDN** are added by default.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to override the confirmation prompt and set the new self-signed certificate as the default certificate for TLS for internal SMTP communication. By default, this cmdlet requires a confirmation before setting the new certificate as the default certificate for TLS encryption of internal SMTP communication.

FriendlyName	Optional	System.String	The *FriendlyName* parameter specifies a friendly name for the certificate. The friendly name must be less than 64 characters. The default friendly name is `Microsoft Exchange`.
GenerateRequest	Optional	System.Management.Automation.SwitchParameter	The *GenerateRequest* switch specifies whether to generate a certificate request for a public key infrastructure (PKI) certificate (PKCS #10) in the local request store. By default, this cmdlet creates a self-signed certificate in the local computer certificate store. The *GenerateRequest* switch overrides this behavior.
IncludeAcceptedDomains	Optional	System.Management.Automation.SwitchParameter	The *IncludeAcceptedDomains* switch specifies whether all accepted domains in the organization are included in the **Subject Alternative Name** field of the certificate request or self-signed certificate.

			You can also specify one or more domain names using the *DomainName* parameter in addition to the accepted domains. The resulting certificate or request contains the specified domains and all accepted domains. **Note:** When you use the *IncludeAcceptedDomains* switch, any accepted domains you specify in the *DomainName* parameter aren't duplicated.
IncludeAutoDiscover	Optional	System.Management.Automation.SwitchParameter	The *IncludeAutoDiscover* switch specifies whether to add a **Subject Alternative Name** with the prefix `autodiscover` for each accepted domain in the Exchange organization. For example, if the organization has the accepted domains `woodgrovebank.com` and `woodgrovebank.co.uk`, using this switch results in the addition of the following **Subject Alternative Names**: • `autodiscover.woodgrovebank.com` • `autodiscover.woodgrovebank.co.uk`

			The switch can only be used on Client Access servers. The `autodiscover` prefix isn't added if the domain name already contains the prefix.
IncludeServerFQDN	Optional	System.Management.Automation.SwitchParameter	The *IncludeServerFQDN* switch specifies whether to include the FQDN of the server in the **Subject Alternative Name** field of the new certificate request or self-signed certificate. **Note:** When you use the *IncludeServerFQDN* switch, any FQDNs you specify in the *DomainName* parameter aren't duplicated.
IncludeServerNetBIOSName	Optional	System.Management.Automation.SwitchParameter	The *IncludeServerNetBIOSName* switch specifies whether to include the NetBIOS name of the server in the **Subject Alternative Name** field of the new certificate request or self-signed certificate. **Note:** When you use the

			IncludeServerNetBIOSName switch, any NetBIOS names you specify in the *DomainName* parameter aren't duplicated.
Instance	Optional	System.Security.Cryptography.X509Certificates.X509Certificate2	The *Instance* parameter is no longer used and will be deprecated.
KeySize	Optional	System.Int32	The *KeySize* parameter specifies the size (in bits) of the RSA public key associated with the certificate that you're creating. Acceptable values are 4096, 2048, and 1024. The default value is 2048.
PrivateKeyExportable	Optional	System.Boolean	The *PrivateKeyExportable* parameter specifies whether the new certificate has an exportable private key. By default, all certificate requests and certificates created by this cmdlet don't allow the private key to be exported. **Important:** If you can't export the private key, the certificate can't be exported or imported. To allow exporting the

			private key when exporting the certificate, set this parameter to `$true`.
RequestFile	Optional	System.String	The *RequestFile* parameter specifies the name of the file that contains the certificate request. This parameter is used with the *BinaryEncoded* and *GenerateRequest* parameters.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server name for which you want to create the new certificate. If not specified, the certificate or certificate request is generated for the Exchange server on which the command is executed.
Services	Optional	Microsoft.Exchange.Management.SystemConfigurationTasks.AllowedServices	The *Services* parameter specifies the services that will use the resulting certificate. **Important:** You can specify services using the **New-ExchangeCertificate** cmdlet only if you're generating a self-signed certificate. If you're

			creating a certificate request for a CA using the *GenerateRequest* switch, you must install the certificate after it's issued by the CA, and then specify services using the **Enable-ExchangeCertificate** cmdlet. Valid values include a combination of the following: • `Federation` • `IIS` • `IMAP` • `None` • `POP` • `SMTP` • `UM` • `UMCallRouter` The default values are `IMAP`,`POP`, and `SMTP`.
SubjectKeyIdentifier	Optional	System.String	The *SubjectKeyIdentifier* parameter specifies the subject key identifier extension for the certificate, which isn't required for normal operation.
SubjectName	Optional	System.Security.Cryptography.X509Certificates.X500DistinguishedName	The *SubjectName* parameter specifies the subject name of the resulting certificate. A subject name is an X.500 distinguished name that consists of one or more relative distinguished

			names (also known as RDNs). The subject name of a certificate is the field used by Domain Name System (DNS)-aware services. It binds a certificate to a particular server or domain name. If the *SubjectName* parameter isn't specified, the host name of the server where the cmdlet is run is used as the common name (CN) in the resulting certificate. For example, for the server EXMBX01, the *SubjectName* parameter value `CN=EXMBX01` is used.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ExchangeCertificate

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-ExchangeCertificate** cmdlet to remove an existing certificate from the local certificate store.

Important:
There are many factors to consider when you configure certificates for Transport Layer Security (TLS) and Secure Sockets Layer (SSL) services. You must understand how these factors may affect your overall configuration.

```
Remove-ExchangeCertificate -Thumbprint <String> [-Server
<ServerIdParameter>] <COMMON PARAMETERS>


Remove-ExchangeCertificate [-Identity <ExchangeCertificateIdParameter>]
<COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes a certificate with the specified thumbprint.

```
Remove-ExchangeCertificate -Thumbprint
```

5113ae0233a72fccb75b1d0198628675333d010e

Detailed Description

You can't remove the certificate that's being used. If you want to replace the default certificate for the server with another certificate that has the same fully qualified domain name (FQDN), you must create the new certificate first, and then remove the old certificate.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Certificate management" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Thumbprint	Required	System.String	The *Thumbprint* parameter specifies the thumbprint of the certificate that you're removing. Each certificate contains a thumbprint, which is the digest of the certificate data.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ExchangeCertificateIdParameter	The *Identity* parameter specifies the certificate ID.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server name from which you want to remove the certificate.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on

			the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-OAuthConnectivity

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-07*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Test-OAuthConnectivity** cmdlet to test OAuth authentication to partner applications for a user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-OAuthConnectivity -Service <EWS | AutoD | Generic> -TargetUri <Uri>
[-AppOnly <SwitchParameter>] [-Confirm [<SwitchParameter>]] [-Mailbox
<MailboxIdParameter>] [-OrganizationDomain <String>] [-ReloadConfig
<SwitchParameter>] [-UseCachedToken <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example tests OAuth connectivity with Exchange for Gurinder Singh.

```
Test-OAuthConnectivity -Service EWS -TargetUri https://
cas.contoso.com/ews/ -Mailbox "Gurinder Singh"
```

Detailed Description

Important:
Exchange 2013 creates the SharePoint 2013 and Lync 2013 partner applications in on-premises Exchange 2013 deployments. For **Test-OAuthConnectivity** cmdlet to succeed for other partner applications, you must have created the partner application first using the `Configure-EnterpriseApplication.ps1` script.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Service	Required	Microsoft.Exchange.Monitoring.ModServiceType	The *Service* parameter specifies the partner application. Valid values for this parameter are: • `EWS` • `AutoD` • `Generic`
TargetUri	Required	System.Uri	The *TargetUri* parameter specifies the URL for the service you want to test OAuth connectivity with.
AppOnly	Optional	System.Management.Automation.SwitchParameter	The *AppOnly* switch specifies the cmdlet will authenticate to the

			specified service as Exchange without any user context. You don't need to specify a value with this switch.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the mailbox for which you want to test OAuth connectivity to the specified partner application.
OrganizationDomain	Optional	System.String	This parameter is available only in the cloud-based service. The *OrganizationDomain* parameter specifies the domain name of the Office 365 organization. For example, `contoso.com`.
ReloadConfig	Optional	System.Management.Automation.SwitchPar	The *ReloadConfig* switch reloads all the

		ameter	configuration settings from the Exchange configuration objects. You don't need to specify a value with this switch. If you don't use this switch, the cached configuration settings are used.
UseCachedToken	Optional	System.Management.Automation.SwitchParameter	The *UseCachedToken* switch specifies that OAuth will try to use an existing, cached authorization token. You don't need to specify a value with this switch.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PartnerApplication

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-PartnerApplication** cmdlet to retrieve settings for a partner application.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PartnerApplication [-Identity <PartnerApplicationIdParameter>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves settings for all partner applications configured in Exchange and pipes them to the **Format-List** cmdlet to display all properties in a list view.

```
Get-PartnerApplication | Format-List *
```

Detailed Description

In Microsoft Exchange Server 2013, you can configure partner applications such as Microsoft SharePoint to access Exchange resources. For details, see Integration with SharePoint and Lync.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.PartnerApplicationIdParameter	The *Identity* parameter specifies the identity of a partner application.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-PartnerApplication

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-PartnerApplication** cmdlet to create a partner application configuration.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-PartnerApplication -ApplicationIdentifier <String> [-Realm <String>]
<COMMON PARAMETERS>


New-PartnerApplication -AuthMetadataUrl <String> [-TrustAnySSLCertificate
<SwitchParameter>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Name <String> [-AcceptSecurityIdentifierInformation
<$true | $false>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-Enabled <$true | $false>] [-LinkedAccount <UserIdParameter>] [-
Organization <OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the HRApp partner application and configures it to use an authorization server.

```
New-PartnerApplication HRApp -ApplicationIdentifier
00000006-0000-0dd1-ac00-000000000000 -Realm contoso.com -
UseAuthServer $true
```

Detailed Description

In Microsoft Exchange Server 2013, you can configure partner applications such as Microsoft SharePoint to access Exchange resources. Use the **New-PartnerApplication** cmdlet to create a partner application configuration for an application that needs to access Exchange 2013 resources. For details, see Integration with SharePoint and Lync.

We recommend that you use the Configure-EnterprisePartnerApplication.ps1 script in the \Exchange Server\V15\Scripts folder to configure partner applications.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
ApplicationIdentifier	Required	System.String	The *ApplicationIdentifier* parameter specifies a unique application identifier for the partner application that uses an authorization server. When specifying a value for the *ApplicationIdentifier* parameter, you must also use the *UseAuthServer* parameter.
AuthMetadataUrl	Required	System.String	This parameter is available only in on-premises Exchange 2013. The *AuthMetadataUrl* parameter specifies the URL that Exchange can retrieve the AuthMetadata document from for a partner application that doesn't use an authorization server. When specifying the *AuthMetadataUrl* parameter for a partner application, you can't specify the *ApplicationIdentifier* and

			UseAuthServer parameters.
Name	Required	System.String	The *Name* parameter specifies a name for the partner application.
AcceptSecurityIdentifierInformation	Optional	System.Boolean	The *AcceptSecurityIdentifierInformation* parameter specifies whether Exchange should accept security identifiers (SIDs) from another trusted Active Directory forest for the partner application. By default, new partner applications are configured to not accept SIDs from another forest. If you're in deployment with a trusted forest, set the parameter to `$true`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the partner application is enabled. By default, new partner applications are enabled. Set the parameter to `$false` to create the application configuration in a disabled state.
LinkedAccount	Optional	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	The *LinkedAccount* parameter specifies a linked Active Directory user account for the application. Exchange evaluates Role Based Access Control (RBAC) permissions for the linked account when authorizing a token used to perform a task.

Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Realm	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *Realm* parameter specifies a security realm for the partner application. If the token is from a domain that's not an accepted domain, Exchange checks the realm specified in the token. In such a scenario, only tokens with the same realm specified in the partner application can access Exchange 2013 resources.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *TrustAnySSLCertificate* switch specifies whether Exchange should trust certificates issued by a certification authority (CA) not trusted by the server.

			Caution: We don't recommend using this switch in a production environment.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-PartnerApplication

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-PartnerApplication** cmdlet to remove a partner application from Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-PartnerApplication -Identity <PartnerApplicationIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This command removes the HRApp partner application.

```
Remove-PartnerApplication HRApp
```

Detailed Description

In Exchange 2013, you can configure partner applications such as Microsoft SharePoint to access Exchange Server resources. Use the **Remove-PartnerApplication** cmdlet to remove a partner application configuration if the application no longer needs to access Exchange 2013 resources. For details, see Integration with SharePoint and Lync.

We recommend that you use the Configure-EnterprisePartnerApplication.ps1 script in the \Exchange Server\V15\Scripts folder to configure partner applications.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.PartnerApplicationIdParameter	The *Identity* parameter specifies the identity of the partner application.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt

			that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-PartnerApplication

Exchange Management Shell > Exchange 2013 cmdlets > Security cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-21*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-PartnerApplication** cmdlet to configure a partner application.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-PartnerApplication [-ApplicationIdentifier <String>] [-Realm <String>]
<COMMON PARAMETERS>


Set-PartnerApplication [-AuthMetadataUrl <String>] [-
TrustAnySSLCertificate <SwitchParameter>] <COMMON PARAMETERS>


Set-PartnerApplication [-RefreshAuthMetadata <SwitchParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: -Identity <PartnerApplicationIdParameter> [-
AcceptSecurityIdentifierInformation <$true | $false>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Enabled <$true |
$false>] [-LinkedAccount <UserIdParameter>] [-Name <String>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example refreshes the auth metadata for the HRApp partner application.

```
Set-PartnerApplication HRApp -RefreshAuthMetadata
```

Detailed Description

In Microsoft Exchange Server 2013, you can configure partner applications such as Microsoft SharePoint to access Exchange resources. Use the **New-PartnerApplication** cmdlet to create a partner application configuration for an application that needs to access Exchange 2013 resources. For details, see Integration with SharePoint and Lync. We recommend that you use the Configure-EnterprisePartnerApplication.ps1 script in the \Exchange Server\V15\Scripts folder to configure partner applications.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Partner applications - configure" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.PartnerApplicationIdParameter	The *Identity* parameter specifies the identity of the partner application you want to modify.
AcceptSecurityIdentifierInformation	Optional	System.Boolean	The *AcceptSecurityIdentifierInformation* parameter specifies whether Exchange should accept security identifiers (SIDs) from another trusted Active Directory forest for the partner application. By default, new partner applications are configured to not

			accept SIDs from another forest. If you're in deployment with a trusted forest, set the parameter to `$true`.
ApplicationIdentifier	Optional	System.String	The *ApplicationIdentifier* parameter specifies a unique application identifier for the partner application that uses an authorization server.
AuthMetadataUrl	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *AuthMetadataUrl* parameter specifies the URL that Exchange can retrieve the AuthMetadata document from for a partner application that doesn't use an authorization server.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run.

			To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether the partner application is enabled. By default, new partner applications are enabled. Set the parameter to `$false` to create the application configuration in a disabled state.
LinkedAccount	Optional	Microsoft.Exchange.Configuration.Tasks.UserI	The *LinkedAccount* parameter specifies a

		dParameter	linked Active Directory user account for the application. Exchange evaluates Role Based Access Control (RBAC) permissions for the linked account when authorizing a token used to perform a task.
Name	Optional	System.String	The *Name* parameter specifies a new name for the partner application.
Realm	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *Realm* parameter specifies a security realm for the partner application. If the token is from a domain that's not an accepted domain, Exchange checks the realm specified in the token. In such a scenario, only tokens with the same realm specified in the partner application can access Exchange 2013 resources.

RefreshAuthMetadata	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RefreshAuthMetadata* switch specifies that the auth metadata should be refreshed from the authorization server.
TrustAnySSLCertificate	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *TrustAnySSLCertificate* switch specifies whether Exchange should trust certificates issued by a certification authority (CA) not trusted by the server. We don't recommend using this switch in a production environment.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the

			WhatIf switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Server health, monitoring, and performance cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-22*

Exchange workload management cmdlets

Get-ThrottlingPolicy

New-ThrottlingPolicy

Remove-ThrottlingPolicy

Set-ThrottlingPolicy

Get-ThrottlingPolicyAssociation

Set-ThrottlingPolicyAssociation

Managed availability cmdlets

Add-GlobalMonitoringOverride

Get-GlobalMonitoringOverride

Remove-GlobalMonitoringOverride

Get-HealthReport

Get-MonitoringItemHelp

Get-MonitoringItemIdentity

Get-ServerComponentState

Set-ServerComponentState

Add-ServerMonitoringOverride

Get-ServerMonitoringOverride

Remove-ServerMonitoringOverride

Exchange server monitoring cmdlets

Get-EventLogLevel

Set-EventLogLevel

Invoke-MonitoringProbe

Get-ServerHealth

Set-ServerMonitor

Test-ServiceHealth

Get-EventLogLevel

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-EventLogLevel** cmdlet to display a list of event categories and log levels for a

specified computer running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-Eventloglevel -Server <ServerIdParameter> <COMMON PARAMETERS>

Get-Eventloglevel [-Identity <ECIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example displays the event categories and log levels for the server Exchange01.

```
Get-EventLogLevel -Server "Exchange01"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange server configuration settings" entry in the Exchange and Shell infrastructure permissions topic.

Note:
You can specify either the *Server* or *Identity* parameter, but not both.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server for which you want to review event categories and log levels.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ECIdParameter	The *Identity* parameter specifies the name of the event category and log level to display.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-EventLogLevel

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-EventLogLevel** cmdlet to set the event log level registry value for the specified category.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-Eventloglevel -Identity <ECIdParameter> -Level <Lowest | Low | Medium
| High | Expert> [-Confirm [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the event log level to High for the MSExchangeTransport\SmtpReceive event logging category on the Exchange server Exchange01.

Note:
Run the **Get-EventLogLevel** cmdlet to retrieve a list of the event categories on your server. For more information, see Get-EventLogLevel.

```
Set-EventLogLevel -Identity "Exchange01\MSExchangeTransport
\SmtpReceive" -Level High
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ECIdParameter	The *Identity* parameter specifies the name of the event logging category for which you want to set the event logging level.
Level	Required	Microsoft.Exchange.Diagnostics.ExEventLog	The *Level* parameter specifies the log level for

		+EventLevel	the specific event logging category. The valid values are: • `Lowest` • `Low` • `Medium` • `High` • `Expert`
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-GlobalMonitoringOverride

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-21*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Add-GlobalMonitoringOverride** cmdlet to override the thresholds and parameters used by the probes, monitors, and responders on all Exchange 2013 servers in an Exchange Server 2013 environment. The cmdlet enables monitoring changes and threshold tuning to the environment.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-GlobalMonitoringOverride [-Duration <EnhancedTimeSpan>] <COMMON
PARAMETERS>


Add-GlobalMonitoringOverride -ApplyVersion <Version> <COMMON PARAMETERS>


COMMON PARAMETERS: -Identity <String> -ItemType <Probe | Monitor |
Responder | Maintenance> -PropertyName <String> -PropertyValue <String> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds a global monitoring override that disables the OnPremisesInboundProxy probe for 30 days for all servers running version 15.00.0847.32.

```
Add-GlobalMonitoringOverride -Identity "FrontendTransport
\OnPremisesInboundProxy" -PropertyName Enabled -
PropertyValue 0 -Duration 30.00:00:00 -ItemType Probe -
ApplyVersion "15.00.0847.32"
```

EXAMPLE 2

This example adds a global monitoring override that disables the StorageLogicalDriveSpaceEscalate responder for 21 days for all servers running version 15.00.0847.32.

```
Add-GlobalMonitoringOverride -Identity "MailboxSpace
\StorageLogicalDriveSpaceEscalate" -PropertyName Enabled -
PropertyValue 0 -Duration 21.00:00:00 -ItemType Responder -
ApplyVersion "15.00.0847.32"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
ApplyVersion	Required	System.Version	The *ApplyVersion* parameter specifies the version of override. The valid input for this parameter is a number in the format 15.00.xxxx.xxx.
Identity	Required	System.String	The *Identity* parameter specifies the identity of the server to access.
ItemType	Required	Microsoft.Exchange.Data.MonitoringItemTypeEnum	The *ItemType* parameter specifies the item type that you want to designate the new property. It can be any of the following values: • `Probe` • `Monitor` • `Responder`

PropertyName	Required	System.String	The *PropertyName* parameter specifies the property you want to override.
PropertyValue	Required	System.String	The *PropertyValue* parameter specifies the new value of the property you are trying to override.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the

			local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Duration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *Duration* parameter specifies the length of time to keep the monitoring override. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, 30.10:00:00 specifies 30 days and 10 hours.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-GlobalMonitoringOverride

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-GlobalMonitoringOverride** cmdlet to retrieve the thresholds and parameters used by the probes, monitors, and responders in a Microsoft Exchange Server 2013 environment.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-GlobalMonitoringOverride [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves the override settings and values for the Contoso domain.

```
Get-GlobalMonitoringOverride -DomainController Contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-GlobalMonitoringOverride

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-GlobalMonitoringOverride** cmdlet to remove a managed availability global override that has been configured for a probe, monitor, or responder.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-GlobalMonitoringOverride -Identity <String> -ItemType <Probe |
Monitor | Responder | Maintenance> -PropertyName <String> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes a global monitoring override of the ActiveDirectoryConnectivityConfigDCServerReboot responder in the Exchange health set.

```
Remove-GlobalMonitoringOverride -Identity Exchange
\ActiveDirectoryConnectivityConfigDCServerReboot -ItemType
Responder -PropertyName Enabled
```

EXAMPLE 2

This example removes a global monitoring override of the ExtensionAttributes property of the OnPremisesInboundProxy probe in the FrontEndTransport health set.

```
Remove-GlobalMonitoringOverride -Identity FrontEndTransport
\OnPremisesInboundProxy -ItemType Probe -PropertyName
ExtensionAttributes
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell Infrastructure Permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	System.String	The *Identity* parameter

			specifies the monitoring item that was overridden. The value is in the form of HealthSet \MonitoringItem, or HealthSet\MonitoringItem \TargetResource.
ItemType	Required	Microsoft.Exchange.Data.MonitoringItemTypeEnum	The *ItemType* parameter specifies the item type that you want to remove. It can be any of the following values: • Probe • Monitor • Responder
PropertyName	Required	System.String	The *PropertyName* parameter specifies the property for the override you want to remove.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-HealthReport

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-HealthReport** cmdlet to return health information related to the server you specify. You can use the health values to determine the state of the server.The cmdlet also returns an alert value that provides the specific state of your server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-HealthReport -Identity <ServerIdParameter> [-GroupSize <Int32>] [-
HaImpactingOnly <SwitchParameter>] [-HealthSet <String>] [-
MinimumOnlinePercent <Int32>] [-RollupGroup <SwitchParameter>]
```

Examples

EXAMPLE 1

This example retrieves health information about a server running Microsoft Exchange Server 2013.

```
Get-HealthReport -RollupGroup
```

Detailed Description

The following list contains the health values that are returned:

- Online
- Partially Online
- Offline
- Sidelined
- Functional
- Unavailable

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Test system health" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the identity of the server you want health information for.
GroupSize	Optional	System.Int32	The *GroupSize* parameter determines the size of the group to process against for a rollup. The default value is 12.
HaImpactingOnly	Optional	System.Management.Automation.SwitchParameter	The *HaImpactingOnly* switch specifies whether the cmdlet must roll up only the monitors that have **HaImpacting** set to `True`.
HealthSet	Optional	System.String	The *HealthSet* parameter returns the health state of a group of monitors. Monitors that are similar or are tied to a component's architecture are grouped to form a

			health set. You can determine the collection of monitors (and associated probes and responders) in a given health set by using the *Get-MonitoringItemIdentity* cmdlet.
MinimumOnlinePercent	Optional	System.Int32	The *MinimumOnlinePercent* parameter specifies the number of members in the group to be functioning with rollup information Degraded instead of Unhealthy. The default value is 70 percent.
RollupGroup	Optional	System.Management.Automation.SwitchParameter	The *RollupGroup* parameter specifies that the health data is rolled up across servers with redundancy limits.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MonitoringItemHelp

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MonitoringItemHelp** cmdlet to discover the monitoring items that you can use to return health information about your Exchange servers. Monitoring items are preconfigured to help you with your server health and monitoring.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MonitoringItemHelp -Identity <String> -Server <ServerIdParameter>
```

Examples

EXAMPLE 1

This example retrieves health set information on the server Exch01.

```
Get-MonitoringItemHelp -"HealthSet\MonitorName" -Server
Exch01
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange server configuration settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	System.String	The *Identity* parameter specifies the identity of

			the monitoring item.
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server running Microsoft Exchange Server 2013 to query for health set information. The default is the local Exchange 2013 server.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MonitoringItemIdentity

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-MonitoringItemIdentity** cmdlet to discover the monitoring items that you can use to return health information about your Exchange servers.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MonitoringItemIdentity -Identity <String> -Server <ServerIdParameter>
```

Examples

EXAMPLE 1

This example retrieves monitoring information about the HealthSet01 monitoring item on the ExchSrv01 server.

```
Get-MonitoringItemIdentity -Identity HealthSet01 -Server ExchSrv01
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Exchange server configuration settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	System.String	The *Identity* parameter specifies the identity of the monitoring item on a server running Microsoft Exchange Server 2013.
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange 2013 server to query for health set information.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Invoke-MonitoringProbe

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-06-25*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Invoke-MonitoringProbe** cmdlet to run a Managed Availability probe on a selected server.

For information about the parameter sets in the Syntax section below, see Syntax.

Note:

This cmdlet cannot be used to run every Managed Availability probe. Only some probes (mainly the deep test probes) can be run manually using this cmdlet. Probes that cannot be run with this cmdlet will generate an error message when Invoke-MonitoringProbe is used to run them.

```
Invoke-MonitoringProbe -Identity <String> -Server <ServerIdParameter> [-
Account <String>] [-Endpoint <String>] [-ItemTargetExtension <String>] [-
Password <String>] [-PropertyOverride <String>] [-SecondaryAccount
<String>] [-SecondaryEndpoint <String>] [-SecondaryPassword <String>] [-
TimeOutSeconds <String>]
```

Examples

EXAMPLE 1

This example creates an Exchange ActiveSync monitoring probe on the EX1 server.

```
Invoke-MonitoringProbe -Identity ActiveSync.Protocol
\ActiveSyncSelfTestProbe -Server EX1
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	System.String	The *Identity* parameter specifies the identity of the monitoring probe to run.
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can't use this parameter to configure other Edge Transport servers remotely.
Account	Optional	System.String	The *Account* parameter specifies the identity of the mailbox or user account that will run the monitoring probe.

Endpoint	Optional	System.String	The *Endpoint* parameter specifies the name of the monitoring probe endpoint to connect to, for example, contoso.mail.com.
ItemTargetExtension	Optional	System.String	The *ItemTargetExtension* parameter specifies cmdlet extension data that you can pass to the monitoring probe. The probe that runs on the server may require specific data for its execution. This data is presented to the probe on execution in an XML format.
Password	Optional	System.String	The *Password* parameter specifies the password of the mailbox or user account that will run the monitoring probe.
PropertyOverride	Optional	System.String	The *PropertyOverride* parameter specifies a property that you want to override, for example, to set the time-out value to be extended beyond the default value.
SecondaryAccount	Optional	System.String	The *SecondaryAccount*

			parameter specifies the identity of the delegate mailbox or user account that will run the monitoring probe.
SecondaryEndpoint	Optional	System.String	The *SecondaryEndpoint* parameter specifies the name of the secondary monitoring probe endpoint to connect to, for example, contoso.mail.fabrikam.com.
SecondaryPassword	Optional	System.String	The *SecondaryPassword* parameter specifies the password of the delegate mailbox or user account that will run the monitoring probe.
TimeOutSeconds	Optional	System.String	The *TimeOutSeconds* parameter specifies the monitoring operation time-out period.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ServerComponentState

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ServerComponentState** cmdlet to retrieve configuration settings for Microsoft Exchange Server 2013 components and endpoints.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ServerComponentState -Identity <ServerIdParameter> [-Component
<String>] [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves the component state for the Unified Messaging component.

```
Get-ServerComponentState -Component UnifiedMessaging
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Test system health" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the requester.
Component	Optional	System.String	The *Component* parameter specifies the

			component or endpoint for which you want to retrieve the state.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ServerComponentState

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-15*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ServerComponentState** cmdlet to configure and update Microsoft Exchange Server 2013 components and endpoints on servers you specify.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ServerComponentState -Component <String> -Identity <ServerIdParameter>
-Requester <String> -State <Inactive | Active | Draining> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-LocalOnly
<SwitchParameter>] [-RemoteOnly <SwitchParameter>] [-TimeoutInSeconds
<Int32>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the Unified Messaging (UM) component state to Active, as requested by maintenance mode.

```
Set-ServerComponentState -Component UMCallRouter -Identity
MailboxServer01 -Requester Maintenance -State Active
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Test system health" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Component	Required	System.String	The *Component* parameter specifies the component or endpoint for which you want to set the state.

Identity	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the server you want to configure.
Requester	Required	System.String	The *Requester* parameter specifies the system requesting this state change. The string can be one of the following: • `HealthAPI` • `Maintenance` • `Sidelined` • `Functional` • `Deployment`
State	Required	Microsoft.Exchange.Data.ServiceState	The *State* parameter specifies the state that you want for the component. The state can be one of the following: • `Active` • `Inactive` • `Draining`
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
LocalOnly	Optional	System.Management.Automation.SwitchParameter	The *LocalOnly* parameter specifies that the cmdlet changes should be written to the Windows registry only.
RemoteOnly	Optional	System.Management.Automation.SwitchParameter	The *RemoteOnly* parameter specifies that the cmdlet changes should be written to Active Directory only.
TimeoutInSeconds	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ServerHealth

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ServerHealth** cmdlet to return health information related to the server you specify.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ServerHealth -Identity <ServerIdParameter> [-HaImpactingOnly
<SwitchParameter>] [-HealthSet <String>]
```

Examples

EXAMPLE 1

This example returns the server health for server Server01.

```
Get-ServerHealth -Identity Server01
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Test system health" entry in the Exchange and Shell infrastructure permissions topic.

The cmdlet returns health values that you can use to determine the state of the server. See Server health and performance for related information.

The cmdlet also returns an alert value that provides the specific state of your server. The following values may be returned:

- Degraded
- Unhealthy
- Repairing
- Disabled
- Unavailable
- UnInitialized

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Identity* parameter specifies the identity of the server you want health information for.
HaImpactingOnly	Optional	System.Management.Automation.SwitchParameter	The *HaImpactingOnly* switch specifies whether the cmdlet must roll up only the monitors that have **HaImpacting** set to `True`.
HealthSet	Optional	System.String	The *HealthSet* parameter returns the

			health state of a group of monitors. Monitors that are similar or are tied to a component's architecture are grouped to form a *health set*. You can determine the collection of monitors (and associated probes and responders) in a given health set by using the *Get-MonitoringItemIdentity* cmdlet.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ServerMonitor

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-14*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ServerMonitor** cmdlet to edit or set a parameter on a single monitor on an Exchange

server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ServerMonitor -Name <String> -Repairing <$true | $false> -Server
<ServerIdParameter> [-Confirm [<SwitchParameter>]] [-TargetResource
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the maintenance monitor on the Exch01 server.

```
Set-ServerMonitor -Name Maintenance -Repairing $true -
Server Exch01
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the monitor identity.
Repairing	Required	System.Boolean	The *Repairing* parameter specifies whether to set or clear the repairing property on a monitor. The default value is `$true`.
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You

			can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
TargetResource	Optional	System.String	The *TargetResource* parameter specifies the target resource that you want to set the monitor on.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-ServerMonitoringOverride

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-15*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Add-ServerMonitoringOverride** cmdlet to override the properties of managed availability probes, monitors, and responders on a server running Microsoft Exchange Server 2013.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-ServerMonitoringOverride [-Duration <EnhancedTimeSpan>] <COMMON
PARAMETERS>


Add-ServerMonitoringOverride -ApplyVersion <Version> <COMMON PARAMETERS>


COMMON PARAMETERS: -Identity <String> -ItemType <Probe | Monitor |
Responder | Maintenance> -PropertyName <String> -PropertyValue <String> -
Server <ServerIdParameter> [-Confirm [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds a maintenance server monitoring override for 20 days.

```
Add-ServerMonitoringOverride -Duration 20.00:00:00 -
Identity Server01 -ItemType Monitor -PropertyName Enabled -
PropertyValue 0
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell infrastructure permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
ApplyVersion	Required	System.Version	The *ApplyVersion* parameter specifies the version of override. The valid input for this parameter is a number in the format 15.00.xxxx.xxx.
Identity	Required	System.String	The *Identity* parameter specifies HealthSet \MonitoringItemName \TargetResource. You can use *Get-ServerHealth* to find the correct object for the monitoring item you want to override.

ItemType	Required	Microsoft.Exchange.Data.MonitoringItemTypeEnum	The *ItemType* parameter specifies the server item type that you want to designate the new property. It can be any of the following values: • `Probe` • `Monitor` • `Responder`
PropertyName	Required	System.String	The *PropertyName* parameter specifies the server property you want to override.
PropertyValue	Required	System.String	The *PropertyValue* parameter specifies the new value for the server property.
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server.

			You can't use this parameter to configure other Edge Transport servers remotely.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Duration	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *Duration* parameter specifies the length of time to keep the monitoring override. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, 30.10:00:00 specifies 30 days and 10 hours.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ServerMonitoringOverride

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ServerMonitoringOverride** cmdlet to return all overrides created on the specified server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ServerMonitoringOverride -Server <ServerIdParameter>
```

Examples

EXAMPLE 1

This example retrieves all monitoring overrides for the Exch01 server.

```
Get-ServerMonitoringOverride -Server Exch01
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Test system health" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Exchange server on which you want to run this command. You can use any value that uniquely identifies the server. For example: • Name • FQDN • Distinguished name (DN) • Exchange Legacy DN If you don't use the *Server* parameter, the command is run on the local server. You can't use this parameter to configure other Edge Transport servers remotely.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ServerMonitoringOverride

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-ServerMonitoringOverride** cmdlet to remove a managed availability local server override that has been configured for a probe, monitor, or responder.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ServerMonitoringOverride -Identity <String> -ItemType <Probe |
Monitor | Responder | Maintenance> -PropertyName <String> -Server
<ServerIdParameter> [-Confirm [<SwitchParameter>]] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes a server monitoring override of the ActiveDirectoryConnectivityConfigDCServerReboot responder in the Exchange health set from server EX1.

```
Remove-ServerMonitoringOverride -Server EX1 -Identity
Exchange\ActiveDirectoryConnectivityConfigDCServerReboot -
ItemType Responder -PropertyName Enabled
```

Example 2

This example removes a server monitoring override of the ExtensionAttributes property of the OnPremisesInboundProxy probe in the FrontEndTransport health set from server EX2.

```
Remove-ServerMonitoringOverride -Server EX2 -Identity
FrontEndTransport\OnPremisesInboundProxy -ItemType Probe -
PropertyName ExtensionAttributes
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Shell Infrastructure Permissions" section in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	System.String	The *Identity* parameter specifies the monitoring item that was overridden. The value is in the form of HealthSet \MonitoringItem, or HealthSet\MonitoringItem \TargetResource.
ItemType	Required	Microsoft.Exchange.Data.MonitoringItemTypeEnum	The *ItemType* parameter specifies the item type of the override that you want to remove. It can be any of the following values: • Probe • Monitor • Responder
PropertyName	Required	System.String	The *PropertyName* parameter specifies the property for the override you want to remove.

Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter is used to specify the server from which the override is being removed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-ServiceHealth

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-ServiceHealth** cmdlet to test whether all the Microsoft Windows services that Exchange requires on a server have started. The **Test-ServiceHealth** cmdlet returns an error for any service required by a configured role when the service is set to start automatically and isn't currently running.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-ServiceHealth [-ActiveDirectoryTimeout <Int32>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-MonitoringContext <$true
| $false>] [-Server <ServerIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example uses the **Test-ServiceHealth** command without parameters to test the services on the local server.

```
Test-ServiceHealth
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Test system health" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description

ActiveDirectoryTimeout	Optional	System.Int32	The *ActiveDirectoryTimeout* parameter specifies the amount of time, in seconds, allowed for each Active Directory operation to complete before the operation times out. The default value is 15 seconds.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An

			Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
MonitoringContext	Optional	System.Boolean	The *MonitoringContext* parameter includes or excludes the associated monitoring events and performance counters in the results. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you specify the value `$true`, the monitoring events and performance counters are included in the command results. Typically, you include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.

Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the server on which to check that the required services are running. If you don't specify this parameter, the command checks the services on the local server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ThrottlingPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ThrottlingPolicy** cmdlet to view the user throttling settings for one or more throttling policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ThrottlingPolicy [-Identity <ThrottlingPolicyIdParameter>] [-
Diagnostics <SwitchParameter>] [-DomainController <Fqdn>] [-Explicit
<SwitchParameter>] [-IgnoreDehydratedFlag <SwitchParameter>] [-
Organization <OrganizationIdParameter>] [-ThrottlingPolicyScope <Regular |
Organization | Global>]
```

Examples

EXAMPLE 1

This example returns the settings for all throttling policies.

```
Get-ThrottlingPolicy | Format-List
```

EXAMPLE 2

This example displays the parameters and values for throttling policy ThrottlingPolicy2.

```
Get-ThrottlingPolicy -Identity ThrottlingPolicy2 | Format-
List
```

Detailed Description

The **Get-ThrottlingPolicy** cmdlet returns the client throttling settings for one or more throttling policies. If you use the *Identity* parameter, the cmdlet returns the settings for the identified throttling policy. If you don't use the *Identity* parameter, the cmdlet returns the settings for all throttling policies.

For more information about how to control the resources consumed by individual users, see Exchange workload management.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "User throttling" entry in the Server health and performance permissions topic.

Parameters

Parameter	Required	Type	Description
Diagnostics	Optional	System.Management.Automation.SwitchParameter	The *Diagnostics* switch specifies whether you want the output to include the diagnostics string. To return diagnostics information, use the syntax `Get-ThrottlingPolicy -Diagnostics`. You don't specify a value for this switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write

			data.
Explicit	Optional	System.Management.Automation.SwitchParameter	The *Explicit* switch specifies whether you only want to return the policy settings that have been directly assigned using this policy. By default, this parameter is set to `$false`.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ThrottlingPolicyIdParameter	The *Identity* parameter identifies the name of the throttling policy that you want to return settings for.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
ThrottlingPolicyScope	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ThrottlingPolicyScopeType	The *ThrottlingPolicyScope* parameter specifies the scope of the throttling policy. You can use the following values: • `Global` • `Organization` • `Regular` For information about each of these policy scopes, see Exchange

			workload management.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-ThrottlingPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-15*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **New-ThrottlingPolicy** cmdlet to create a non-default user throttling policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-ThrottlingPolicy -Name <String> [-AnonymousCutoffBalance <Unlimited>]
[-AnonymousMaxBurst <Unlimited>] [-AnonymousMaxConcurrency <Unlimited>] [-
AnonymousRechargeRate <Unlimited>] [-ComplianceMaxExpansionDGRecipients
<Unlimited>] [-ComplianceMaxExpansionNestedDGs <Unlimited>] [-Confirm
[<SwitchParameter>]] [-CpaCutoffBalance <Unlimited>] [-CpaMaxBurst
<Unlimited>] [-CpaMaxConcurrency <Unlimited>] [-CpaRechargeRate
<Unlimited>] [-DiscoveryMaxConcurrency <Unlimited>] [-DiscoveryMaxKeywords
<Unlimited>] [-DiscoveryMaxKeywordsPerPage <Unlimited>] [-
DiscoveryMaxMailboxes <Unlimited>] [-DiscoveryMaxPreviewSearchMailboxes
<Unlimited>] [-DiscoveryMaxRefinerResults <Unlimited>] [-
DiscoveryMaxSearchQueueDepth <Unlimited>] [-
DiscoveryMaxStatsSearchMailboxes <Unlimited>] [-
DiscoveryPreviewSearchResultsPageSize <Unlimited>] [-
DiscoverySearchTimeoutPeriod <Unlimited>] [-DomainController <Fqdn>] [-
EasCutoffBalance <Unlimited>] [-EasMaxBurst <Unlimited>] [-
EasMaxConcurrency <Unlimited>] [-EasMaxDeviceDeletesPerMonth <Unlimited>]
[-EasMaxDevices <Unlimited>] [-EasMaxInactivityForDeviceCleanup
<Unlimited>] [-EasRechargeRate <Unlimited>] [-
EncryptionRecipientCutoffBalance <Unlimited>] [-
EncryptionRecipientMaxBurst <Unlimited>] [-
EncryptionRecipientMaxConcurrency <Unlimited>] [-
EncryptionRecipientRechargeRate <Unlimited>] [-
EncryptionSenderCutoffBalance <Unlimited>] [-EncryptionSenderMaxBurst
<Unlimited>] [-EncryptionSenderMaxConcurrency <Unlimited>] [-
EncryptionSenderRechargeRate <Unlimited>] [-EwsCutoffBalance <Unlimited>]
[-EwsMaxBurst <Unlimited>] [-EwsMaxConcurrency <Unlimited>] [-
EwsMaxSubscriptions <Unlimited>] [-EwsRechargeRate <Unlimited>] [-
ExchangeMaxCmdlets <Unlimited>] [-ForwardeeLimit <Unlimited>] [-
```

```
IgnoreDehydratedFlag <SwitchParameter>] [-ImapCutoffBalance <Unlimited>]
[-ImapMaxBurst <Unlimited>] [-ImapMaxConcurrency <Unlimited>] [-
ImapRechargeRate <Unlimited>] [-IsServiceAccount <SwitchParameter>] [-
MessageRateLimit <Unlimited>] [-Organization <OrganizationIdParameter>] [-
OutlookServiceCutoffBalance <Unlimited>] [-OutlookServiceMaxBurst
<Unlimited>] [-OutlookServiceMaxConcurrency <Unlimited>] [-
OutlookServiceMaxSocketConnectionsPerDevice <Unlimited>] [-
OutlookServiceMaxSocketConnectionsPerUser <Unlimited>] [-
OutlookServiceMaxSubscriptions <Unlimited>] [-OutlookServiceRechargeRate
<Unlimited>] [-OwaCutoffBalance <Unlimited>] [-OwaMaxBurst <Unlimited>] [-
OwaMaxConcurrency <Unlimited>] [-OwaRechargeRate <Unlimited>] [-
OwaVoiceCutoffBalance <Unlimited>] [-OwaVoiceMaxBurst <Unlimited>] [-
OwaVoiceMaxConcurrency <Unlimited>] [-OwaVoiceRechargeRate <Unlimited>] [-
PopCutoffBalance <Unlimited>] [-PopMaxBurst <Unlimited>] [-
PopMaxConcurrency <Unlimited>] [-PopRechargeRate <Unlimited>] [-
PowerShellCutoffBalance <Unlimited>] [-PowerShellMaxBurst <Unlimited>] [-
PowerShellMaxCmdletQueueDepth <Unlimited>] [-PowerShellMaxCmdlets
<Unlimited>] [-PowerShellMaxCmdletsTimePeriod <Unlimited>] [-
PowerShellMaxConcurrency <Unlimited>] [-PowerShellMaxDestructiveCmdlets
<Unlimited>] [-PowerShellMaxDestructiveCmdletsTimePeriod <Unlimited>] [-
PowerShellMaxOperations <Unlimited>] [-PowerShellMaxRunspaces <Unlimited>]
[-PowerShellMaxRunspacesTimePeriod <Unlimited>] [-
PowerShellMaxTenantConcurrency <Unlimited>] [-PowerShellMaxTenantRunspaces
<Unlimited>] [-PowerShellRechargeRate <Unlimited>] [-PswsMaxConcurrency
<Unlimited>] [-PswsMaxRequest <Unlimited>] [-PswsMaxRequestTimePeriod
<Unlimited>] [-PushNotificationCutoffBalance <Unlimited>] [-
PushNotificationMaxBurst <Unlimited>] [-PushNotificationMaxBurstPerDevice
<Unlimited>] [-PushNotificationMaxConcurrency <Unlimited>] [-
PushNotificationRechargeRate <Unlimited>] [-
PushNotificationRechargeRatePerDevice <Unlimited>] [-
PushNotificationSamplingPeriodPerDevice <Unlimited>] [-RcaCutoffBalance
<Unlimited>] [-RcaMaxBurst <Unlimited>] [-RcaMaxConcurrency <Unlimited>]
[-RcaRechargeRate <Unlimited>] [-RecipientRateLimit <Unlimited>] [-
ThrottlingPolicyScope <Regular | Organization | Global>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a non-default user throttling policy that can be associated with specific users. Any parameters that you omit inherit the values from the default throttling policy GlobalThrottlingPolicy_<GUID>. After you create this policy, you must associate it with specific users.

```
New-ThrottlingPolicy -Name ITUserPolicy -EwsMaxConcurrency
4 -ThrottlingPolicyScope Regular
```

EXAMPLE 2

This example creates a policy that applies to all users in your organization. Any parameters that you omit inherit the values from the default throttling policy GlobalThrottlingPolicy_<GUID>.

```
New-ThrottlingPolicy -Name AllUsersEWSPolicy -
EwsMaxConcurrency 4 -ThrottlingPolicyScope Organization
```

EXAMPLE 3

This example creates a throttling policy RemoteSiteUserPolicy that restricts the number of connections for a user to three. The users associated with this policy are only able to create three

remote Exchange Management Shell sessions. This policy also restricts to three the number of Exchange Administration Center operations or Exchange Web Services operations that can be executed at the same time.

```
New-ThrottlingPolicy -Name RemoteSiteUserPolicy -
PowerShellMaxConcurrency 3 -PowerShellMaxCmdletQueueDepth
12
```

EXAMPLE 4

This example creates a throttling policy that restricts a user to be able to only execute 10 cmdlets in a period of five seconds. If the users associated with this policy exceed this number, the cmdlet pipeline execution is stopped with a throttling error message. The user needs to wait for and then resubmit the execution of cmdlets on the open connection.

```
New-ThrottlingPolicy -Name ITStaffUserPolicyCmdletMax -
PowerShellMaxCmdlets 10 -PowerShellMaxCmdletsTimePeriod 5
```

EXAMPLE 5

This example creates a throttling policy that restricts a user to be able to only execute 10 destructive cmdlets in 60 seconds. If the users associated with this policy exceed this number, the cmdlet pipeline execution is stopped with a throttling error message. The user needs to wait for and then resubmit the execution of cmdlets on the open connection.

```
New-ThrottlingPolicy -Name
ITStaffUserPolicyDestructiveCmdlets -
PowerShellMaxDestructiveCmdlets 10 -
PowerShellMaxDestructiveCmdletsTimePeriod 60
```

Detailed Description

By default, there is one default throttling policy named GlobalThrottlingPolicy_<GUID> with a throttling scope of `Global`. Microsoft Exchange Server 2013 Setup creates a default client throttling policy when you install the Exchange 2013 Client Access server role. You should not replace, re-create, or remove the existing default throttling policy. However, you can create additional throttling policies with the scope of Organization or Regular to change your user throttling settings. You can also edit policies with the scope of Organization and Regular that you've created using the **Set-ThrottlingPolicy** cmdlet.

For more information about how to control how resources are consumed by individual users, see Exchange workload management.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "User throttling" entry in the Server health and performance permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name of the object in Active Directory. The default policy is named DefaultThrottlingPolicy<*GUID*>.
AnonymousCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *AnonymousCutoffBalance* parameter specifies the resource consumption limits for an anonymous user before the user is completely blocked from performing operations on a specific component.
AnonymousMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *AnonymousMaxBurst* parameter specifies the amount of time that an anonymous user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each

			component.
AnonymousMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *AnonymousMaxConcurrency* parameter specifies how many anonymous connections can be made to a user's calendar data at the same time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If anonymous users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *AnonymousMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is `1`. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
AnonymousRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *AnonymousRechargeRate*

			parameter specifies the rate at which an anonymous user's budget is charged (budget grows by) during the budget time.
ComplianceMaxExpansionDGRecipients	Optional	Microsoft.Exchange.Data.Unlimited	The *ComplianceMaxExpansionDGRecipients* parameter specifies the maximum number of recipients to expand in distribution groups when a discovery search is looking for a specified recipient.
ComplianceMaxExpansionNestedDGs	Optional	Microsoft.Exchange.Data.Unlimited	The *ComplianceMaxExpansionNestedDGs* parameter specifies the maximum number of nested distribution groups to expand when a discovery search is looking for a specified recipient.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You

			don't have to specify a value with the *Confirm* switch.
CpaCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *CpaCutoffBalance* parameter specifies the resource consumption limits for a cross-premises user before that user is completely blocked from performing operations on a specific component.
CpaMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *CpaMaxBurst* parameter specifies the amount of time that a cross-premises user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
CpaMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *CpaMaxConcurrency* parameter specifies how many concurrent connections a cross-premises user can have against an Exchange server at one time. A connection is held from the moment a request is

			received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *CpaMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
CpaRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *CpaRechargeRate* parameter specifies the rate at which a cross premises user budget is charged (budget grows by) during the budget time.
DiscoveryMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxConcurrency* parameter specifies the number of concurrent discovery search executions that a user can have at the same time.

DiscoveryMaxKeywords	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxKeywords* parameter specifies the maximum number of keywords that a user can include in a discovery search. For more information, see Search-Mailbox.
DiscoveryMaxKeywordsPerPage	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxKeywordsPerPage* parameter specifies the number of keywords for which to show statistics on a single page in the Exchange Administration Center (EAC).
DiscoveryMaxMailboxes	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxMailboxes* parameter specifies the maximum number of source mailboxes that a user can include in a discovery search.
DiscoveryMaxPreviewSearchMailboxes	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxPreviewSearchMailboxes* parameter specifies the maximum number of mailboxes that a user can include in

			eDiscovery Search Preview.
DiscoveryMaxRefinerResults	Optional	Microsoft.Exchange.Data.Unlimited	This parameter isn't used and will be removed.
DiscoveryMaxSearchQueueDepth	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxSearchQueueDepth* parameter specifies the maximum number of concurrent discovery search threads that can be active at the same time.
DiscoveryMaxStatsSearchMailboxes	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxStatsSearchMailboxes* parameter specifies the maximum number of mailboxes that a user can search in an In-Place eDiscovery search without being able to view the statistics. When the number of mailboxes configured with the *DiscoveryMaxStatsSearchMailboxes* parameter is exceeded, the user must copy the search results to a discovery mailbox to view the statistics for the discovery search. For more information, see In-

			Place eDiscovery.
DiscoveryPreviewSearchResultsPageSize	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryPreviewSearchResultsPageSize* parameter specifies the number of messages displayed on a single page in eDiscovery Search Preview.
DiscoverySearchTimeoutPeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoverySearchTimeoutPeriod* parameter specifies the number of minutes that a discovery search will run before it times out.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD

			LDS) to read and write data.
EasCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *EasCutoffBalance* parameter specifies the resource consumption limits for a Microsoft Exchange ActiveSync user before that user is completely blocked from performing operations on a specific component.
EasMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *EasMaxBurst* parameter specifies the amount of time that an Exchange ActiveSync user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
EasMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *EasMaxConcurrency* parameter specifies how many concurrent connections an Exchange ActiveSync user can have against a server running Exchange 2013 at one time. A connection is held from the moment a

			request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *EasMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is `10`. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
EasMaxDeviceDeletesPerMonth	Optional	Microsoft.Exchange.Data.Unlimited	The *EasMaxDeviceDeletesPerMonth* parameter specifies a limit to the number of Exchange ActiveSync partnerships that a user can delete per month. By default, each user can delete a maximum of 20 partnerships per calendar month. When the limit is

			reached, the partnership deletion attempt fails and an error message is displayed to the user.
EasMaxDevices	Optional	Microsoft.Exchange.Data.Unlimited	The *EasMaxDevices* parameter specifies a limit to the number of Exchange ActiveSync partnerships that a user can have at one time. By default, each user can create 10 Exchange ActiveSync partnerships with their Exchange account. After users exceed the limit, they must delete one of their existing partnerships before they can create any more new partnerships. An email error message describing the limitation is sent to the user when the limit is exceeded. Additionally, an event is logged in the Application log when a user exceeds the limit.
EasMaxInactivityForDeviceCleanup	Optional	Microsoft.Exchange.Data.Unlimited	The *EasMaxInactivityForDeviceCleanup* parameter specifies the length of

			time that a user's device partnerships will remain active. By default, there is no limit to the number of days that a user's device partnerships will remain active. Use this value if you want to minimize the amount of inactive device partnerships in your organization. To use this setting, specify a value in days since the user's last sync time to cause the device partnership to be removed.
EasRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *EasRechargeRate* parameter specifies the rate at which an Exchange ActiveSync user's budget is charged (budget grows by) during the budget time.
EncryptionRecipientCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionRecipientMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionRecipientMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionRecipientRe	Optional	Microsoft.Exchange.Da	This parameter is reserved

chargeRate		ta.Unlimited	for internal Microsoft use.
EncryptionSenderCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionSenderMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionSenderMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionSenderRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EwsCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *EwsCutoffBalance* parameter specifies the resource consumption limits for an Exchange Web Services user before that user is completely blocked from performing operations on a specific component.
EwsMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *EwsMaxBurst* parameter specifies the amount of time that an Exchange Web Services user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.

EwsMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *EwsMaxConcurrency* parameter specifies how many concurrent connections an Exchange Web Services user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *EwsMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is `10`. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
EwsMaxSubscriptions	Optional	Microsoft.Exchange.Data.Unlimited	The *EwsMaxSubscriptions* parameter specifies the maximum number of active push and pull

			subscriptions that an Exchange Web Services user can have on a specified Client Access server at the same time. If a user tries to create more subscriptions than the configured maximum, the subscription fails, and an event is logged in Event Viewer.
EwsRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *EwsRechargeRate* parameter specifies the rate at which an Exchange Web Services user's budget is charged (budget grows by) during the budget time.
ExchangeMaxCmdlets	Optional	Microsoft.Exchange.Data.Unlimited	The *ExchangeMaxCmdlets* parameter specifies the number of cmdlets that can be executed within a specific time period before their execution is slowed down. The value specified by this parameter should be less than the value specified by the *PowerShellMaxCmdlets* parameter. The time period used for

			this limit is specified by the *PowerShellMaxCmdletsTimePeriod* parameter. We recommend that you set values for both parameters at the same time.
ForwardeeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *ForwardeeLimit* parameter specifies the limits for the number of recipients that can be configured in Inbox Rules when using the forward or redirect action. This parameter doesn't limit the number of messages that can be forwarded or redirected to the recipients that are configured.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ImapCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *ImapCutoffBalance* parameter specifies the resource consumption limits for an IMAP user before that user is completely blocked from performing operations on

			a specific component.
ImapMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *ImapMaxBurst* parameter specifies the amount of time that an IMAP user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
ImapMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *ImapMaxConcurrency* parameter specifies how many concurrent connections an IMAP user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *ImapMaxConcurrency* parameter has a valid range from 0 through

			2147483647 inclusive. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
ImapRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *ImapRechargeRate* parameter specifies the rate at which the IMAP user's budget is charged (budget grows by) during the budget time.
IsServiceAccount	Optional	System.Management.Automation.SwitchParameter	The *IsServiceAccount* switch specifies whether you want the user accounts associated with this policy to be moderated by the per-user thresholds specified by this policy, and also by additional throttling based on the health of system resources, such as overall CPU usage. This value is set to `$false` by default. You may want to set this value to `$true` if you intend to associate this policy with user accounts that require higher throttling limits. An

			account that might require higher throttling limits is a service account that performs a lot of non-interactive work (for example, service accounts that perform IMAP mailbox migrations or nightly Windows PowerShell tasks). By setting the *IsServiceAccount* switch to `$true`, work done by these accounts is moderated by the higher user throttling settings that you configure using the user throttling policy, but is slowed if resources start getting unhealthy.
MessageRateLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *MessageRateLimit* parameter specifies the number of messages per minute that can be submitted to transport. For messages submitted through the Mailbox server role (using Microsoft Outlook, Microsoft Office Outlook Web App, Exchange ActiveSync, or Exchange

			Web Services), this results in the deferral of messages until the quota for the user is available. Specifically, messages appear in the Outbox or Drafts folder for longer periods of time when users submit messages at a rate greater than the *MessageRateLimit* parameter. For POP or IMAP clients submitting messages directly to transport using SMTP, clients receive a transient error if they submit at a rate that exceeds the *MessageRateLimit* parameter. Exchange attempts to connect and send the messages at a later time.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
OutlookServiceCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OutlookServiceMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.

OutlookServiceMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OutlookServiceMaxSocketConnectionsPerDevice	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OutlookServiceMaxSocketConnectionsPerUser	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OutlookServiceMaxSubscriptions	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OutlookServiceRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OwaCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaCutoffBalance* parameter specifies the resource consumption limits for an Outlook Web App user before that user is completely blocked from performing operations on a specific component.
OwaMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaMaxBurst* parameter specifies the amount of time that an Outlook Web App user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is

			set separately for each component.
OwaMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaMaxConcurrency* parameter specifies how many concurrent connections an Outlook Web App user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *OwaMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is `5`. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
OwaRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaRechargeRate* parameter specifies the

			rate at which an Outlook Web App user's budget is charged (budget grows by) during the budget time.
OwaVoiceCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaVoiceCutoffBalance* parameter specifies the resource consumption limits for an Outlook Web App voice user before that user is completely blocked from performing operations on a specific component.
OwaVoiceMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaVoiceMaxBurst* parameter specifies the amount of time that an Outlook Web App voice user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
OwaVoiceMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaVoiceMaxConcurrency* parameter specifies how many concurrent connections an Outlook

			Web App voice user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *OwaVoiceMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is `5`. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
OwaVoiceRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaVoiceRechargeRate* parameter specifies the rate at which an Outlook Web App voice user's budget is charged (budget grows by) during the

			budget time.
PopCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *PopCutoffBalance* parameter specifies the resource consumption limits for a user before that user is completely blocked from performing operations on a specific component.
PopMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *PopMaxBurst* parameter specifies the amount of time that a user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
PopMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *PopMaxConcurrency* parameter specifies how many concurrent connections a POP user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make

			more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *PopMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is `20`. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
PopRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *PopRechargeRate* parameter specifies the rate at which the user budget is charged (budget grows by) during the budget time.
PowerShellCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellCutoffBalance* parameter specifies the resource consumption limits for a user before that user is completely blocked from performing operations on a specific component.

PowerShellMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxBurst* parameter specifies the amount of time that a user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
PowerShellMaxCmdletQueueDepth	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxCmdletQueueDepth* parameter specifies the number of operations allowed to be executed by the user. This value directly affects the behavior of the *PowerShellMaxCmdlets* and *PowerShellMaxConcurrency* parameters. For example, the *PowerShellMaxConcurrency* parameter consumes at least two operations defined by the *PowerShellMaxCmdletQueueDepth* parameter but additional operations are also consumed per cmdlet execution. The number of

			operations depends on the cmdlets executed. We recommend that the value for the *PowerShellMaxCmdletQueueDepth* parameter be at least three times larger than the value of the *PowerShellMaxConcurrency* parameter. This parameter won't affect Exchange Administration Center operations or Exchange Web Services operations.
PowerShellMaxCmdlets	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxCmdlets* parameter specifies the number of cmdlets that can be executed within a specific time period before their execution is stopped. The value specified by this parameter should be more than the value specified by the *ExchangeMaxCmdlets* parameter. The time period used for this limit is specified by the *PowerShellMaxCmdletsTi*

			mePeriod parameter. Both values should be set at the same time.
PowerShellMaxCmdletsTimePeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxCmdletsTimePeriod* parameter specifies the time period, in seconds, that the throttling policy uses to determine whether the number of cmdlets being executed exceeds the limits specified by the *PowerShellMaxCmdlets* and *ExchangeMaxCmdlets* parameters.
PowerShellMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxConcurrency* parameter specifies different information depending on context: • In the context of Remote PowerShell, the *PowerShellMaxConcurrency* parameter specifies the maximum number of Remote PowerShell sessions that a Remote PowerShell user can have open at the same time. • In the context of

			Exchange Web Services, the *PowerShellMaxConcurrency* parameter specifies the number of concurrent cmdlet executions that a user can have at the same time. This parameter value doesn't necessarily correlate to the number of browsers opened by the user.
PowerShellMaxDestructiveCmdlets	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxDestructiveCmdlets* parameter specifies the number of destructive cmdlets that can be executed within a specific time period before their execution is stopped. Destructive cmdlets are cmdlets that can make significant changes to user data and configuration settings in your Exchange organization. Throttling these cmdlets may help prevent accidental data loss. The following cmdlets are designated as

			destructive: • **Disable-Mailbox** • **Move-ActiveMailboxDatabase** • **Remove-AcceptedDomain** • **Remove-Mailbox** • **Remove-MailUser** • **Remove-Organization** • **Set-Mailbox** • **Set-MailUser** • **Update-MailboxDatabaseCopy** The time period used for this limit is specified by the *PowerShellMaxDestructiveCmdletsTimePeriod* parameter. Both values should be set at the same time. This feature isn't on by default. For more information, see the "Examples" section.
PowerShellMaxDestructiveCmdletsTimePeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxDestructiveCmdletsTimePeriod* parameter specifies the time period, in seconds, that the throttling policy

			uses to determine how many destructive cmdlets can be run. You set a value for this parameter when you set the *PowerShellMaxDestructiveCmdlets* parameter. Both values should be set at the same time. For more information, see the description for the *PowerShellMaxDestructiveCmdlets* parameter.
PowerShellMaxOperations	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxOperations* parameter specifies the protocol-level operations that are used to send and receive data. If the execution of a cmdlet results in a significant number of operations (for example, if there is a lot of input/output occurring), throttling may occur. The default setting is `Unlimited`.
PowerShellMaxRunspaces	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxRunspaces* parameter specifies the number of concurrent Windows PowerShell

			sessions that a user is allowed to have. The default setting is `Unlimited`.
PowerShellMaxRunspacesTimePeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxRunspacesTimePeriod* parameter specifies the time period, in seconds, that the throttling policy uses to determine how many Windows PowerShell sessions can be run. You set this value when you set the *PowerShellMaxRunspaces* parameter.
PowerShellMaxTenantConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxTenantConcurrency* parameter limits the number of concurrent Windows PowerShell connections per tenant organization. By default, the limit for concurrent Windows PowerShell connections per tenant organization is set to `9`. If users in a tenant organization try to make more concurrent requests than the limit set by the

			PowerShellMaxTenantConcurrency parameter, the new connection attempt fails. However, the existing connections remain valid. This limit is enforced even if a single user hasn't exceeded the per-user limit set by the *PowerShellMaxConcurrency* parameter. The *PowerShellMaxTenantConcurrency* parameter has a valid range from 0 through 100 inclusive. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`. **Note:** This property can only be set for the default throttling policy.
PowerShellMaxTenantRunspaces	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxTenantRunspaces* parameter specifies the number of concurrent Windows PowerShell sessions that a tenant is allowed to have.
PowerShellRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellRechargeRate*

			parameter specifies the rate at which the user budget is charged (budget grows by) during the budget time.
PswsMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *PswsMaxConcurrency* parameter specifies how many concurrent connections a Windows PowerShell Web Services user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *PswsMaxConcurrency* parameter has a default value of `18`. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.

PswsMaxRequest	Optional	Microsoft.Exchange.Data.Unlimited	The *PswsMaxRequest* parameter specifies how many requests a Windows PowerShell Web Services user can have against an Exchange server at one time. The default setting is `Unlimited`.
PswsMaxRequestTimePeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *PswsMaxRequestTimePeriod* parameter specifies the period of time, in seconds, that the throttling policy uses to determine how many requests can be run. The default setting is `Unlimited`.
PushNotificationCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
PushNotificationMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
PushNotificationMaxBurstPerDevice	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
PushNotificationMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
PushNotificationRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
PushNotificationRechargeRatePerDevice	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.

PushNotificationSamplingPeriodPerDevice	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
RcaCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *RcaCutoffBalance* parameter specifies the resource consumption limits for a user before that user is completely blocked from performing operations on a specific component.
RcaMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *RcaMaxBurst* parameter specifies the amount of time that a user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
RcaMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *RcaMaxConcurrency* parameter specifies how many concurrent connections an RPC Client Access user can have against a server running Exchange 2013 at one time. A connection is held from the moment a request is received until the connection is closed

			or the connection is otherwise disconnected (for example, if the user goes offline). If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *RcaMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is `20`. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
RcaRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *RcaRechargeRate* parameter specifies the rate at which the user budget is charged (budget grows by) during the budget time.
RecipientRateLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *RecipientRateLimit* parameter specifies the limits on the number of recipients that a user can address in a 24-hour period.

ThrottlingPolicyScope	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ThrottlingPolicyScopeType	The *ThrottlingPolicyScope* parameter specifies the scope of the throttling policy. You can use the following values. • `Regular` Specifies a custom policy that applies to specific users. • `Organization` Specifies a custom policy that applies to all users in your organization. • `Global` Reserved for the default throttling policy. For more information about throttling policy scopes, see Exchange workload management.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-ThrottlingPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-ThrottlingPolicy** cmdlet to remove a non-default Microsoft Exchange throttling policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-ThrottlingPolicy -Identity <ThrottlingPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Force <SwitchParameter>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the user throttling policy ClientThrottlingPolicy2.

```
Remove-ThrottlingPolicy -Identity ClientThrottlingPolicy2
```

EXAMPLE 2

You can't remove a policy that's associated with any users. This example reassigns the users subject to ClientThrottlingPolicy2 to the default policy. Then, it removes ClientThrottlingPolicy2.

```
$policy = Get-ThrottlingPolicy ClientThrottlingPolicy2;
$mailboxes = Get-Mailbox | where-object
{$_.ThrottlingPolicy -eq $policy.Identity};
$defaultPolicy = Get-ThrottlingPolicy | where-object
{$_.IsDefault -eq $true};
```

```
foreach ($mailbox in $mailboxes)
{
  set-mailbox -Identity $mailbox.Identity -ThrottlingPolicy
$defaultPolicy;
}
Remove-ThrottlingPolicy ClientThrottlingPolicy2;
```

Detailed Description

You can't remove the default client throttling policy. Also, you can't remove a policy associated with any users. For more information, see EXAMPLE 2.

For more information about how to control the resources consumed by individual users, see Exchange workload management.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "User throttling" entry in the Server health and performance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ThrottlingPolicyIdParameter	The *Identity* parameter identifies the throttling policy you want to remove. Use the name that matches the name of the policy in Active Directory.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't

			have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't

			provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ThrottlingPolicy

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-01*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ThrottlingPolicy** cmdlet to modify the settings for a user throttling policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ThrottlingPolicy -Identity <ThrottlingPolicyIdParameter> [-
AnonymousCutoffBalance <Unlimited>] [-AnonymousMaxBurst <Unlimited>] [-
AnonymousMaxConcurrency <Unlimited>] [-AnonymousRechargeRate <Unlimited>]
[-ComplianceMaxExpansionDGRecipients <Unlimited>] [-
ComplianceMaxExpansionNestedDGs <Unlimited>] [-Confirm
[<SwitchParameter>]] [-CpaCutoffBalance <Unlimited>] [-CpaMaxBurst
<Unlimited>] [-CpaMaxConcurrency <Unlimited>] [-CpaRechargeRate
<Unlimited>] [-DiscoveryMaxConcurrency <Unlimited>] [-DiscoveryMaxKeywords
<Unlimited>] [-DiscoveryMaxKeywordsPerPage <Unlimited>] [-
DiscoveryMaxMailboxes <Unlimited>] [-DiscoveryMaxPreviewSearchMailboxes
<Unlimited>] [-DiscoveryMaxRefinerResults <Unlimited>] [-
DiscoveryMaxSearchQueueDepth <Unlimited>] [-
DiscoveryMaxStatsSearchMailboxes <Unlimited>] [-
DiscoveryPreviewSearchResultsPageSize <Unlimited>] [-
DiscoverySearchTimeoutPeriod <Unlimited>] [-DomainController <Fqdn>] [-
EasCutoffBalance <Unlimited>] [-EasMaxBurst <Unlimited>] [-
EasMaxConcurrency <Unlimited>] [-EasMaxDeviceDeletesPerMonth <Unlimited>]
[-EasMaxDevices <Unlimited>] [-EasMaxInactivityForDeviceCleanup
<Unlimited>] [-EasRechargeRate <Unlimited>] [-
EncryptionRecipientCutoffBalance <Unlimited>] [-
EncryptionRecipientMaxBurst <Unlimited>] [-
EncryptionRecipientMaxConcurrency <Unlimited>] [-
EncryptionRecipientRechargeRate <Unlimited>] [-
EncryptionSenderCutoffBalance <Unlimited>] [-EncryptionSenderMaxBurst
<Unlimited>] [-EncryptionSenderMaxConcurrency <Unlimited>] [-
EncryptionSenderRechargeRate <Unlimited>] [-EwsCutoffBalance <Unlimited>]
[-EwsMaxBurst <Unlimited>] [-EwsMaxConcurrency <Unlimited>] [-
EwsMaxSubscriptions <Unlimited>] [-EwsRechargeRate <Unlimited>] [-
ExchangeMaxCmdlets <Unlimited>] [-Force <SwitchParameter>] [-
ForceSettingGlobal <SwitchParameter>] [-ForwardeeLimit <Unlimited>] [-
IgnoreDehydratedFlag <SwitchParameter>] [-ImapCutoffBalance <Unlimited>]
[-ImapMaxBurst <Unlimited>] [-ImapMaxConcurrency <Unlimited>] [-
ImapRechargeRate <Unlimited>] [-IsServiceAccount <SwitchParameter>] [-
MessageRateLimit <Unlimited>] [-Name <String>] [-
OutlookServiceCutoffBalance <Unlimited>] [-OutlookServiceMaxBurst
<Unlimited>] [-OutlookServiceMaxConcurrency <Unlimited>] [-
OutlookServiceMaxSocketConnectionsPerDevice <Unlimited>] [-
OutlookServiceMaxSocketConnectionsPerUser <Unlimited>] [-
OutlookServiceMaxSubscriptions <Unlimited>] [-OutlookServiceRechargeRate
<Unlimited>] [-OwaCutoffBalance <Unlimited>] [-OwaMaxBurst <Unlimited>] [-
OwaMaxConcurrency <Unlimited>] [-OwaRechargeRate <Unlimited>] [-
OwaVoiceCutoffBalance <Unlimited>] [-OwaVoiceMaxBurst <Unlimited>] [-
OwaVoiceMaxConcurrency <Unlimited>] [-OwaVoiceRechargeRate <Unlimited>] [-
PopCutoffBalance <Unlimited>] [-PopMaxBurst <Unlimited>] [-
PopMaxConcurrency <Unlimited>] [-PopRechargeRate <Unlimited>] [-
PowerShellCutoffBalance <Unlimited>] [-PowerShellMaxBurst <Unlimited>] [-
PowerShellMaxCmdletQueueDepth <Unlimited>] [-PowerShellMaxCmdlets
<Unlimited>] [-PowerShellMaxCmdletsTimePeriod <Unlimited>] [-
PowerShellMaxConcurrency <Unlimited>] [-PowerShellMaxDestructiveCmdlets
<Unlimited>] [-PowerShellMaxDestructiveCmdletsTimePeriod <Unlimited>] [-
PowerShellMaxOperations <Unlimited>] [-PowerShellMaxRunspaces <Unlimited>]
[-PowerShellMaxRunspacesTimePeriod <Unlimited>] [-
PowerShellMaxTenantConcurrency <Unlimited>] [-PowerShellMaxTenantRunspaces
<Unlimited>] [-PowerShellRechargeRate <Unlimited>] [-PswsMaxConcurrency
<Unlimited>] [-PswsMaxRequest <Unlimited>] [-PswsMaxRequestTimePeriod
<Unlimited>] [-PushNotificationCutoffBalance <Unlimited>] [-
PushNotificationMaxBurst <Unlimited>] [-PushNotificationMaxBurstPerDevice
<Unlimited>] [-PushNotificationMaxConcurrency <Unlimited>] [-
PushNotificationRechargeRate <Unlimited>] [-
PushNotificationRechargeRatePerDevice <Unlimited>] [-
PushNotificationSamplingPeriodPerDevice <Unlimited>] [-RcaCutoffBalance
```

```
<Unlimited>] [-RcaMaxBurst <Unlimited>] [-RcaMaxConcurrency <Unlimited>]
[-RcaRechargeRate <Unlimited>] [-RecipientRateLimit <Unlimited>] [-
ThrottlingPolicyScope <Regular | Organization | Global>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies a throttling policy so that users associated with this policy can have a maximum of four concurrent requests running in Exchange Web Services.

```
$a = Get-ThrottlingPolicy RemoteSiteUserPolicy
$a | Set-ThrottlingPolicy -EwsMaxConcurrency 4
```

EXAMPLE 2

This example modifies a throttling policy so that it restricts the number of connections for a user to three. The users associated with this policy are only able to create three Exchange Management Shell sessions. This policy also restricts to three the number of Exchange Administration Center operations or Exchange Web Services operations that can be executed at the same time. In addition, the *PowerShellMaxCmdletQueueDepth* parameter specifies that 12 operations maximum are allowed to be executed by the user.

```
Set-ThrottlingPolicy RemoteSiteUserPolicy -
PowerShellMaxConcurrency 3 -PowerShellMaxCmdletQueueDepth
12
```

EXAMPLE 3

This example modifies a throttling policy so that it restricts a user to be able to execute only 10 destructive cmdlets in 60 seconds. If the users associated with this policy exceed this number, the cmdlet pipeline execution is stopped with a throttling error message. The user needs to wait for, and then resubmit the execution of cmdlets on the open connection.

```
Set-ThrottlingPolicy <ThrottlingPolicyName> -
PowerShellMaxDestructiveCmdlets 10 -
PowerShellMaxDestructiveCmdletsTimePeriod 60
```

Detailed Description

Throttling policy settings are stored in Active Directory.

By default, there is one default user throttling policy named GlobalThrottlingPolicy with a throttling scope of Global. Microsoft Setup creates this policy when you install the Client Access server role. You shouldn't replace, re-create, or remove the existing default throttling policy. However, you can

edit any additional throttling policies with the scope of Organization or Regular if you want to change your user throttling settings. You can create polices with the scope of Organization or Regular using the **New-ThrottlingPolicy** cmdlet.

For more information about how to control the resources consumed by individual users, see Exchange workload management.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "User throttling" entry in the Server health and performance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ThrottlingPolicyIdParameter	The *Identity* parameter uniquely identifies the throttling policy that you want to modify values for. The name you use is the name of the throttling policy object in Active Directory.
AnonymousCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *AnonymousCutoffBalance* parameter specifies the resource consumption limits for an anonymous user before the user is completely blocked from performing operations on a specific component.
AnonymousMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *AnonymousMaxBurst* parameter specifies the amount of time that an anonymous user can

			consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
AnonymousMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *AnonymousMaxConcurrency* parameter specifies how many anonymous connections can be made to a user's calendar data at the same time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If anonymous users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *AnonymousMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is 1. To indicate that the number

			of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
AnonymousRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *AnonymousRechargeRate* parameter specifies the rate at which an anonymous user's budget is charged (budget grows by) during the budget time.
ComplianceMaxExpansionDGRecipients	Optional	Microsoft.Exchange.Data.Unlimited	The *ComplianceMaxExpansionDGRecipients* parameter specifies the maximum number of recipients to expand in distribution groups when a discovery search is looking for a specified recipient.
ComplianceMaxExpansionNestedDGs	Optional	Microsoft.Exchange.Data.Unlimited	The *ComplianceMaxExpansionNestedDGs* parameter specifies the maximum number of nested distribution groups to expand when a discovery search is looking for a specified recipient.
Confirm	Optional	System.Management.	The *Confirm* switch

		Automation.SwitchParameter	causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CpaCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *CpaCutoffBalance* parameter specifies the resource consumption limits for a cross-premises user before that user is completely blocked from performing operations on a specific component.
CpaMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *CpaMaxBurst* parameter specifies the amount of time that a cross-premises user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
CpaMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *CpaMaxConcurrency* parameter specifies how many concurrent

			connections a cross-premises user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *CpaMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
CpaRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *CpaRechargeRate* parameter specifies the rate at which a cross-premises user budget is charged (budget grows by) during the budget time.
DiscoveryMaxConcurr	Optional	Microsoft.Exchange.Da	The

ency		ta.Unlimited	*DiscoveryMaxConcurrency* parameter specifies the number of concurrent discovery search cmdlet executions that a user can have at the same time.
DiscoveryMaxKeywords	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxKeywords* parameter specifies the maximum number of keywords that a user can include in a discovery search. For more information, see Search-Mailbox.
DiscoveryMaxKeywordsPerPage	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxKeywordsPerPage* parameter specifies the number of keywords for which to show statistics on a single page in the Exchange Administration Center (EAC).
DiscoveryMaxMailboxes	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxMailboxes* parameter specifies the maximum number of source mailboxes that a user can include in a discovery search.

DiscoveryMaxPreviewSearchMailboxes	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxPreviewSearchMailboxes* parameter specifies the maximum number of mailboxes that a user can include in eDiscovery Search Preview.
DiscoveryMaxRefinerResults	Optional	Microsoft.Exchange.Data.Unlimited	This parameter isn't used and will be removed.
DiscoveryMaxSearchQueueDepth	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxSearchQueueDepth* parameter specifies the maximum number of concurrent discovery search threads that can be active at the same time.
DiscoveryMaxStatsSearchMailboxes	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryMaxStatsSearchMailboxes* parameter specifies the maximum number of mailboxes that a user can search in an In-Place eDiscovery search and still be able to view the keyword statistics. When the number of mailboxes configured with the *DiscoveryMaxStatsSearch*

			Mailboxes parameter is exceeded, these keyword statistics won't be available. In this case, a user must copy the search results to a discovery mailbox to view the keyword statistics for the discovery search. For more information, see In-Place eDiscovery.
DiscoveryPreviewSearchResultsPageSize	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoveryPreviewSearchResultsPageSize* parameter specifies the number of messages displayed on a single page in eDiscovery Search Preview.
DiscoverySearchTimeoutPeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *DiscoverySearchTimeoutPeriod* parameter specifies the number of minutes that a discovery search will run before it times out.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
EasCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *EasCutoffBalance* parameter specifies the resource consumption limits for a Microsoft Exchange ActiveSync user before that user is completely blocked from performing operations on a specific component.
EasMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *EasMaxBurst* parameter specifies the amount of time that an Exchange ActiveSync user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.

EasMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *EasMaxConcurrency* parameter indicates how many concurrent connections an Exchange ActiveSync user can have against a server running Microsoft Exchange Server 2013 at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *EasMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is 10. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
EasMaxDeviceDeletesPerMonth	Optional	Microsoft.Exchange.Data.Unlimited	The *EasMaxDeviceDeletesPerMonth* parameter

			specifies a limit to the number of Exchange ActiveSync partnerships that a user can delete per month. By default, each user can delete a maximum of 20 partnerships per calendar month. When the limit is reached, the partnership deletion attempt fails and an error message is displayed to the user.
EasMaxDevices	Optional	Microsoft.Exchange.Data.Unlimited	The *EasMaxDevices* parameter specifies a limit to the number of Exchange ActiveSync partnerships that a user can have at one time. By default, each user can create 10 Exchange ActiveSync partnerships with their Exchange account. After users exceed the limit, they must delete one of their existing partnerships before they can create any more new partnerships. An email error message describing the limitation is sent to the user when the

			limit is exceeded. Additionally, an event is logged in the Application log when a user exceeds the limit.
EasMaxInactivityForDeviceCleanup	Optional	Microsoft.Exchange.Data.Unlimited	The *EasMaxInactivityForDeviceCleanup* parameter specifies the length of time that a user's device partnerships will remain active. By default, there is no limit to the number of days that a user's device partnerships will remain active. Use this value if you want to minimize the amount of inactive device partnerships in your organization. To use this setting, specify a value in days since the user's last sync time to cause the device partnership to be removed.
EasRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *EasRechargeRate* parameter specifies the rate at which an Exchange ActiveSync user's budget is charged (budget grows by) during the budget time.

EncryptionRecipientCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionRecipientMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionRecipientMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionRecipientRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionSenderCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionSenderMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionSenderMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EncryptionSenderRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
EwsCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *EwsCutoffBalance* parameter specifies the resource consumption limits for an Exchange Web Services user before that user is completely blocked from performing operations on a specific component.
EwsMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *EwsMaxBurst* parameter specifies the amount of time that an

			Exchange Web Services user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
EwsMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *EwsMaxConcurrency* parameter specifies how many concurrent connections an Exchange Web Services user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *EwsMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is 10. To indicate that the number

			of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
EwsMaxSubscriptions	Optional	Microsoft.Exchange.Data.Unlimited	The *EwsMaxSubscriptions* parameter specifies the maximum number of active push and pull subscriptions that an Exchange Web Services user can have on a specific Client Access server at the same time. If a user tries to create more subscriptions than the configured maximum, the subscription fails, and an event is logged in Event Viewer.
EwsRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *EwsRechargeRate* parameter specifies the rate at which an Exchange Web Services user's budget is charged (budget grows by) during the budget time.
ExchangeMaxCmdlets	Optional	Microsoft.Exchange.Data.Unlimited	The *ExchangeMaxCmdlets* parameter specifies the number of cmdlets that can be executed within a specific time period before their execution is

			slowed down. The value specified by this parameter should be less than the value specified by the *PowerShellMaxCmdlets* parameter. The time period used for this limit is specified by the *PowerShellMaxCmdletsTimePeriod* parameter. We recommend that you set values for both parameters at the same time.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
ForceSettingGlobal	Optional	System.Management.	This parameter is reserved

		Automation.SwitchParameter	for internal Microsoft use.
ForwardeeLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *ForwardeeLimit* parameter specifies the limits for the number of recipients that can be configured in Inbox Rules when using the forward or redirect action. This parameter doesn't limit the number of messages that can be forwarded or redirected to the recipients that are configured.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ImapCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *ImapCutoffBalance* parameter specifies the resource consumption limits for an IMAP user before that user is completely blocked from performing operations on a specific component.
ImapMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *ImapMaxBurst* parameter specifies the amount of time that an IMAP user can consume an elevated amount of

			resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
ImapMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *ImapMaxConcurrency* parameter specifies how many concurrent connections an IMAP user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *ImapMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.

ImapRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *ImapRechargeRate* parameter specifies the rate at which the IMAP user's budget is charged (budget grows by) during the budget time.
IsServiceAccount	Optional	System.Management.Automation.SwitchParameter	The *IsServiceAccount* switch specifies whether you want the user accounts associated with this policy to be moderated by the per-user thresholds specified by this policy, and also by additional throttling based on the health of system resources, such as overall CPU usage. This value is set to `$false` by default. You may want to set this value to `$true` if you intend to associate this policy with user accounts that require higher throttling limits. An account that might require higher throttling limits is a service account that performs a lot of non-interactive work (for example, service accounts

			that perform IMAP mailbox migrations or nightly Windows PowerShell tasks) By setting the *IsServiceAccount* switch to `$true`, work done by these accounts is moderated by the higher user throttling settings that you configure using the user throttling policy, but is slowed if resources start getting unhealthy.
MessageRateLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *MessageRateLimit* parameter specifies the number of messages per minute that can be submitted to transport. For messages submitted through the Mailbox server role (using Microsoft Outlook, Microsoft Office Outlook Web App, Exchange ActiveSync, or Exchange Web Services), this results in the deferral of messages until the quota for the user is available. Specifically, messages appear in the Outbox or

			Drafts folder for longer periods of time when users submit messages at a rate greater than the *MessageRateLimit* parameter. For POP or IMAP clients submitting messages directly to transport using SMTP, clients receive a transient error if they submit at a rate that exceeds the *MessageRateLimit* parameter. Exchange attempts to connect and send the messages at a later time.
Name	Optional	System.String	The *Name* parameter specifies the name of the object in Active Directory. The default policy is named DefaultThrottlingPolicy<*GUID*>.
OutlookServiceCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OutlookServiceMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OutlookServiceMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.

OutlookServiceMaxSocketConnectionsPerDevice	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OutlookServiceMaxSocketConnectionsPerUser	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OutlookServiceMaxSubscriptions	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OutlookServiceRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
OwaCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaCutoffBalance* parameter specifies the resource consumption limits for an Outlook Web App user before that user is completely blocked from performing operations on a specific component.
OwaMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaMaxBurst* parameter specifies the amount of time that an Outlook Web App user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.

OwaMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaMaxConcurrency* parameter specifies how many concurrent connections an Outlook Web App user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *OwaMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is 5. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
OwaRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaRechargeRate* parameter specifies the rate at which an Outlook Web App user's budget is

			charged (budget grows by) during the budget time.
OwaVoiceCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaVoiceCutoffBalance* parameter specifies the resource consumption limits for an Outlook Web App voice user before that user is completely blocked from performing operations on a specific component.
OwaVoiceMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaVoiceMaxBurst* parameter specifies the amount of time that an Outlook Web App voice user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
OwaVoiceMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaVoiceMaxConcurrency* parameter specifies how many concurrent connections an Outlook Web App voice user can have against an Exchange

			server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *OwaVoiceMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is 5. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
OwaVoiceRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *OwaVoiceRechargeRate* parameter specifies the rate at which an Outlook Web App voice user's budget is charged (budget grows by) during the budget time.
PopCutoffBalance	Optional	Microsoft.Exchange.Da	The *PopCutoffBalance*

		ta.Unlimited	parameter specifies the resource consumption limits for a user before that user is completely blocked from performing operations on a specific component.
PopMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *PopMaxBurst* parameter specifies the amount of time that a user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
PopMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *PopMaxConcurrency* parameter specifies how many concurrent connections a POP user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection

			attempt fails. However, the existing connections remain valid. The *PopMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is 20. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
PopRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *PopRechargeRate* parameter specifies the rate at which the user budget is charged (budget grows by) during the budget time.
PowerShellCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellCutoffBalance* parameter specifies the resource consumption limits for a user before that user is completely blocked from performing operations on a specific component.
PowerShellMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxBurst* parameter specifies the amount of time that a

			user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
PowerShellMaxCmdletQueueDepth	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxCmdletQueueDepth* parameter specifies the number of operations allowed to be executed by the user. This value directly affects the behavior of the *PowerShellMaxCmdlets* and *PowerShellMaxConcurrency* parameters. For example, the *PowerShellMaxConcurrency* parameter consumes at least two operations defined by the *PowerShellMaxCmdletQueueDepth* parameter but additional operations are also consumed by per cmdlet execution. The number of operations depends on the cmdlets that are executed. We

			recommend that the value for the *PowerShellMaxCmdletQueueDepth* parameter be at least three times larger than the value of the *PowerShellMaxConcurrency* parameter. This parameter won't affect Exchange Administration Center operations or Exchange Web Services operations.
PowerShellMaxCmdlets	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxCmdlets* parameter specifies the number of cmdlets that can be executed within a specific time period before their execution is stopped. The value specified by this parameter should be more than the value specified by the *ExchangeMaxCmdlets* parameter. The time period used for this limit is specified by the *PowerShellMaxCmdletsTimePeriod* parameter. Both values should be set at the

			same time.
PowerShellMaxCmdletsTimePeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxCmdletsTimePeriod* parameter specifies the time period, in seconds, that the throttling policy uses to determine whether the number of cmdlets being executed exceeds the limits specified by the *PowerShellMaxCmdlets* and *ExchangeMaxCmdlets* parameters.
PowerShellMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxConcurrency* parameter specifies different information depending on context: • In the context of Remote PowerShell, the *PowerShellMaxConcurrency* parameter specifies the maximum number of Remote PowerShell sessions that a Remote PowerShell user can have open at the same time. • In the context of Exchange Web Services, the

			PowerShellMaxConcurrency parameter specifies the number of concurrent cmdlet executions that a user can have at the same time. This parameter value doesn't necessarily correlate to the number of browsers opened by the user.
PowerShellMaxDestructiveCmdlets	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxDestructiveCmdlets* parameter specifies the number of destructive cmdlets that can be executed within a specific time period before their execution is stopped. Destructive cmdlets are cmdlets that can make significant changes to user data and configuration settings in your Exchange organization. Throttling these cmdlets may help prevent accidental data loss. The following cmdlets are designated as destructive: • **Disable-Mailbox**

			• **Move-ActiveMailboxDatabase** • **Remove-AcceptedDomain** • **Remove-Mailbox** • **Remove-MailUser** • **Remove-Organization** • **Remove-SecondaryDomain** • **Remove-SyncMailbox** • **Remove-SyncMailUser** • **Set-Mailbox** • **Set-MailUser** • **Set-SyncMailbox** • **Set-SyncMailUser** • **Start-OrganizationUpgrade** • **Update-MailboxDatabaseCopy** The time period used for this limit is specified by the *PowerShellMaxDestructiveCmdletsTimePeriod* parameter. Both values should be set at the same time. This feature isn't enabled by default. For more information, see the example in this topic.

PowerShellMaxDestructiveCmdletsTimePeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxDestructiveCmdletsTimePeriod* parameter specifies the time period, in seconds, that the throttling policy uses to determine how many destructive cmdlets can be run. You set a value for this parameter when set the *PowerShellMaxDestructiveCmdlets* parameter. Both values should be set at the same time. For more information, see the description for the *PowerShellMaxDestructiveCmdlets* parameter.
PowerShellMaxOperations	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxOperations* parameter specifies the protocol-level operations used to send and receive data. If the execution of a cmdlet results in a significant number of operations (for example, if there is a lot of input/output occurring), throttling may occur. The default setting is

			Unlimited.
PowerShellMaxRunspaces	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxRunspaces* parameter specifies the number of concurrent Windows PowerShell sessions that a user is allowed to have. The default setting is Unlimited.
PowerShellMaxRunspacesTimePeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxRunspacesTimePeriod* parameter specifies the time period, in seconds, that the throttling policy uses to determine how many Windows PowerShell sessions can be run. You set this value when you set the *PowerShellMaxRunspaces* parameter.
PowerShellMaxTenantConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxTenantConcurrency* parameter limits the number of concurrent Windows PowerShell connections per tenant organization. By default, the limit for concurrent Windows PowerShell

			connections per tenant organization is set to 9. If users in a tenant organization try to make more concurrent requests than the limit set by the *PowerShellMaxTenantConcurrency* parameter, the new connection attempt fails. However, the existing connections remain valid. This limit is enforced even if a single user hasn't exceeded the per-user limit set by the *PowerShellMaxConcurrency* parameter. The *PowerShellMaxTenantConcurrency* parameter has a valid range from 0 through 100 inclusive. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`. **Note:** This property can only be set for the default throttling policy.
PowerShellMaxTenantRunspaces	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellMaxTenantRunspaces* parameter

			specifies the number of concurrent Windows PowerShell sessions that a tenant is allowed to have.
PowerShellRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *PowerShellRechargeRate* parameter specifies the rate at which the user budget is charged (budget grows by) during the budget time.
PswsMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *PswsMaxConcurrency* parameter specifies how many concurrent connections a Windows PowerShell Web Services user can have against an Exchange server at one time. A connection is held from the moment a request is received until a response is sent in its entirety to the requestor. If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. This parameter has a default value of 18. To

			indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
PswsMaxRequest	Optional	Microsoft.Exchange.Data.Unlimited	The *PswsMaxRequest* parameter specifies how many requests a Windows PowerShell Web Services user can have against an Exchange server at one time. The default setting is Unlimited.
PswsMaxRequestTimePeriod	Optional	Microsoft.Exchange.Data.Unlimited	The *PswsMaxRequestTimePeriod* parameter specifies the period of time, in seconds, that the throttling policy uses to determine how many requests can be run. You set this value when you set the *PswsMaxRequest* parameter. The default setting is Unlimited.
PushNotificationCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
PushNotificationMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
PushNotificationMaxB	Optional	Microsoft.Exchange.Da	This parameter is reserved

urstPerDevice		ta.Unlimited	for internal Microsoft use.
PushNotificationMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
PushNotificationRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
PushNotificationRechargeRatePerDevice	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
PushNotificationSamplingPeriodPerDevice	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is reserved for internal Microsoft use.
RcaCutoffBalance	Optional	Microsoft.Exchange.Data.Unlimited	The *RcaCutoffBalance* parameter specifies the resource consumption limits for a user before that user is completely blocked from performing operations on a specific component.
RcaMaxBurst	Optional	Microsoft.Exchange.Data.Unlimited	The *RcaMaxBurst* parameter specifies the amount of time that a user can consume an elevated amount of resources before being throttled. This is measured in milliseconds. This value is set separately for each component.
RcaMaxConcurrency	Optional	Microsoft.Exchange.Data.Unlimited	The *RcaMaxConcurrency* parameter specifies how

			many concurrent connections an RPC Client Access user can have against a server running Exchange 2013 at one time. A connection is held from the moment a request is received until the connection is closed or the connection is otherwise disconnected (for example, if the user goes offline). If users attempt to make more concurrent requests than their policy allows, the new connection attempt fails. However, the existing connections remain valid. The *RcaMaxConcurrency* parameter has a valid range from 0 through 2147483647 inclusive. The default value is 20. To indicate that the number of concurrent connections should be unthrottled (no limit), this value should be set to `$null`.
RcaRechargeRate	Optional	Microsoft.Exchange.Data.Unlimited	The *RcaRechargeRate* parameter specifies the rate at which the user

			budget is charged (budget grows by) during the budget time.
RecipientRateLimit	Optional	Microsoft.Exchange.Data.Unlimited	The *RecipientRateLimit* parameter specifies the limits on the number of recipients that a user can address in a 24-hour period.
ThrottlingPolicyScope	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ThrottlingPolicyScopeType	The *ThrottlingPolicyScope* parameter specifies the scope of the throttling policy. You can use the following values: • `Regular` This scope specifies a custom policy that applies to specific users. • `Organization` This scope specifies a custom policy that applies to all users in your organization. • `Global` This scope is reserved for the default throttling policy. For more information about throttling policy scopes, see Exchange workload management.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-ThrottlingPolicyAssociation

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-ThrottlingPolicyAssociation** cmdlet to view the relationship between objects and their associated throttling policies. The object can be a user with a mailbox, a user without a mailbox, a contact, or a computer account.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-ThrottlingPolicyAssociation [-Identity
<ThrottlingPolicyAssociationIdParameter>] <COMMON PARAMETERS>

Get-ThrottlingPolicyAssociation [-Anr <String>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-Credential <PSCredential>] [-DomainController <Fqdn>]
```

```
[-Filter <String>] [-IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>] [-SortBy <String>] [-
ThrottlingPolicy <ThrottlingPolicyIdParameter>]
```

Examples

EXAMPLE 1

This example returns a list of all the mailboxes in your organization in the Users OU.

```
Get-ThrottlingPolicyAssociation -OrganizationalUnit Users
```

EXAMPLE 2

This example returns all the mailboxes that resolve from the ambiguous name resolution search on the string "Chr" that are in the domain DC01. This example returns mailboxes for users such as Chris Ashton, Christian Hess, and Christa Geller.

```
Get-ThrottlingPolicyAssociation -Anr Chr -DomainController
DC01
```

Detailed Description

For more information about how to control the resources consumed by individual users, see Exchange workload management.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "User throttling" entry in the Server health and performance permissions topic.

Parameters

Parameter	Required	Type	Description
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an

			attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the

			local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ThrottlingPolicyAssociationIdParameter	The *Identity* parameter specifies the object to query to see its throttling policy associations. The object can be a user with a mailbox, a user without a mailbox, a contact, or a computer account. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example:

			CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default

			scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the distinguished name (DN) for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter specifies the organization that the object specified by the *Identity* parameter is in.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParam	The *OrganizationalUnit* parameter specifies an organizational unit (OU)

		eter	and is used to limit the results. If you use this parameter, you only get mailboxes in the container that you specify. You can use either the OU or the domain name. If you use the OU, you must specify the canonical name of the OU.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest and don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Exchange.

ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all mailboxes that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. You can sort by only one attribute at a time. You can sort by the following attributes: • Alias • Display name • Name The results are sorted in ascending order.
ThrottlingPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.ThrottlingPolicyIdParameter	The *ThrottlingPolicy* parameter specifies the identity of the throttling policy for which you want to view throttling policy associations.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-ThrottlingPolicyAssociation

Exchange Management Shell > Exchange 2013 cmdlets > Server health, monitoring, and performance cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-11*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-ThrottlingPolicyAssociation** cmdlet to associate a throttling policy with a specific object. The object can be a user with a mailbox, a user without a mailbox, a contact, or a computer account.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-ThrottlingPolicyAssociation -Identity
<ThrottlingPolicyAssociationIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-IgnoreDefaultScope <SwitchParameter>] [-
ThrottlingPolicy <ThrottlingPolicyIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example associates a user with a user name of tonysmith to the throttling policy ITStaffPolicy that has higher limits.

```
Set-ThrottlingPolicyAssociation -Identity tonysmith -
ThrottlingPolicy ITStaffPolicy
```

You don't need to use the **Set-ThrottlingPolicyAssociation** cmdlet to associate a user with a policy. The following commands show another way to associate tonysmith to the throttling policy ITStaffPolicy.

```
$b = Get-ThrottlingPolicy ITStaffPolicy

Set-Mailbox -Identity tonysmith -ThrottlingPolicy $b
```

Detailed Description

The **Set-ThrottlingPolicyAssociation** cmdlet defines quota limits for specific objects. For example, if you notice that a user or other object is using excessive bandwidth, you can associate that object with a throttling policy that's more restrictive.

Note:
In data center deployments, the object referred to by the *Identity* and *ThrottlingPolicy* parameters must be in the same tenant.

For more information about how to control the resources consumed by individual users, see Exchange workload management.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "User throttling" entry in the Server health and performance permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ThrottlingPolicyAssociationIdParameter	The *Identity* parameter specifies the object to which you want to associate a throttling policy. The object can be a user with a mailbox, a user without a mailbox, a contact, or a computer account.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You

			don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't

			currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • If you use the *Identity* parameter, you must use enter the value in the distinguished name (DN) format.
ThrottlingPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.ThrottlingPolicyIdParameter	The *ThrottlingPolicy* parameter specifies the throttling policy that you want to be associated with the object specified by the *Identity* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Sharing and collaboration cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-28*

Availability cmdlets

Add-AvailabilityAddressSpace

Get-AvailabilityAddressSpace

Remove-AvailabilityAddressSpace

Get-AvailabilityConfig

Set-AvailabilityConfig

Sharing cmdlets

Get-OrganizationRelationship

New-OrganizationRelationship

Remove-OrganizationRelationship

Set-OrganizationRelationship

Test-OrganizationRelationship

Get-SharingPolicy

New-SharingPolicy

Remove-SharingPolicy

Set-SharingPolicy

Public folder cmdlets

Disable-MailPublicFolder

Enable-MailPublicFolder

Get-MailPublicFolder

Set-MailPublicFolder

Get-PublicFolder

New-PublicFolder

Remove-PublicFolder

Set-PublicFolder

Add-PublicFolderClientPermission

Get-PublicFolderClientPermission

Remove-PublicFolderClientPermission

Get-PublicFolderDatabase

Get-PublicFolderItemStatistics

Update-PublicFolderMailbox

Get-PublicFolderMailboxDiagnostics

Get-PublicFolderStatistics

Site mailbox cmdlets

Get-SiteMailbox

Set-SiteMailbox

Test-SiteMailbox

Update-SiteMailbox

Get-SiteMailboxDiagnostics

Get-SiteMailboxProvisioningPolicy

New-SiteMailboxProvisioningPolicy

Remove-SiteMailboxProvisioningPolicy

Set-SiteMailboxProvisioningPolicy

Add-AvailabilityAddressSpace

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Add-AvailabilityAddressSpace** cmdlet to define the access method and associated credentials used to exchange free/busy data across forests.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-AvailabilityAddressSpace -AccessMethod <PerUserFB | OrgWideFB |
PublicFolder | InternalProxy | OrgWideFBBasic> -ForestName <String> [-
Confirm [<SwitchParameter>]] [-Credentials <PSCredential>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>] [-
ProxyUrl <Uri>] [-TargetAutodiscoverEpr <Uri>] [-UseServiceAccount <$true
| $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example is useful with an untrusted cross-forest Availability service, or if detailed cross-forest free/busy service isn't desired. When you're prompted by the command, type a user name and password. For an untrusted cross-forest configuration, make sure that the user doesn't have a mailbox.

```
Add-AvailabilityAddressSpace -ForestName
<example.contoso.com> -AccessMethod OrgWideFB -Credentials
<ExampleCredential>
```

EXAMPLE 2

This example is useful with a trusted cross-forest Availability service. The contoso.com forest trusts the current forest, and the specified account connects to the contoso.com forest. The specified account must be an existing account in the contoso.com forest.

```
Add-AvailabilityAddressSpace -ForestName
<example.contoso.com> -AccessMethod PerUserFB -Credentials
<ExampleCredential>
```

EXAMPLE 3

This example is useful with a trusted cross-forest Availability service. The contoso.com forest trusts the current forest and uses the service account (typically the local system account or the computer account) to connect to the contoso.com forest. Because the service is trusted, there is no issue with authorization when the current forest tries to retrieve free/busy information from contoso.com.

```
Add-AvailabilityAddressSpace -ForestName
<example.contoso.com> -AccessMethod PerUserFB -
UseServiceAccount $true
```

EXAMPLE 4

This example is useful for interoperability between Exchange 2013 and versions of Exchange earlier than Exchange 2007, for example Exchange 2003.

```
Add-AvailabilityAddressSpace -ForestName
<example.contoso.com> -AccessMethod PublicFolder
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Availability service address space settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
AccessMethod	Required	Microsoft.Exchange.Data.Directory.SystemConfiguration.AvailabilityAccessMethod	The *AccessMethod* parameter specifies the availability access method and can have the following values: • `PerUserFB` This value causes the command to

			access the free/busy data in the defined per-user free/busy proxy account or group, or in the All Exchange Servers group. The `PerUserFB` value requires trust between the two forests. You must use either the *UseServiceAccount* parameter or *Credentials* parameter. • `OrgWideFB` This value causes the command to access the free/busy data in the per-user free/busy proxy account or group in the target forest. You must use either the *UseServiceAccount* parameter or *Credentials* parameter. • `OrgWideFBBasic` This value is reserved for internal Microsoft use. • `InternalProxy` This value is used to proxy a request to the Client Access server in the site with the latest version of Exchange. • `PublicFolder` This value causes the command to access free/busy data on servers running Microsoft Exchange Server 2003. The Exchange Inter-Organization Replication tool must be running between the two forests.

ForestName	Required	System.String	The *ForestName* parameter specifies the SMTP domain name of the target forest for users whose free/busy data must be retrieved. If your users are distributed among multiple SMTP domains in the target forest, run the **Add-AvailabilityAddressSpace** command once for each SMTP domain.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Credentials	Optional	System.Management.Automation.PSCredential	The *Credentials* parameter specifies the credentials for an account that has permission to access the Availability services in the target forest. This parameter requires the creation and passing of a credential object. This

			credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ProxyUrl	Optional	System.Uri	This parameter is available only in on-premises Exchange 2013. The *ProxyUrl* parameter specifies whether to direct a Microsoft Exchange Server 2007 Client Access server to proxy its free/busy requests through an Exchange Server 2013 Client Access server when requesting federated free/

			busy data for a user in another organization. Before you can configure this setting, you must create the proper trust relationships and sharing relationships. For more information, see New-FederationTrust.
TargetAutodiscoverEpr	Optional	System.Uri	The *TargetAutodiscoverEpr* parameter specifies the Autodiscover URL of Exchange Web Services for the external organization, for example, https://contoso.com/autodiscover/autodiscover.svc/wssecurity. Exchange uses Autodiscover to automatically detect the correct server endpoint for external requests.
UseServiceAccount	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *UseServiceAccount* parameter, when assigned a value of `$true`, uses the local Availability service

			account for authorization.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AvailabilityAddressSpace

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-AvailabilityAddressSpace** cmdlet to return details about how your Exchange organization is configured in regard to the exchange of free/busy information.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-AvailabilityAddressSpace [-Identity
<AvailabilityAddressSpaceIdParameter>] [-DomainController <Fqdn>] [-
Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example returns details about how your Exchange organization is configured in regard to the exchange of free/busy information across your organization.

```
Get-AvailabilityAddressSpace
```

EXAMPLE 2

This example returns details about how your Exchange organization is configured in regard to the exchange of free/busy information for the specific domain in the organization Contoso.com.

```
Get-AvailabilityAddressSpace -Identity Contoso.com
```

EXAMPLE 3

This example returns details about how your Exchange organization is configured in regard to the exchange of free/busy information for a specific domain in your organization. This example uses the FQDN of the domain controller.

```
Get-AvailabilityAddressSpace -DomainController <FQDN of the
domain controller>
```

Detailed Description

The **Get-AvailabilityAddressSpace** cmdlet returns details about the access method and associated credentials used to exchange free/busy information across forests. To effectively use the command, run it on a computer that has the Client Access server role installed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Availability service address space settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.AvailabilityAddressSpaceIdParameter	The *Identity* parameter specifies the availability address space entry to be retrieved.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-AvailabilityAddressSpace

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-AvailabilityAddressSpace** cmdlet to remove a previously defined availability address space and the associated credentials used in cross-forest requests for free/busy information.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-AvailabilityAddressSpace -Identity
<AvailabilityAddressSpaceIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the availability address space entry Contoso.com.

```
Remove-AvailabilityAddressSpace -Identity Contoso.com
```

EXAMPLE 2

This example removes the availability address space entry Contoso.com and then requires confirmation before completing the removal.

```
Remove-AvailabilityAddressSpace -Identity Contoso.com -
Confirm
```

Detailed Description

To effectively use the **Remove-AvailabilityAddressSpace** cmdlet, run it on a computer that has the Client Access server role installed.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Availability service address space settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.AvailabilityAddressSpaceIdParameter	The *Identity* parameter specifies the availability address space entry to be removed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.A	The *WhatIf* switch

		utomation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-AvailabilityConfig

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-AvailabilityConfig** cmdlet to retrieve the accounts that are trusted in the cross-forest exchange of free/busy information.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-AvailabilityConfig [-Identity <OrganizationIdParameter>] [-
DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example retrieves the accounts that are trusted in the cross-forest exchange of free/busy information.

```
Get-AvailabilityConfig
```

EXAMPLE 2

This example retrieves the accounts that are trusted in the cross-forest exchange of free/busy information. This example is scoped to return only the results of the specified *Identity* parameter.

```
Get-AvailabilityConfig -Identity <AvailabilityConfig Value>
```

Detailed Description

The **Get-AvailabilityConfig** cmdlet lists the accounts that have permissions to issue proxy availability service requests on an organizational or per-user basis.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Availability service configuration settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter specifies the availability configuration to be retrieved.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-AvailabilityConfig

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-AvailabilityConfig** cmdlet to set the access level for free/busy information.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-AvailabilityConfig [-Identity <OrganizationIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-OrgWideAccount
<SecurityPrincipalIdParameter>] [-PerUserAccount
<SecurityPrincipalIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example is useful with a trusted cross-forest Availability service. If the remote forest is trusted, and a per-user free/busy proxy account or group in the remote forest is configured to use the service account, the configuration is added to the current forest to authorize the Microsoft ActiveSync request from the remote forest.

```
Set-AvailabilityConfig -PerUserAccount <domain name of servers group in remote forest>
```

EXAMPLE 2

This example is useful if the remote forest isn't trusted. Because this account is used for a cross-forest free/busy proxy account or group, minimize security vulnerabilities by using the credentials of a user who doesn't have an Exchange mailbox. When you're prompted, type the user name and password.

```
Set-AvailabilityConfig -OrgWideAccount <ExampleCredentials>
```

Detailed Description

The **Set-AvailabilityConfig** cmdlet defines two accounts or security groups: a per-user free/busy proxy account or group, and an organization-wide free/busy proxy account or group. These accounts and groups are trusted by all availability services in the current organization for availability proxy requests.

For cross-forest availability services to retrieve free/busy information in the current forest, they must be using one of the specified accounts, belong to one of the specified security groups, or have a user name and password for one of the specified accounts or security groups.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Availability service configuration settings" entry in the Clients and mobile devices permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Identity* parameter specifies the OrganizationID.
OrgWideAccount	Optional	Microsoft.Exchange.Configuration.Tasks.SecurityPrincipalIdParameter	The *OrgWideAccount* parameter specifies an account or security group that has permission to issue proxy Availability service requests on an organization-wide basis.
PerUserAccount	Optional	Microsoft.Exchange.Co	This parameter is

		nfiguration.Tasks.SecurityPrincipalIdParameter	available only in on-premises Exchange 2013. The *PerUserAccount* parameter specifies an account or security group that has permission to issue proxy Availability service requests on a per-user basis.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-MailPublicFolder

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-MailPublicFolder** cmdlet to mail-disable a public folder.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-MailPublicFolder -Identity <MailPublicFolderIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example mail-disables the public folder Help Desk.

```
Disable-MailPublicFolder -Identity "\Help Desk"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders, mail-enabled" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailPublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also include the path

			using the format *TopLevelPublicFolder\PublicFolder*. You can omit the parameter label *Identity* so that only the public folder name or GUID is supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-MailPublicFolder

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-MailPublicFolder** cmdlet to mail-enable public folders.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-MailPublicFolder -Identity <PublicFolderIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-
HiddenFromAddressListsEnabled <$true | $false>] [-OverrideRecipientQuotas
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example mail-enables the top-level public folder My Public Folder.

```
Enable-MailPublicFolder "\My Public Folder"
```

EXAMPLE 2

This example mail-enables the public folder Reports that's in the parent folder Marketing.

```
Enable-MailPublicFolder "\Marketing\Reports"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders, mail enabled" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also include the path using the format *TopLevelPublicFolder\PublicFolder*. You can omit the parameter label so that only the public folder

			name or GUID is supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
HiddenFromAddressListsEnabled	Optional	System.Boolean	The *HiddenFromAddressListsEnabled* parameter specifies whether the folder is hidden from address lists. Valid

			values are `$true` and `$false`. The default value is `$false`.
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailPublicFolder

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailPublicFolder** cmdlet to retrieve mail-related information about mail-enabled public folders. If you want information about the basic (not mail-related) settings of mail-enabled public folders, use the **Get-PublicFolder** cmdlet instead.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailPublicFolder [-Anr <String>] <COMMON PARAMETERS>


Get-MailPublicFolder [-Identity <MailPublicFolderIdParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-Filter <String>]
[-IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-ReadFromDomainController <SwitchParameter>]
[-ResultSize <Unlimited>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example returns the information for up to 100 mail-enabled public folders. In this example, the output of the **Get-MailPublicFolder** command is piped to the **Format-List** command so that all the available information is displayed in the result.

```
Get-MailPublicFolder -ResultSize 100 | Format-List
```

EXAMPLE 2

This example returns information for the mail-enabled public folder Reports that resides in the Marketing top-level public folder.

```
Get-MailPublicFolder -Identity \Marketing\Reports
```

EXAMPLE 3

This example returns all mail-enabled public folders that begin with the word Marketing by using the *Anr* parameter.

```
Get-MailPublicFolder -Anr Marketing*
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders, mail-enabled" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password to use to access Active Directory.

			This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailPublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also

			include the path using the format *TopLevelPublicFolder* *PublicFolder*. You can omit the parameter label so that only the public folder name or GUID is supplied.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can't use the *Anr*

			and *Identity* parameters together. • You can't use the *Credential* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest, and if you don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Microsoft Exchange.

ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. Sorting is done one attribute at a time. The results are sorted in ascending order.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailPublicFolder

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailPublicFolder** cmdlet to configure the mail-related settings of mail-enabled public folders. If you want to configure basic settings that aren't mail related, use the **Set-PublicFolder** cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailPublicFolder -Identity <MailPublicFolderIdParameter> [-
AcceptMessagesOnlyFrom <MultiValuedProperty>] [-
AcceptMessagesOnlyFromDLMembers <MultiValuedProperty>] [-
AcceptMessagesOnlyFromSendersOrMembers <MultiValuedProperty>] [-Alias
<String>] [-ArbitrationMailbox <MailboxIdParameter>] [-
BypassModerationFromSendersOrMembers <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-Contacts <RecipientIdParameter[]>] [-CreateDTMFMap
<$true | $false>] [-CustomAttribute1 <String>] [-CustomAttribute10
<String>] [-CustomAttribute11 <String>] [-CustomAttribute12 <String>] [-
CustomAttribute13 <String>] [-CustomAttribute14 <String>] [-
CustomAttribute15 <String>] [-CustomAttribute2 <String>] [-
CustomAttribute3 <String>] [-CustomAttribute4 <String>] [-CustomAttribute5
<String>] [-CustomAttribute6 <String>] [-CustomAttribute7 <String>] [-
CustomAttribute8 <String>] [-CustomAttribute9 <String>] [-
DeliverToMailboxAndForward <$true | $false>] [-DisplayName <String>] [-
DomainController <Fqdn>] [-EmailAddresses <ProxyAddressCollection>] [-
EmailAddressPolicyEnabled <$true | $false>] [-ExtensionCustomAttribute1
<MultiValuedProperty>] [-ExtensionCustomAttribute2 <MultiValuedProperty>]
[-ExtensionCustomAttribute3 <MultiValuedProperty>] [-
ExtensionCustomAttribute4 <MultiValuedProperty>] [-
ExtensionCustomAttribute5 <MultiValuedProperty>] [-ExternalEmailAddress
<ProxyAddress>] [-ForwardingAddress <RecipientIdParameter>] [-
GrantSendOnBehalfTo <MultiValuedProperty>] [-HiddenFromAddressListsEnabled
<$true | $false>] [-IgnoreDefaultScope <SwitchParameter>] [-MailTip
<String>] [-MailTipTranslations <MultiValuedProperty>] [-MaxReceiveSize
<Unlimited>] [-MaxSendSize <Unlimited>] [-ModeratedBy
<MultiValuedProperty>] [-ModerationEnabled <$true | $false>] [-Name
<String>] [-PhoneticDisplayName <String>] [-PrimarySmtpAddress
<SmtpAddress>] [-RejectMessagesFrom <MultiValuedProperty>] [-
RejectMessagesFromDLMembers <MultiValuedProperty>] [-
RejectMessagesFromSendersOrMembers <MultiValuedProperty>] [-
RequireSenderAuthenticationEnabled <$true | $false>] [-
SendModerationNotifications <Never | Internal | Always>] [-
SimpleDisplayName <String>] [-UMDtmfMap <MultiValuedProperty>] [-WhatIf
[<SwitchParameter>]] [-WindowsEmailAddress <SmtpAddress>]
```

Examples

EXAMPLE 1

This example sets the primary SMTP address of the mail-enabled public folder MyPublicFolder@contoso.com to MyPublicFolder@fabrikam.com.

```
Set-MailPublicFolder -Identity MyPublicFolder@contoso.com -
PrimarySmtpAddress MyPublicFolder@fabrikam.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders, mail-enabled" section in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailPublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also include the path using the format *TopLevelPublicFolder**PublicFolder*. You can omit the parameter label so that only the public folder name or GUID is supplied.
AcceptMessagesOnlyFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFrom* parameter specifies the mailbox users, mail users, and mail contacts that can send email messages to this mail-enabled public folder. You can also specify Microsoft Exchange as a valid recipient for this parameter. If you configure a mail-enabled public folder to accept messages only from the Microsoft Exchange recipient, it only receives system-generated messages.

			The *AcceptMessagesOnlyFrom* parameter can take any of the following values for valid senders: • Distinguished name (DN) • Canonical name • GUID • Name • Display name • Alias • Exchange DN • Primary SMTP email address By default, this parameter is blank. This default value enables the mail-enabled public folder to accept messages from all senders.
AcceptMessagesOnlyFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromDLMembers* parameter specifies the distribution groups whose members are allowed to send email messages to this mail-enabled public folder. You can use any of the following values for the allowed distribution

			groups: • DN • Canonical name • GUID • Name • Display name • Legacy Exchange DN • Primary SMTP email address By default, this parameter is blank. This default value enables the mail-enabled public folder to accept messages from all senders.
AcceptMessagesOnlyFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromSendersOrMembers* parameter specifies the mailbox users, mail users, mail contacts, and distribution groups who are allowed to send email messages to this mail-enabled public folder. You can use any of the following values for the allowed distribution groups: • DN • Canonical name • GUID

			• Name • Display name • Legacy Exchange DN • Primary SMTP email address By default, this parameter is blank. This default value enables the mail-enabled public folder to accept messages from all senders.
Alias	Optional	System.String	The *Alias* parameter specifies the alias (mail nickname) of the public folder. If not specified, the *Alias* parameter value is stamped with the name of the public folder. The string must comply with RFC 2821 requirements for valid "local part" SMTP addresses.
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *ArbitrationMailbox* parameter specifies the mailbox used to manage the moderation process.
BypassModerationFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassModerationFromSendersOrMembers* parameter specifies the

			recipients whose messages bypass moderation when sending to this mail-enabled public folder. You can use any of the following values: • DN • Canonical name • GUID • Name • Display name • Legacy Exchange DN • Primary SMTP email address By default, this parameter is blank. This default value enables the mail-enabled public folder to accept messages from all senders. Senders designated as moderators for this mail-enabled public folder are allowed to send messages to this mail-enabled public folder.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Contacts	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *Contacts* parameter specifies the contacts for the public folder. *Contacts* are persons about whom you can save several types of information, such as addresses, telephone numbers, and web page URLs.
CreateDTMFMap	Optional	System.Boolean	The *CreateDTMFMap* parameter specifies that a dual tone multi-frequency (DTMF) map be created for the user.
CustomAttribute1	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute10	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to

			store additional information.
CustomAttribute11	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute12	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute13	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute14	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.

CustomAttribute15	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute2	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute3	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute4	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute5	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify

			custom attributes. You can use these attributes to store additional information.
CustomAttribute6	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute7	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute8	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute9	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional

			information.
DeliverToMailboxAndForward	Optional	System.Boolean	The *DeliverToMailboxAndForward* parameter specifies whether email messages are sent to a forwarding address.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the display name of the **Public Folder Proxy** object.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EmailAddresses	Optional	Microsoft.Exchange.Data.ProxyAddressCollection	The *EmailAddresses* parameter specifies proxy addresses (for example, user@contoso.com).
EmailAddressPolicyEnabled	Optional	System.Boolean	The *EmailAddressPolicyEnabled* parameter specifies

			whether to enable an email address policy that's applied to the folder.
ExtensionCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list.

			Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.

ExtensionCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to

			1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExternalEmailAddress	Optional	Microsoft.Exchange.Data.ProxyAddress	The *ExternalEmailAddress* parameter specifies an email address outside the organization.
ForwardingAddress	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *ForwardingAddress* parameter specifies the forwarding address of the folder.
GrantSendOnBehalfTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *GrantSendOnBehalfTo* parameter specifies the DN of other mailboxes that can send on behalf of this folder.
HiddenFromAddressListsEnabled	Optional	System.Boolean	The *HiddenFromAddressListsEnabled* parameter specifies whether the mailbox is viewable from address lists.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchPar	The *IgnoreDefaultScope* parameter instructs the

		ameter	command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically.
MailTip	Optional	System.String	The *MailTip* parameter specifies the message that's displayed to senders when they start drafting an email message to this recipient. The *MailTip* parameter message must be less than or equal to 250 characters.
MailTipTranslations	Optional	Microsoft.Exchange.Da	The *MailTipTranslations*

		ta.MultiValuedProperty	parameter specifies additional languages when you want to provide the *MailTip* parameter information for this recipient in multiple languages. For each language, you must provide the locale, followed by a colon and the specific *MailTip* parameter value in that language. Each *MailTip* parameter value must be less than or equal to 250 characters. Multiple languages can be separated by commas.
MaxReceiveSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxReceiveSize* parameter specifies the maximum size of email messages that can be received by the folder. Valid values are from 1 kilobyte (KB) to 2,097,151 KB. If a value isn't specified for this parameter, no size limit is imposed.
MaxSendSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxSendSize* parameter specifies the maximum size of email

			messages that can be sent. Valid values are from 1 KB to 2,097,151 KB. If a value isn't specified for this parameter, no size limit is imposed.
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users who are responsible for moderating the messages sent to the mail-enabled public folder. To designate more than one user, separate the users with commas. This parameter is required if you set the *ModerationEnabled* parameter to `$true`. If you leave this parameter blank and there is a user who is already specified as the manager of this distribution group, the *ModeratedBy* parameter is automatically set by the *ManagedBy* parameter of the mail-enabled public folder. Otherwise, an error is returned.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies

			whether to enable moderation for the mail-enabled public folder. To enable moderation, set this parameter to `$true`. To disable moderation, set this parameter to `$false`. The default value is `$false`.
Name	Optional	System.String	The *Name* parameter specifies the name of the public folder. Use this parameter if you want to change the name of the public folder.
PhoneticDisplayName	Optional	System.String	The *PhoneticDisplayName* parameter specifies a phonetic pronunciation of the *DisplayName* parameter. The maximum length of this parameter value is 255 characters.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the primary SMTP address for the folder.
RejectMessagesFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFrom* parameter specifies the recipients from which to reject messages. You can

			use any of the following values to specify the recipients: • DN • Canonical name • GUID • Name • Display name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank. This default value enables the mail-enabled public folder to accept messages from all senders.
RejectMessagesFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromDLMembers* parameter specifies the distribution list members from which to reject messages. You can use any of the following values to specify the recipients: • DN • Canonical name • GUID • Name • Display name • LegacyExchangeDN

			• Primary SMTP email address By default, this parameter is blank. This default value enables the mail-enabled public folder to accept messages from all senders.
RejectMessagesFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromSendersOrMembers* parameter specifies the mailbox users, mail users, mail contacts, and distribution groups whose members aren't allowed to send email messages to this mail-enabled public folder. You can use any of the following values to specify the recipients: • DN • Canonical name • GUID • Name • Display name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank. This default value enables the mail-enabled

			public folder to accept messages from all senders.
RequireSenderAuthenticationEnabled	Optional	System.Boolean	The *RequireSenderAuthenticationEnabled* parameter specifies whether senders must be authenticated.
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when they send a message to the moderated distribution group. You can specify one of the following values: • `Always` • `Internal` • `Never` Set this parameter to `Always` if you want notifications to be sent to all senders. Set this parameter to `Internal` if you want notifications to be sent only to the senders who are internal to your organization. Set this parameter to

			Never to disable all status notifications. **Note:** The sender is always notified if the message is rejected by the moderators, regardless of the value of this parameter. The default value is Never.
SimpleDisplayName	Optional	System.String	The *SimpleDisplayName* parameter is used to display an alternative description of the object when only a limited set of characters is permitted. This limited set of characters consists of ASCII characters 26 through 126, inclusively.
UMDtmfMap	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UMDtmfMap* parameter specifies if you want to create a user-defined DTMF map for the UM-enabled user.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply

			any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsEmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *WindowsEmailAddress* parameter specifies an email address in the format *EmailAddress*@contoso.com.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-OrganizationRelationship

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-OrganizationRelationship** cmdlet to retrieve settings for an organization relationship that has been created for federated sharing with other federated Exchange organizations or for hybrid deployments with Exchange Online.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-OrganizationRelationship [-Identity
<OrganizationRelationshipIdParameter>] [-DomainController <Fqdn>] [-
Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves the organization relationship settings for Contoso using the *Identity* parameter.

```
Get-OrganizationRelationship -Identity Contoso
```

EXAMPLE 2

This example retrieves the organization relationship settings by using the FQDN of the domain controller.

```
Get-OrganizationRelationship -DomainController
'mail.contoso.com'
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Organization relationships" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active

			Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationRelationshipIdParameter	The *Identity* parameter specifies the identity of the organizational relationship. You can use the following values: • Canonical name • GUID • Name
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-OrganizationRelationship

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-OrganizationRelationship** cmdlet to create a relationship with an external Microsoft Exchange Server 2010 and Exchange Server 2013 organization for the purpose of accessing calendar free/busy information or moving mailboxes between on-premises Exchange servers and

the Exchange Online service as part of a hybrid deployment.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-OrganizationRelationship -DomainNames <MultiValuedProperty> -Name
<String> [-ArchiveAccessEnabled <$true | $false>] [-Confirm
[<SwitchParameter>]] [-DeliveryReportEnabled <$true | $false>] [-
DomainController <Fqdn>] [-Enabled <$true | $false>] [-
FreeBusyAccessEnabled <$true | $false>] [-FreeBusyAccessLevel <None |
AvailabilityOnly | LimitedDetails>] [-FreeBusyAccessScope
<GroupIdParameter>] [-MailboxMoveEnabled <$true | $false>] [-
MailTipsAccessEnabled <$true | $false>] [-MailTipsAccessLevel <None |
Limited | All>] [-MailTipsAccessScope <GroupIdParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationContact <SmtpAddress>] [-
PhotosEnabled <$true | $false>] [-TargetApplicationUri <Uri>] [-
TargetAutodiscoverEpr <Uri>] [-TargetOwaURL <Uri>] [-TargetSharingEpr
<Uri>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates an organization relationship with Contoso. The domain name to connect to is contoso.com. The following settings are used:

- Free/busy access is enabled.
- The requesting organization receives time, subject, and location information from the target organization.

This example attempts to automatically discover configuration information from the external organization by using the domain names provided in the **Get-FederationInformation** command. If you use this method to create your organization relationship, you must first ensure that you've created an organization identifier by using the **Set-FederationOrganizationIdentifier** cmdlet.

```
Get-FederationInformation -DomainName Contoso.com | New-
OrganizationRelationship -Name "Contoso" -
FreeBusyAccessEnabled $true -FreeBusyAccessLevel -
LimitedDetails
```

EXAMPLE 2

This example creates the organization relationship with Fourth Coffee using the following settings. In this example, the connection settings with the external organization are provided.

- The domain to connect to is mail.fourthcoffee.com.
- The Exchange Web Services application URL is mail.fourthcoffee.com.
- The Autodiscover URL is https://mail.fourthcoffee.com/autodiscover/autodiscover.svc/wssecurity.
- Free/busy access is enabled.
- The requesting organization only receives free/busy information with the time.

```
New-OrganizationRelationship -Name "Fourth Coffee" -
DomainNames "mail.fourthcoffee.com" -FreeBusyAccessEnabled
```

```
$true -FreeBusyAccessLevel -AvailabilityOnly -
TargetAutodiscoverEpr "https://mail.fourthcoffee.com/
autodiscover/autodiscover.svc/wssecurity" -
TargetApplicationUri "mail.fourthcoffee.com"
```

Detailed Description

Before you can create an organization relationship, you must first create a federation trust. For more information, see Federation.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Organization relationships" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
DomainNames	Required	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *DomainNames* parameter specifies the SMTP domains of the external organization. If adding multiple domain names, separate each entry with a comma.
Name	Required	System.String	The *Name* parameter specifies the meaningful name of the organization relationship.
ArchiveAccessEnabled	Optional	System.Boolean	The *ArchiveAccessEnabled* parameter specifies whether the organization

			relationship has been configured to provide remote archive access. Valid input for the *ArchiveAccessEnabled* parameter is `$true` or `$false`. The default value is `$false`. When the *ArchiveAccessEnabled* parameter is set to `$true`, the external organization specified in the organization relationship provides remote access to mailbox archives.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DeliveryReportEnabled	Optional	System.Boolean	The *DeliveryReportEnabled* parameter specifies whether Delivery Report data should be shared over this organization relationship.

			The accepted values are `$true` or `$false`. The default value is `$false`. If set to `$true`, this means the following two things: • The organization has agreed to share all Delivery Report data with the organization specified in the organization relationship. • This organization relationship should be used to retrieve Delivery Report information from the organization referenced in the organization relationship. For message tracking to work in a cross-premise, Exchange scenario, the *DeliveryReportEnabled* parameter must be set to `$true` on both sides of the organization relationship. If one, or both, of the members of the organization relationship specifies the *DeliveryReportEnabled*

			parameter as `$false`, tracking between the organizations won't work in either direction.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether to enable the sharing relationship. This parameter can be used to completely stop sharing for a particular relationship. The valid values for this parameter are `$true` or `$false`. The default value is `$true`.
FreeBusyAccessEnabled	Optional	System.Boolean	The *FreeBusyAccessEnabled* parameter specifies whether this organization relationship should be

			used for retrieving free/busy information from the external organization. The valid values for this parameter are `$true` or `$false`. The default value is `$false`.
FreeBusyAccessLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.FreeBusyAccessLevel	The *FreeBusyAccessLevel* parameter specifies the maximum amount of detail returned to the requesting organization. You can use the following values: • `None` No free/busy access • `AvailabilityOnly` Free/busy access with time only • `LimitedDetails` Free/busy access with time, subject, and location
FreeBusyAccessScope	Optional	Microsoft.Exchange.Configuration.Tasks.GroupIdParameter	The *FreeBusyAccessScope* parameter specifies a security distribution group in the internal organization that contains users that can have their free/busy information accessed by an external organization. You can use the following values: • Distinguished name (DN)

			• Canonical name • GUID • Name • Display name
MailboxMoveEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *MailboxMoveEnabled* parameter specifies whether the organization relationship is used for moving mailboxes to the external organization. If this parameter isn't set, the move requests require an administrator to provide a remote credential for the remote organization. The valid values for this parameter are `$true` or `$false`. The default value is `$false`.
MailTipsAccessEnabled	Optional	System.Boolean	The *MailTipsAccessEnabled* parameter specifies whether MailTips data for users in this organization are returned over this organization relationship. The accepted values are `$true` or `$false`. The default value is `$false`.

MailTipsAccessLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MailTipsAccessLevel	The *MailTipsAccessLevel* parameter specifies the level of MailTips data that's externally shared over this organization relationship. This parameter can have the following values: • `All` All MailTips are returned, but the recipients in the remote organization are considered external. For the Auto Reply MailTip, the external Auto Reply message is returned. • `Limited` Only those MailTips that could prevent a non-delivery report (NDR) or an Auto Reply are returned. Custom MailTips, the Large Audience MailTip, and Moderated Recipient MailTips won't be returned. • `None` No MailTips are returned to the remote organization. The default value is `None`.
MailTipsAccessScope	Optional	Microsoft.Exchange.Configuration.Tasks.GroupIdParameter	The *MailTipsAccessScope* parameter specifies a Security Distribution group in the organization that contains users for whom recipient-specific MailTips are returned over

			this organization relationship. The recipient-specific MailTips are: • Auto Reply • Mailbox Full • Custom If a group is specified, these MailTips are returned only for those recipients that are members of the specified group. If a group isn't specified, recipient-specific MailTips are returned for all recipients in the organization. By default, no group is specified. This restriction only applies to mailboxes, mail users, and mail contacts. It doesn't apply to distribution groups.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationContact	Optional	Microsoft.Exchange.Data.SmtpAddress	The *OrganizationContact* parameter specifies the email address that can be used to contact the

			external organization, for example, administrator@fourthcoffee.com.
PhotosEnabled	Optional	System.Boolean	The *PhotosEnabled* parameter specifies whether photo data for users in this organization are returned over this organization relationship. The accepted values are `$true` or `$false`. The default value is `$false`.
TargetApplicationUri	Optional	System.Uri	The *TargetApplicationUri* parameter specifies the target Uniform Resource Identifier (URI) of the external organization. The *TargetApplicationUri* parameter is specified by Exchange when requesting a delegated token for the external organization to fetch free and busy information, for example, mail.contoso.com.
TargetAutodiscoverEpr	Optional	System.Uri	The *TargetAutodiscoverEpr* parameter specifies the Autodiscover URL of Exchange Web Services

			for the external organization, for example, https://contoso.com/autodiscover/autodiscover.svc/wssecurity. Exchange uses the Autodiscover service to automatically detect the correct Client Access server endpoint for external requests.
TargetOwaURL	Optional	System.Uri	The *TargetOwaURL* parameter specifies the Microsoft Office Outlook Web App URL of the external organization defined in the organization relationship. It is used for Outlook Web App redirection in a cross-premise Exchange scenario. Configuring this attribute enables users in the organization to use their current Outlook Web App URL to access Outlook Web App in the external organization.
TargetSharingEpr	Optional	System.Uri	The *TargetSharingEpr* parameter specifies the URL of the target Exchange Web Services

			for the external organization. If the *TargetSharingEpr* parameter is used, Exchange always uses this URL to reach the external Client Access server and doesn't use the *TargetAutodiscoverEpr* parameter information to locate the Client Access server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-OrganizationRelationship

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-OrganizationRelationship** cmdlet to remove the organization relationship with an external Exchange organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-OrganizationRelationship -Identity
<OrganizationRelationshipIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the organization relationship Contoso using the *Identity* parameter.

```
Remove-OrganizationRelationship -Identity Contoso
```

Detailed Description

The **Remove-OrganizationRelationship** cmdlet removes the organization relationship objects. To stop sharing information without removing the organization relationship objects, disable the organization relationship by using the **Set-OrganizationRelationship** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Organization relationships" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.Orga	The *Identity* parameter specifies the identity of

		nizationRelationshipId Parameter	the organization relationship that you want to remove. You can use one of the following values: • Canonical name • GUID • Name
Confirm	Optional	System.Management. Automation.SwitchPar ameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Da ta.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management. Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-OrganizationRelationship

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-OrganizationRelationship** cmdlet to modify a relationship with an external Exchange organization for the purposes of accessing calendar free/busy information or moving mailboxes between on-premises Exchange servers and the Exchange Online service as part of a hybrid deployment.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-OrganizationRelationship -Identity
```

```
<OrganizationRelationshipIdParameter> [-ArchiveAccessEnabled <$true |
$false>] [-Confirm [<SwitchParameter>]] [-DeliveryReportEnabled <$true |
$false>] [-DomainController <Fqdn>] [-DomainNames <MultiValuedProperty>]
[-Enabled <$true | $false>] [-Force <SwitchParameter>] [-
FreeBusyAccessEnabled <$true | $false>] [-FreeBusyAccessLevel <None |
AvailabilityOnly | LimitedDetails>] [-FreeBusyAccessScope
<GroupIdParameter>] [-MailboxMoveEnabled <$true | $false>] [-
MailTipsAccessEnabled <$true | $false>] [-MailTipsAccessLevel <None |
Limited | All>] [-MailTipsAccessScope <GroupIdParameter>] [-Name <String>]
[-OrganizationContact <SmtpAddress>] [-PhotosEnabled <$true | $false>] [-
TargetApplicationUri <Uri>] [-TargetAutodiscoverEpr <Uri>] [-TargetOwaURL
<Uri>] [-TargetSharingEpr <Uri>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the free/busy access level to `LimitedDetails`, which includes time, subject, and location.

```
Set-OrganizationRelationship -Identity "Fourth Coffee" -
FreeBusyAccessLevel LimitedDetails
```

EXAMPLE 2

This example disables the organization relationship with Contoso

```
Set-OrganizationRelationship -Identity "Contoso" -Enabled
$false
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Organization relationships" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.OrganizationRelationshipIdParameter	The *Identity* parameter specifies the organization relationship to be modified. You can use the following values: • Canonical name

			• GUID • Name
ArchiveAccessEnabled	Optional	System.Boolean	The *ArchiveAccessEnabled* parameter specifies whether the organization relationship has been configured to provide remote archive access. Valid input for the *ArchiveAccessEnabled* parameter is `$true` or `$false`. The default value is `$false`. When the *ArchiveAccessEnabled* parameter is set to `$true`, the external organization specified in the organization relationship provides remote access to mailbox archives.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DeliveryReportEnabled	Optional	System.Boolean	The *DeliveryReportEnabled*

			parameter specifies whether Delivery Report data should be shared over this organization relationship. The accepted values are `$true` or `$false`. The default value is `$false`. If set to `$true`, this means the following two things: • The organization has agreed to share all Delivery Report data with the organization specified in the organization relationship. • This organization relationship should be used to retrieve Delivery Report information from the organization referenced in the organization relationship. For message tracking to work in a cross-premises, Exchange scenario, the *DeliveryReportEnabled* parameter must be set to `$true` on both sides of the organization relationship.

			If one, or both, of the members of the organization relationship specifies the *DeliveryReportEnabled* parameter as `$false`, tracking between the organizations won't work in either direction.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
DomainNames	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *DomainNames* parameter specifies the SMTP domains of the external organization. If adding multiple domain names, separate each entry with a comma, for example, "contoso.com","northamerica.contoso.com".
Enabled	Optional	System.Boolean	The *Enabled* parameter

			specifies whether to enable the organization relationship. This parameter can be used to completely stop the sharing for a particular relationship. The valid input values are `$true` or `$false`. The default value is `$true`.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress the warning or confirmation messages that appear during specific configuration changes.
FreeBusyAccessEnabled	Optional	System.Boolean	The *FreeBusyAccessEnabled* parameter specifies whether this organization relationship should be used for retrieving free/busy information from the external organization. The accepted values are `$true` or `$false`. The default value is `$false`.
FreeBusyAccessLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.FreeBusyA	The *FreeBusyAccessLevel* parameter specifies the maximum amount of

		ccessLevel	detail returned to the requesting organization. You can use the following values: • `None` No free/busy access • `AvailabilityOnly` Free/busy access with time only • `LimitedDetails` Free/busy access with time, subject, and location
FreeBusyAccessScope	Optional	Microsoft.Exchange.Configuration.Tasks.GroupIdParameter	The *FreeBusyAccessScope* parameter specifies a security distribution group in the internal organization that contains users that can have their free/busy information accessed by an external organization. You can use the following values: • Canonical name • Display name • Distinguished name (DN) • GUID • Name
MailboxMoveEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *MailboxMoveEnabled* parameter specifies that the organization

			relationship is used to provide the credential information for moving mailboxes to the external organization. If this parameter isn't set, the move requests require an administrator to provide a remote credential for the remote organization. The accepted values are `$true` or `$false`. The default value is `$false`.
MailTipsAccessEnabled	Optional	System.Boolean	The *MailTipsAccessEnabled* parameter specifies whether MailTips data for users in this organization are returned over this organization relationship. The accepted values are `$true` or `$false`. The default value is `$false`.
MailTipsAccessLevel	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MailTipsAccessLevel	The *MailTipsAccessLevel* parameter specifies the level of MailTips data externally shared over this organization relationship. This parameter can have the following values: • `All` All MailTips are returned, but the recipients in the remote

			organization are considered external. For the Auto Reply MailTip, the external Auto Reply message is returned. • `Limited` Only those MailTips that could prevent a non-delivery report (NDR) or an Auto Reply are returned. Custom MailTips, the Large Audience MailTip, and Moderated Recipient MailTips won't be returned. • `None` No MailTips are returned to the remote organization. The default value is `None`.
MailTipsAccessScope	Optional	Microsoft.Exchange.Configuration.Tasks.GroupIdParameter	The *MailTipsAccessScope* parameter specifies a Security Distribution group in the organization that contains users for whom recipient-specific MailTips are returned over this organization relationship. The recipient-specific MailTips are: • Auto Reply • Mailbox Full • Custom If a group is specified, these MailTips are returned only for those recipients that are

			members of the specified group. If a group isn't specified, recipient-specific MailTips are returned for all recipients in the organization. By default, no group is specified. This restriction only applies to mailboxes, mail users, and mail contacts. It doesn't apply to distribution groups.
Name	Optional	System.String	The *Name* parameter specifies the meaningful name of the organization relationship. Use this parameter to change the name of the organization relationship.
OrganizationContact	Optional	Microsoft.Exchange.Data.SmtpAddress	The *OrganizationContact* parameter specifies the email address that can be used to contact the external organization, for example, administrator@fourthcoffee.com.
PhotosEnabled	Optional	System.Boolean	The *PhotosEnabled* parameter specifies whether photo data for

			users in this organization are returned over this organization relationship. The accepted values are `$true` or `$false`. The default value is `$false`.
TargetApplicationUri	Optional	System.Uri	The *TargetApplicationUri* parameter specifies the target Uniform Resource Identifier (URI) of the external organization. The *TargetApplicationUri* parameter is specified by Exchange when requesting a delegated token to retrieve free and busy information, for example, mail.contoso.com.
TargetAutodiscoverEpr	Optional	System.Uri	The *TargetAutodiscoverEpr* parameter specifies the Autodiscover URL of Exchange Web Services for the external organization, for example, https://contoso.com/autodiscover/autodiscover.svc/wssecurity. Exchange uses Autodiscover to automatically detect the

			correct Client Access server endpoint for external requests.
TargetOwaURL	Optional	System.Uri	The *TargetOwaURL* parameter specifies the Microsoft Office Outlook Web App URL of the external organization defined in the organization relationship. It is used for Outlook Web App redirection in a cross-premise Exchange scenario. Configuring this attribute enables users in the organization to use their current Outlook Web App URL to access Outlook Web App in the external organization.
TargetSharingEpr	Optional	System.Uri	The *TargetSharingEpr* parameter specifies the URL of the target Exchange Web Services for the external organization. If the *TargetSharingEpr* parameter is used, Exchange always uses this URL to reach the external Client Access server and doesn't use the

			TargetAutoDiscoverEpr parameter information to locate the Client Access server.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-OrganizationRelationship

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Test-OrganizationRelationship** cmdlet to verify that the organization relationship is properly configured and functioning as expected.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-OrganizationRelationship -UserIdentity <RecipientIdParameter> [-
Identity <OrganizationRelationshipIdParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example validates the organization relationship deployed in the Exchange organization and checks whether a delegation token can be retrieved for a mailbox for the external organization domain contoso.com.

```
Test-OrganizationRelationship -UserIdentity
katherine@contoso.com -Identity contoso.com -Confirm
```

Detailed Description

The **Test-OrganizationRelationship** cmdlet doesn't include any functional tests of federated sharing features, such as accessing user free/busy information or moving mailboxes between organizations. It only verifies that the configuration will allow these features to work correctly.

Before you can test an organization relationship, you must first create an organization relationship. For more information, see Create an organization relationship.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Organization relationships" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
UserIdentity	Required	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *UserIdentity* parameter specifies the mailbox for which a delegation token is requested to access the

			external organization's configuration information. You can use any of the following values: • Distinguished name (DN) • Canonical name • GUID • Name • Display name • Alias • Exchange DN • Primary SMTP email address
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationRelationshipIdParameter	The *Identity* parameter specifies the organization relationship to be tested. You can use the following values: • Canonical name • GUID • Name
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PublicFolder

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-PublicFolder** cmdlet to retrieve the attributes of a public folder or a set of public folders.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PublicFolder [-Identity <PublicFolderIdParameter>] <COMMON PARAMETERS>

Get-PublicFolder -Recurse <SwitchParameter> [-Identity
<PublicFolderIdParameter>] [-ResultSize <Unlimited>] <COMMON PARAMETERS>

Get-PublicFolder -GetChildren <SwitchParameter> [-Identity
<PublicFolderIdParameter>] [-ResultSize <Unlimited>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-Mailbox
<MailboxIdParameter>] [-Organization <OrganizationIdParameter>] [-
ResidentFolders <SwitchParameter>]
```

Examples

EXAMPLE 1

This example uses the **Get-PublicFolder** command without parameters to return the root public folder object (IPM_SUBTREE).

```
Get-PublicFolder
```

EXAMPLE 2

This example returns the names of all the system folders (which aren't shown by default), starting at the system folder root (\NON_IPM_SUBTREE).

```
Get-PublicFolder -Identity \NON_IPM_SUBTREE -Recurse |
Format-List Name
```

EXAMPLE 3

This example returns the Pending Litigation public folder from \Legal\Documents\.

```
Get-PublicFolder -Identity "\Legal\Documents\Pending
Litigation"
```

EXAMPLE 4

This example returns the Pending Litigation public folder from \Legal\Documents\ and all the public folders under the Pending Litigation public folder. Because the result size isn't specified, the command returns up to the maximum number of public folders, which is 10,000.

```
Get-PublicFolder -Identity "\Legal\Documents\Pending
Litigation" -Recurse
```

EXAMPLE 5

This example returns the Pending Litigation public folder from \Legal\Documents\ and all the public folders under the Pending Litigation public folder, without a limit on the number returned.

```
Get-PublicFolder -Identity "\Legal\Documents\Pending
Litigation" -Recurse -ResultSize Unlimited
```

EXAMPLE 6

This example returns the public folders that reside in the public folder content mailbox Legal Department.

```
Get-PublicFolder -Mailbox "Legal Department" -
ResidentFolders
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
GetChildren	Required	System.Management.	The *GetChildren*

		Automation.SwitchParameter	parameter specifies whether to retrieve only the children of the folder specified by the *Identity* parameter. You can't use the *GetChildren* parameter with the *Recurse* parameter.
Recurse	Required	System.Management.Automation.SwitchParameter	The *Recurse* parameter specifies that the command must return the specified public folder and all its children. You don't need to specify a value with this parameter. You can't use the *GetChildren* parameter with the *Recurse* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.Publi	The *Identity* parameter specifies the GUID or

		cFolderIdParameter	public folder name that represents a specific public folder. You can also include the path using the format *TopLevelPublicFolder* *PublicFolder*. You can omit the parameter label *Identity* so that only the public folder name or GUID is supplied.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the hierarchy public folder mailbox. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-

			4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ResidentFolders	Optional	System.Management.Automation.SwitchParameter	The *ResidentFolders* specifies whether to return public folders that reside in a specific content public folder mailbox. If this parameter isn't specified, the command will only return public folders whose contents reside in the primary hierarchy public folder mailbox.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum

			number of results to return. The default maximum is 10,000. For no limit on the returned results, set this parameter to `Unlimited`. This parameter can only be passed in combination with the *Recurse* or *GetChildren* parameters.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-PublicFolder

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-PublicFolder** cmdlet to create a public folder with the specified name.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-PublicFolder -Name <String> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-EformsLocaleId <CultureInfo>] [-Mailbox
<MailboxIdParameter>] [-Organization <OrganizationIdParameter>] [-Path
<PublicFolderIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the public folder Marketing in the root of the public folder.

```
New-PublicFolder -Name Marketing
```

EXAMPLE 2

This example creates the public folder FY2013 under the existing folders \Legal\Cases. The path to the new folder is \Legal\Cases\FY2013.

```
New-PublicFolder -Name FY2013 -Path \Legal\Cases
```

EXAMPLE 3

This example creates the public folder Support in the North_America hierarchy public folder mailbox.

```
New-PublicFolder -Name Support -Mailbox North_America
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name for the public folder.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EformsLocaleId	Optional	System.Globalization.CultureInfo	The *EformsLocaleId* parameter specifies the locale-specific version of the e-forms library. The valid input for the *EformsLocaleId* parameter is the string names listed in the **Culture Name** column in the Microsoft .NET Class Library class reference available at CultureInfo Class.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the hierarchy public folder

			mailbox in which you want this public folder created. You can use the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SMTP address • Alias
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Path	Optional	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter	The *Path* parameter specifies the location of the folder in the folder hierarchy, for example, \Legal\Cases.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-PublicFolder

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-PublicFolder** cmdlet to remove an existing public folder.

Caution:

The **Remove-PublicFolder** cmdlet removes the public folder data from all servers in your organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-PublicFolder -Identity <PublicFolderIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Recurse
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the public folder My Public Folder from the \Test\Directory tree.

```
Remove-PublicFolder -Identity "\Test\Directory\My Public
Folder"
```

EXAMPLE 2

This example deletes the public folder Directory Folder and all its child public folders by using the *Recurse* switch.

```
Remove-PublicFolder -Identity "\Test\Directory Folder" -
Recurse
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also include the path using the format *\TopLevelPublicFolder\PublicFolder*. You can omit the parameter label so that only the public folder name or GUID is supplied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default

			when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Recurse	Optional	System.Management.Automation.SwitchParameter	The *Recurse* switch specifies whether all subfolders of the specified folder should be removed. If the *Recurse* switch isn't specified and the public folder has subfolders, the command doesn't run and an error message is returned.
WhatIf	Optional	System.Management.A	The *WhatIf* switch

		utomation.SwitchParameter	instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-PublicFolder

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-PublicFolder** cmdlet to set the attributes of public folders.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-PublicFolder -Identity <PublicFolderIdParameter> [-AgeLimit
<EnhancedTimeSpan>] [-Confirm [<SwitchParameter>]] [-DomainController
```

```
<Fqdn>] [-EformsLocaleId <CultureInfo>] [-Force <SwitchParameter>] [-
IssueWarningQuota <Unlimited>] [-LastMovedTime <ExDateTime>] [-MaxItemSize
<Unlimited>] [-Name <String>] [-OverrideContentMailbox
<MailboxIdParameter>] [-Path <PublicFolderIdParameter>] [-
PerUserReadStateEnabled <$true | $false>] [-ProhibitPostQuota <Unlimited>]
[-RetainDeletedItemsFor <EnhancedTimeSpan>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the content location of the public folder hierarchy mailbox to North_America.

```
Set-PublicFolder "\Customer Service Requests" -
OverrideContentMailbox North_America
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also include the path using the format *TopLevelPublicFolder**PublicFolder*.
AgeLimit	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *AgeLimit* parameter specifies the overall age limit on the folder. Replicas of this public

			folder are automatically deleted when the age limit is exceeded.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EformsLocaleId	Optional	System.Globalization.CultureInfo	The *EformsLocaleId* parameter specifies the locale-specific version of the e-forms library. The valid input for the *EformsLocaleId* parameter is the string names listed

			in the **Culture Name** column in the Microsoft .NET Class Library class reference available at CultureInfo Class.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* switch specifies whether to suppress warning or confirmation messages. This switch can be used when the task is run programmatically and prompting for administrative input is inappropriate. If the *Force* switch isn't provided in the command, you're prompted for administrative input. You don't have to specify a value with this parameter.
IssueWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *IssueWarningQuota* parameter specifies the public folder size that triggers a warning to public folder owners stating that the folder is almost full. The default value is `unlimited`, which is 2 terabytes. When you enter a value, qualify the value with one of the following

			units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as kilobytes. The valid input range for this parameter is from 1 through 2TB.
LastMovedTime	Optional	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *LastMovedTime* parameter specifies the date and time that the public folder was last moved. Use the short date format defined in the **Regional Options** settings for the computer on which the command is run. For example, if the computer is configured to use the short date format mm/dd/yyyy, enter **03/01/2010** to specify March 1, 2010. You can enter the date only, or you can enter the date and time of day. If you enter the date and time of day, you must enclose the argument in quotation marks ("), for

			example, **"10/05/2010 5:00 PM"**.
MaxItemSize	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxItemSize* parameter specifies the maximum size for posted items. Items larger than the value of the *MaxItemSize* parameter are rejected. The default value is `unlimited`, which is 2 gigabytes. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as kilobytes. The valid input range for this parameter is from 1 through 2GB.
Name	Optional	System.String	The *Name* parameter specifies the name for the public folder.
OverrideContentMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *OverrideContentMailbox*

			parameter specifies the identity of the public folder mailbox that you want to move this public folder's content to.
Path	Optional	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter	The *Path* parameter specifies the path of the public folder, for example, *\TopLevelPublicFolder\PublicFolder*.
PerUserReadStateEnabled	Optional	System.Boolean	The *PerUserReadStateEnabled* parameter specifies whether to maintain read and unread data on a per-user basis.
ProhibitPostQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *ProhibitPostQuota* parameter specifies the size of a public folder at which users are notified that the public folder is full. Users can't post to a folder whose size is larger than the *ProhibitPostQuota* parameter value. The default value is `unlimited`, which is 2 terabytes. When you enter a value, qualify the value with one of the following units: • `B` (bytes)

			• KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as kilobytes. The valid input range for this parameter is from 1 through 2TB.
RetainDeletedItemsFor	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *RetainDeletedItemsFor* parameter specifies the retention time for deleted items.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-PublicFolderClientPermission

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Add-PublicFolderClientPermission** cmdlet to add permissions to public folders.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-PublicFolderClientPermission -Identity <PublicFolderIdParameter> -
AccessRights <MailboxFolderAccessRight[]> -User
<MailboxFolderUserIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds permission for the user Chris to create items in the public folder My Public Folder.

```
Add-PublicFolderClientPermission -Identity "\My Public
Folder" -User Chris -AccessRights CreateItems
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
AccessRights	Required	Microsoft.Exchange.M	The *AccessRights*

		anagement.StoreTasks.MailboxFolderAccessRight[]	parameter specifies the rights being added. This parameter accepts the following values: • `ReadItems` The user has the right to read items within the specified public folder. • `CreateItems` The user has the right to create items within the specified public folder. • `EditOwnedItems` The user has the right to edit the items that the user owns in the specified public folder. • `DeleteOwnedItems` The user has the right to delete items that the user owns in the specified public folder. • `EditAllItems` The user has the right to edit all items in the specified public folder. • `DeleteAllItems` The user has the right to delete all items in the specified public folder. • `CreateSubfolders` The user has the right to create subfolders in the specified public folder. • `FolderOwner` The user is the owner of the specified public folder. The user has the right to view and move the public folder and create subfolders. The user can't read items, edit

			items, delete items, or create items. • `FolderContact` The user is the contact for the specified public folder. • `FolderVisible` The user can view the specified public folder, but can't read or edit items within the specified public folder. In addition to access rights, you can create rights based upon roles, which includes multiple access rights. This parameter accepts the following values for roles: • `None` FolderVisible • `Owner` CreateItems, ReadItems, CreateSubfolders, FolderOwner, FolderContact, FolderVisible, EditOwnedItems, EditAllItems, DeleteOwnedItems, DeleteAllItems • `PublishingEditor` CreateItems, ReadItems, CreateSubfolders, FolderVisible, EditOwnedItems, EditAllItems, DeleteOwnedItems, DeleteAllItems • `Editor` CreateItems, ReadItems,

			FolderVisible, EditOwnedItems, EditAllItems, DeleteOwnedItems, DeleteAllItems • `PublishingAuthor` CreateItems, ReadItems, CreateSubfolders, FolderVisible, EditOwnedItems, DeleteOwnedItems • `Author` CreateItems, ReadItems, FolderVisible, EditOwnedItems, DeleteOwnedItems • `NonEditingAuthor` CreateItems, ReadItems, FolderVisible • `Reviewer` ReadItems, FolderVisible • `Contributor` CreateItems, FolderVisible
Identity	Required	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also include the path using the format *TopLevelPublicFolder**PublicFolder*. You can omit the parameter label so that only the public folder name or GUID is supplied.

User	Required	Microsoft.Exchange.Management.StoreTasks.MailboxFolderUserIdParameter	The *User* parameter specifies the user principal name (UPN), *domain**user*, or alias of the user for whom rights are being added.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PublicFolderClientPermission

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-PublicFolderClientPermission** cmdlet to retrieve the user permissions for a public folder.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PublicFolderClientPermission -Identity <PublicFolderIdParameter> [-
DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>] [-User
<MailboxFolderUserIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves the permissions for all users of \My Public Folder.

```
Get-PublicFolderClientPermission "\My Public Folder"
```

EXAMPLE 2

This example retrieves the permissions for the public folder My Public Folder, for the user Chris. In this example, the output of the **Get-PublicFolderClientPermission** command is piped to the **Format-List** command so that all available information is displayed in the result.

```
Get-PublicFolderClientPermission -Identity "\My Public
Folder" -User Chris | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also include the path using the format *\TopLevelPublicFolder\PublicFolder*. You can omit the parameter label so that only the public folder name or GUID is

			supplied.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the public folder mailbox for which you want to view the permissions. By default, the permissions are returned for the primary public folder mailbox. Using the *Mailbox* parameter allows you to specify a different public folder mailbox.
User	Optional	Microsoft.Exchange.Management.StoreTasks.MailboxFolderUserIdParameter	The *User* parameter specifies the user principal name (UPN), *domain**user*, or alias of a specific user for

			whom you want to view the permissions on the public folder.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-PublicFolderClientPermission

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-PublicFolderClientPermission** cmdlet to remove permissions from public folders.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-PublicFolderClientPermission -Identity <PublicFolderIdParameter> -
User <MailboxFolderUserIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes permission for the user Chris to the public folder My Public Folder.

```
Remove-PublicFolderClientPermission -Identity "\My Public
Folder" -User Contoso\Chris
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also include the path by using the format *\TopLevelPublicFolder\PublicFolder*. You can omit the parameter label so that only the public folder name or GUID is supplied.
User	Required	Microsoft.Exchange.Management.StoreTasks.MailboxFolderUserIdParameter	The *User* parameter specifies the user principal name (UPN), *domain\user*, or alias of the user whose permissions are being removed.
Confirm	Optional	System.Management.Automation.SwitchPara	The *Confirm* switch can be used to suppress the

		meter	confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$false`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PublicFolderDatabase

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-PublicFolderDatabase** cmdlet to view public folder database settings for Microsoft Exchange Server 2010 or earlier public folders.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PublicFolderDatabase [-Identity <DatabaseIdParameter>] <COMMON
PARAMETERS>

Get-PublicFolderDatabase -Server <ServerIdParameter> <COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-Status <SwitchParameter>]
```

Examples

EXAMPLE 1

This example returns all the attributes of all the public folder databases in the organization by piping the results of the **Get-PublicFolderDatabase** command to the **Format-List** command.

```
Get-PublicFolderDatabase | Format-List
```

EXAMPLE 2

This example returns information about the public folder database PFDatabase that resides on Server01.

```
Get-PublicFolderDatabase -Identity "Server01\PFDatabase"
```

EXAMPLE 3

This example returns information about all public folders on Server01.

```
Get-PublicFolderDatabase -Server Server01
```

Detailed Description

You can specify either the *Server* or *Identity* parameter, but not both. Only the *Server* and *Identity* parameters can be piped.

Note:
When you run the **Get-PublicFolderDatabase** cmdlet with no parameters, it returns attributes of all of the public folder databases in the Exchange organization. To return specific database properties (including backup and mount status information) where the **Get-PublicFolderDatabase** cmdlet has to contact servers directly or perform a complex or slow calculation, make sure you use the *Status* parameter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Server	Required	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name of a server that contains a public folder database. If specified, only the public folder database on the specified server is returned. This parameter can't be used with the

			Identity parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	The *Identity* parameter specifies a public folder database. You can use the following values: • GUID • Distinguished name (DN) • Database name If you don't specify the server name, the cmdlet searches for databases on the local server. If you have multiple databases with the same name, the cmdlet retrieves all databases with the same name in the specified scope. This parameter can't be used with the *Server* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Status	Optional	System.Management.Automation.SwitchParameter	The *Status* parameter specifies whether additional backup and mount status information is included for servers running Exchange 2010. If the *Status* parameter is included, additional backup and mount status information is included for Exchange 2010 servers.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PublicFolderItemStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-PublicFolderItemStatistics** cmdlet to view information about items within a specified public folder. Information returned includes subject, last modification time, last access time, creation time, attachments, message size, and the type of item. You can use this raw information to better understand the distribution of items and item characteristics across public folders.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PublicFolderItemStatistics -Identity <PublicFolderIdParameter> [-
DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>]
```

Examples

EXAMPLE 1

This example returns default statistics for all items in the Pamphlets public folder under the \Marketing\2013 path. Default information includes item identity, creation time, and subject.

```
Get-PublicFolderItemStatistics -Identity "\Marketing\2013
\Pamphlets"
```

EXAMPLE 2

This example returns additional information about the items within the public folder, such as subject, last modification time, creation time, attachments, message size, and the type of item by piping the results of the **Get-PublicFolderItemStatistics** command to the **Format-List** command.

```
Get-PublicFolderItemStatistics -Identity "\Marketing\2013
\Pamphlets" | Format-List
```

EXAMPLE 3

This example exports the output of the **Get-PublicFolderItemStatistics** command to the PFItemStats.csv file that includes the following information for all items within the public folder \Marketing\Reports:

- Subject of the message (*Subject*)
- Date and time when the item was last modified (*LastModificationTime*)
- If the item has attachments (*HasAttachments*)
- Type of item (*ItemType*)
- Size of the item (*MessageSize*)

```
Get-PublicFolderItemStatistics -Identity "\Marketing
\Reports" | Select
Subject,LastModificationTime,HasAttachments,ItemType,Messag
eSize | Export-CSV C:\PFItemStats.csv
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also include the path using the following format: *TopLevelPublicFolder**PublicFolder*
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the hierarchy public folder mailbox.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-PublicFolderMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Update-PublicFolderMailbox** cmdlet to update the hierarchy for public folders.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-PublicFolderMailbox -Identity <MailboxIdParameter> <COMMON
PARAMETERS>

Update-PublicFolderMailbox -Identity <MailboxIdParameter> [-FullSync
<SwitchParameter>] [-InvokeSynchronizer <SwitchParameter>] [-
SuppressStatus <SwitchParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example updates the public folder hierarchy on the public folder mailbox PF_marketing and suppresses the command's output.

```
Update-PublicFolderMailbox -Identity PF_marketing -
SuppressStatus
```

EXAMPLE 2

This example updates all public folder mailboxes and suppresses the command's output.

```
Get-Mailbox -PublicFolder | Update-PublicFolderMailbox -
SuppressStatus
```

Detailed Description

This cmdlet only needs to be used if you want to manually invoke the hierarchy synchronizer and the mailbox assistant. Both these are invoked at least once every 24 hours for each public folder mailbox in the organization. The hierarchy synchronizer is invoked every 15 minutes if any users are logged on to a secondary mailbox through Microsoft Outlook or a Microsoft Exchange Web Services client.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identity of the public folder mailbox. This parameter accepts the following values: • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • Legacy Exchange DN • SMTP address • Alias
Confirm	Optional	System.Management.	The *Confirm* switch can be

		Automation.SwitchParameter	used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$false`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
FullSync	Optional	System.Management.Automation.SwitchParameter	The *FullSync* parameter specifies that you want to perform a full synchronization of the public folder mailbox.
InvokeSynchronizer	Optional	System.Management.Automation.SwitchParameter	The *InvokeSynchronizer* parameter can only be used on secondary hierarchy public folder mailboxes and triggers hierarchy synchronization

			from the primary public folder mailbox to the specified secondary public folder mailbox. This parameter should only be used for troubleshooting purposes.
SuppressStatus	Optional	System.Management.Automation.SwitchParameter	The *SuppressStatus* parameter specifies that the output of this cmdlet is suppressed and that the command will run asynchronously in the background from the Exchange Management Shell. If you don't use this parameter in the command, the output will display status messages every 3 seconds for up to one minute. Until the minute passes, you can't use that instance of the Shell.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply

			any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PublicFolderMailboxDiagnostics

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-17*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-PublicFolderMailboxDiagnostics** cmdlet to view event-level information about a public folder mailbox. This information can be used to troubleshoot public folder issues.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PublicFolderMailboxDiagnostics -Identity <MailboxIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
IncludeDumpsterInfo <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example returns the diagnostic information for the public folder mailbox Customer Escalations.

```
Get-PublicFolderMailboxDiagnostics -Identity "Customer
```

```
Escalations"
```

EXAMPLE 2

This example returns the diagnostic information for the public folder mailbox Sales Forecasts and exports the report to a CSV file.

```
Get-PublicFolderMailboxDiagnostics -Identity "Sales
Forecasts" | Export-CSV C:\Diagnostics\SalesForecasts.csv
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identity of the public folder mailbox. The public folder mailbox is where the content of the public folder resides.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
IncludeDumpsterInfo	Optional	System.Management.Automation.SwitchParameter	The *IncludeDumpsterInfo* parameter specifies that diagnostic information for the \NON_IPM_TREE \DUMPSTER_ROOT folder, which serves as the dumpster for public folder mailboxes, is included in the returned information.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes

			would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-PublicFolderStatistics

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-PublicFolderStatistics** cmdlet to retrieve statistical information about public folders, such as folder size and last logon time.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-PublicFolderStatistics [-Identity <PublicFolderIdParameter>] [-
DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>] [-Organization
<OrganizationIdParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example retrieves statistics about the public folder Marketing\2013\Pamphlets. The output of

the **Get-PublicFolderStatistics** command is piped to the **Format-List** command so that all the available information is displayed in the result.

```
Get-PublicFolderStatistics -Identity "\Marketing\2013
\Pamphlets" | Format-List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Public folders" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.PublicFolderIdParameter	The *Identity* parameter specifies the GUID or public folder name that represents a specific public folder. You can also include the path using the format *\TopLevelPublicFolder*

			\PublicFolder. You can omit the parameter label so that only the public folder name or GUID is supplied.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the identity of the hierarchy public folder mailbox.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return statistics for all public folders that match the query, use `unlimited` for the value of this parameter. The default value is `100`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SharingPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-SharingPolicy** cmdlet to view the settings of sharing policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-SharingPolicy [-Identity <SharingPolicyIdParameter>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example retrieves the default information for the sharing policy FourthCoffee.

```
Get-SharingPolicy -Identity FourthCoffee
```

EXAMPLE 2

This example retrieves the full information for the sharing policy Fabrikam.

```
Get-SharingPolicy Fabrikam | Format List
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Sharing policies" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.SharingPolicyIdParameter	The *Identity* parameter specifies the identity of the sharing policy for which you want to view the settings. You can use one of the following values: • ADObjectID • Distinguished name (DN) • Legacy DN • GUID The *Identity* parameter can't be used with the *Organization* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-SharingPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-SharingPolicy** cmdlet to create a sharing policy to regulate how users inside your organization can share calendar and contact information with users outside the organization. Users can only share this information after federation has been configured in Exchange. After federation is configured, users can send sharing invitations that comply with a sharing policy to external recipients in other Microsoft Exchange Server 2013 and Exchange Server 2010 organizations that have federation enabled. A sharing policy needs to get assigned to a mailbox to be effective. If a mailbox doesn't have a specific sharing policy assigned, a default policy enforces the level of sharing permitted for this mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-SharingPolicy -Domains <MultiValuedProperty> -Name <String> [-Confirm
[<SwitchParameter>]] [-Default <SwitchParameter>] [-DomainController
<Fqdn>] [-Enabled <$true | $false>] [-Organization
<OrganizationIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the sharing policy Contoso for the contoso.com domain, which is a domain outside the organization. This policy allows users in the contoso.com domain to see detailed free/busy information and contacts. By default, this policy is enabled.

```
New-SharingPolicy -Name "Contoso" -Domains
```

```
'mail.contoso.com: CalendarSharingFreeBusyDetail,
ContactsSharing'
```

EXAMPLE 2

This example creates a default sharing policy, which is applied to all mailboxes that don't implicitly have a sharing policy assigned to them. This sharing policy SharingPolicy01 applies to two different domains, and the sharing policy is disabled.

```
New-SharingPolicy -Name "SharingPolicy01" -Domains
'mail.contoso.com: CalendarSharingFreeBusySimple',
'mail.fabrikam.com: CalendarSharingFreeBusySimple' -Enabled
$false -Default $true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Sharing policies" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Domains	Required	Microsoft.Exchange.Data.MultiValuedProperty	The *Domains* parameter specifies the domains to which this sharing policy applies and the sharing policy actions. Values for this parameter take the format: '*Domain:SharingPolicyAction*'. The following sharing policy action values can be used: • **CalendarSharingFreeBusySimple** Share free/busy hours only

			• **CalendarSharingFreeBusyDetail** Share free/busy hours, subject, and location • **CalendarSharingFreeBusyReviewer** Share free/busy hours, subject, location, and the body of the message or calendar item • **ContactsSharing** Share contacts only Separate multiple domains with a comma, for example, 'mail.contoso.com: CalendarSharingFreeBusySimple', 'mail.fabrikam.com: CalendarSharingFreeBusyDetail, ContactsSharing'. **Note:** A domain doesn't include subdomains. You must configure each subdomain separately.
Name	Required	System.String	The *Name* parameter specifies the name of the new sharing policy.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Default	Optional	System.Management.Automation.SwitchParameter	The *Default* switch specifies that this sharing policy is the default sharing policy for all mailboxes. If no sharing policy has been applied to a mailbox, the default policy is automatically applied. If you want to disable sharing across your organization, you can set the default policy as disabled. You don't have to specify a value with this switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether to enable the new sharing policy. Valid input for this parameter is `$true` or `$false`. The default is `$true`. When the sharing policy is disabled, users who are provisioned to use this policy continue to share information until the sharing policy assistant runs and removes the permissions on the shared folder. The frequency with which the sharing policy assistant runs is assigned in the **Set-MailboxServer** cmdlet using the *SharingPolicySchedule* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-SharingPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-SharingPolicy** cmdlet to remove a sharing policy. Before you can remove a sharing policy, you must ensure that no mailbox users are provisioned to use that policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-SharingPolicy -Identity <SharingPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the sharing policy Fabrikam.

```
Remove-SharingPolicy Fabrikam
```

EXAMPLE 2

This example removes the sharing policy Contoso and suppresses the confirmation that you want to remove the policy.

```
Remove-SharingPolicy -Identity Contoso -Confirm:$false
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Sharing policies" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.SharingPolicyIdParameter	The *Identity* parameter specifies the identity of the sharing policy that you want to remove. You can use one of the following values: • ADObjectID • Distinguished name (DN) • Legacy DN • GUID
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when

			this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-SharingPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-SharingPolicy** cmdlet to modify an existing sharing policy to regulate how users inside your organization can share free/busy and contact information with users outside the organization. Users can only share this information after federation has been configured in Exchange. After federation is configured, users can send sharing invitations that comply with a sharing policy to external recipients in other Microsoft Exchange Server 2010 and Exchange Server 2013 organizations that have federation enabled. A sharing policy needs to be assigned to a mailbox to be effective. If a mailbox doesn't have a specific sharing policy assigned, a default policy enforces the level of sharing permitted for this mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-SharingPolicy -Identity <SharingPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-Default <SwitchParameter>] [-DomainController
<Fqdn>] [-Domains <MultiValuedProperty>] [-Enabled <$true | $false>] [-
Name <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the sharing policy Contoso for contoso.com, which is a domain outside your organization. This policy allows users in the Contoso domain to see simple free/busy information.

```
Set-SharingPolicy -Identity Contoso -Domains
'mail.contoso.com: CalendarSharingFreeBusySimple'
```

EXAMPLE 2

This example adds a second domain to the sharing policy SharingPolicy01. When you're adding a domain to an existing policy, you must include any previously included domains.

```
Set-SharingPolicy -Identity SharingPolicy01 -Domains
'contoso.com: CalendarSharingFreeBusySimple',
'atlanta.contoso.com: CalendarSharingFreeBusyReviewer',
'beijing.contoso.com: CalendarSharingFreeBusyReviewer'
```

EXAMPLE 3

This example disables the sharing policy SharingPolicy01.

```
Set-SharingPolicy -Identity "SharingPolicy01" -Enabled
$false
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Sharing policies" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.SharingPolicyIdParameter	The *Identity* parameter specifies the identity of the sharing policy that you want to modify. You can use one of the following values: • ADObjectID • Distinguished name (DN) • Legacy DN • GUID

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Default	Optional	System.Management.Automation.SwitchParameter	The *Default* switch specifies that this sharing policy is the default sharing policy for all mailboxes. If no sharing policy has been applied to a mailbox, the default policy is automatically applied. If you want to disable sharing across your organization, you can set the default policy to be disabled. You don't have to specify a value with this parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Domains	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Domains* parameter specifies domains to which this policy applies and the sharing policy action. Values for this parameter use the format *Domain*: *SharingPolicyAction*. The following sharing policy action values can be used: • `CalendarSharingFreeBusySimple` Share free/busy hours only • `CalendarSharingFreeBusyDetail` Share free/busy hours, subject, and location • `CalendarSharingFreeBusyReviewer` Share free/busy hours, subject, location, and the body of the message or calendar item • `ContactsSharing` Share contacts only

			The following is an example: `'Contoso.com: Ca` **Note:** Adding a domain doesn't include subdomains. You must configure each subdomain separately.
Enabled	Optional	System.Boolean	The *Enabled* parameter specifies whether to enable the sharing policy. Valid values for this parameter are `$true` or `$false`. The default is `$true`. When the sharing policy is disabled, users who are provisioned to use this policy continue to share information until the sharing policy assistant runs.
Name	Optional	System.String	The *Name* parameter specifies the name of the sharing policy.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can

			view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SiteMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-SiteMailbox** cmdlet to view information about site mailboxes. This cmdlet is primarily used by Microsoft SharePoint and Exchange to display information to users in the user interface. However, you may find it helpful for discovering information such as the site mailbox's owners, members, and lifecycle status.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-SiteMailbox <COMMON PARAMETERS>

Get-SiteMailbox [-BypassOwnerCheck <SwitchParameter>] [-DeletedSiteMailbox
<SwitchParameter>] [-Organization <OrganizationIdParameter>] <COMMON
PARAMETERS>
```

```
COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-Anr
<String>] [-DomainController <Fqdn>] [-Identity <RecipientIdParameter>] [-
ReadFromDomainController <SwitchParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns the default information about the site mailbox ContentSite, which includes the site name, when the site mailbox was closed, and the SharePoint URL.

```
Get-SiteMailbox -BypassOwnerCheck -Identity ContentSite
```

EXAMPLE 2

This example returns the full information about the site mailbox ContentSite.

```
Get-SiteMailbox -BypassOwnerCheck -Identity ContentSite |
Format-List
```

EXAMPLE 3

This example queries for site mailboxes that are marked for deletion and removes them from the mailbox database by pipelining the **Remove-Mailbox** cmdlet.

```
Get-SiteMailbox -BypassOwnerCheck -DeletedSiteMailbox |
Remove-Mailbox -Confirm:$false
```

Detailed Description

If you aren't a member or owner of the site mailbox that you want to view the diagnostics information for, you must use the *BypassOwnerCheck* parameter when running this cmdlet. If you aren't a member or owner of the site mailbox and you run this cmdlet without using the *BypassOwnerCheck* parameter, the command fails with an "object not found" error.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Site mailboxes" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Co	This parameter is reserved

		nfiguration.Tasks.AccountPartitionIdParameter	for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches the string. The default attributes searched are: • CommonName (CN) • DisplayName • Name • Alias
BypassOwnerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassOwnerCheck* parameter is used by administrators who aren't members or owners of the site mailbox. If you aren't a member or owner of the site mailbox and you run this cmdlet without using the *BypassOwnerCheck* parameter, the command fails with an "object not found" error.
DeletedSiteMailbox	Optional	System.Management.Automation.SwitchPar	This parameter is available only in on-

		ameter	premises Exchange 2013. The *DeletedSiteMailbox* parameter returns site mailboxes that have been marked for pending deletion. When the lifecycle application in SharePoint closes a site mailbox, the site mailbox is retained for the period specified in the lifecycle policy in the closed state. The mailbox can then be reactivated by an end user or by a SharePoint administrator. After the retention period, the Exchange site mailbox that's housed in the mailbox database will have its name prepended with **MDEL:** to indicate that it has been marked for deletion. To free storage space and the alias, use the Remove-Mailbox cmdlet to manually remove these site mailboxes from the mailbox database.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-

			premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *Identity* parameter specifies the identity of the site mailbox. You can use one of the following values: • GUID • Distinguished name (DN) • Display name • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	The *ReadFromDomainController* switch specifies that information should be read from a domain controller in the user's

			domain. If you run the command `Set-AdServerSettings -ViewEntireForest $true` to include all objects in the forest and you don't use the *ReadFromDomainController* switch, it's possible that information will be read from a global catalog that has outdated information. When you use the *ReadFromDomainController* switch, multiple reads might be necessary to get the information. You don't have to specify a value with this switch. **Note:** By default, the recipient scope is set to the domain that hosts your Exchange servers.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all site mailboxes that match the query, use `unlimited` for the value of this parameter. The default value is `1000`.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-SiteMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-SiteMailbox** cmdlet to change a site mailbox's settings, such as the Microsoft SharePoint URL. This cmdlet is primarily used by the SharePoint and Microsoft Exchange user interfaces, such as the SharePoint URL. This cmdlet should only be used for diagnostic and troubleshooting purposes.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-SiteMailbox <COMMON PARAMETERS>

Set-SiteMailbox [-ShowInMyClient <$true | $false>] <COMMON PARAMETERS>

COMMON PARAMETERS: -Identity <RecipientIdParameter> [-Active <$true |
$false>] [-Confirm [<SwitchParameter>]] [-DisplayName <String>] [-
DomainController <Fqdn>] [-Force <SwitchParameter>] [-Members
<RecipientIdParameter[]>] [-Owners <RecipientIdParameter[]>] [-
RemoveDuplicateMessages <$true | $false>] [-SharePointUrl <Uri>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the SharePoint URL for the MarketingEvents 2013 site mailbox.

```
Set-SiteMailbox -Identity "MarketingEvents 2013" -
SharePointUrl "https://myserver/teams/marketing"
```

EXAMPLE 2

This example disables the duplication of email messages in the site mailbox SMO_ContosoSales.

```
Set-SiteMailbox -Identity SMO_ContosoSales -
RemoveDuplicateMessages $true
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Site mailboxes" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *Identity* parameter specifies the identity of the site mailbox. You can use any of the following values: • Distinguished name (DN) • GUID • Name • Display name • Alias • Primary SMTP address
Active	Optional	System.Boolean	The *Active* parameter specifies whether to change the site mailbox's lifecycle status. This parameter accepts `$true`

			or `$false`. This parameter is intended for use only by the user interface. We recommend that you don't use this parameter in the Exchange Management Shell.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the display name of the site mailbox.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory.
Force	Optional	System.Management.Automation.SwitchParameter	The *Force* parameter specifies that the command should run immediately and bypass confirmation prompts.
Members	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *Members* parameter specifies the members of the site mailbox. You can add or remove members using this parameter. This is a multivalued parameter and multiple recipients should be separated by a comma. This parameter is intended for use only by the user interface. We recommend that you don't use this parameter in the Shell.
Owners	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter[]	The *Owners* parameter specifies the site mailbox's owners. This is a multivalued field and multiple recipients should be separated by commas. This parameter is intended for use only by the user interface. We recommend that you

			don't use this parameter in the Shell.
RemoveDuplicateMessages	Optional	System.Boolean	The *RemoveDuplicateMessages* parameter specifies that when users post messages to a site mailbox, duplicate messages will be deleted. This parameter accepts the values of `$true` or `$false`. The default value is `$false`.
SharePointUrl	Optional	System.Uri	The *SharePointUrl* parameter specifies the URL of the SharePoint site, for example, `"https://myserver/teams/edu"`.
ShowInMyClient	Optional	System.Boolean	The *ShowInMyClient* parameter specifies that the site mailbox folder will show in your email client. This parameter is intended for use only by the user interface. We recommend that you don't use this parameter in the Shell.
WhatIf	Optional	System.Management.Automation.SwitchPar	The *WhatIf* switch instructs the command to

		ameter	simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-SiteMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Test-SiteMailbox** cmdlet to test the site mailbox to Microsoft SharePoint connectivity and to test whether users have the correct permissions to use a site mailbox. This cmdlet should be used for troubleshooting and diagnostic purposes.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-SiteMailbox [-BypassOwnerCheck <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-SharePointUrl <Uri>] [-Confirm
[<SwitchParameter>]] [-Identity <RecipientIdParameter>] [-
```

```
RequestorIdentity <RecipientIdParameter>] [-UseAppTokenOnly
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

The example tests a SharePoint site's connectivity only. You can use this command before creating a site mailbox or if you're having a problem creating a site mailbox.

```
Test-SiteMailbox -BypassOwnerCheck -SharePointUrl "https://
myserver/teams/edu"
```

EXAMPLE 2

This example tests the Exchange server connectivity with an existing site mailbox using the *Identity* and *UseAppTokenOnly* parameters. The *Identity* parameter specifies the site mailbox, and the *UseAppTokenOnly* parameter specifies that you want to test under the identity of the Exchange server. Run this command for troubleshooting documentation synchronization issues.

```
Test-SiteMailbox -BypassOwnerCheck -Identity
mysitemailbox@contoso.com -UseAppTokenOnly
```

EXAMPLE 3

This example tests a specific user's ability to access a SharePoint site by using the *RequestorIdentity* parameter.

```
Test-SiteMailbox -BypassOwnerCheck -RequestorIdentity
"kweku@contoso.com" -SharePointUrl "https://myserver/teams/
edu"
```

Detailed Description

If you don't specify the *RequestorIdentity* parameter, the command uses the identification of the user running this command.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the"Site mailboxes" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description

BypassOwnerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassOwnerCheck* parameter specifies that the user running this command isn't the owner of the site mailbox being tested. If you're running this command and you aren't the owner of the site mailbox, you must use this parameter for the command to complete successfully. You don't have to supply a value with this parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *Identity* parameter specifies the identity of the site mailbox. You can use any of the following values: • Distinguished name (DN)

			• GUID • Name • Display name • Alias • Primary SMTP address You can't use this parameter in conjunction with the *SharePointUrl* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
RequestorIdentity	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *RequestorIdentity* parameter specifies the identity of a user for whom you want to test to make sure that they have the correct permissions to connect to the SharePoint site mailbox. If you don't specify this parameter, the command uses the identification of the user running this command. You can use any of the following values: • DN • GUID • Name • Display name • Alias • Primary SMTP address

			You can't use this parameter in conjunction with the *UseAppTokenOnly* parameter.
SharePointUrl	Optional	System.Uri	This parameter is available only in on-premises Exchange 2013. The *SharePointUrl* parameter specifies the SharePoint URL where the site mailbox is hosted, for example, `"https://myserver/teams/edu"`. You can't use this parameter in conjunction with the *Identity* parameter.
UseAppTokenOnly	Optional	System.Management.Automation.SwitchParameter	The *UseAppTokenOnly* parameter specifies that you want to test under the identity of the Exchange server. You can't use this cmdlet in conjunction with the *RequestorIdentity* parameter.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-SiteMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Update-SiteMailbox** cmdlet to trigger a Microsoft SharePoint synchronization. This command synchronizes document content membership and permissions into Microsoft Exchange. You may need to perform this action when troubleshooting document or membership synchronization issues.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-SiteMailbox -Identity <RecipientIdParameter> [-BypassOwnerCheck
<SwitchParameter>] [-FullSync <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-Server <String>] [-Confirm
[<SwitchParameter>]] [-Target <All | Document | Membership | Maintenance>]
[-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example updates the site mailbox MarketingEvents 2013. If you don't specify the target, this triggers both document and membership synchronization. Because the *BypassOwnerCheck* parameter is used, it isn't necessary to be an owner or member of the site mailbox to run this cmdlet.

```
Update-SiteMailbox -BypassOwnerCheck -Identity
"MarketingEvents 2013"
```

EXAMPLE 2

This example updates the site mailbox WinterHoliday@tailspintoys.com and performs a full synchronization. By default, the update only occurs for synchronization from the last synchronization. This is only applicable to document synchronization

```
Update-SiteMailbox -BypassOwnerCheck -Identity
WinterHoliday@tailspintoys.com -FullSync
```

Detailed Description

If you are running this command against a site mailbox in which you aren't the owner, you need to use the *BypassOwnerCheck* parameter to run this cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Site mailboxes" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *Identity* parameter specifies the identity of the site mailbox. This parameter accepts the following values: • DisplayName • SMTP address

			• Name • GUID
BypassOwnerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassOwnerCheck* parameter specifies that the user running the command in the Exchange Management Shell isn't a site mailbox owner or member. If you run the command without this parameter and you aren't the site mailbox owner or member, the command doesn't run or return any information.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
FullSync	Optional	System.Management.Automation.SwitchParameter	The *FullSync* parameter specifies full sync is expensive and will have a performance impact on the Exchange system.
Organization	Optional	Microsoft.Exchange.Co	The *Organization*

		nfiguration.Tasks.OrganizationIdParameter	parameter is reserved for internal Microsoft use.
Server	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *Server* parameter specifies the fully qualified domain name (FQDN) or the Microsoft SharePoint server on which the site mailbox is located.
Target	Optional	Microsoft.Exchange.Management.RecipientTasks.TeamMailboxDiagnosticsBase+TargetType	The *Target* parameter specifies whether to update the SharePoint documents, the site mailbox's membership list or both. This parameter accepts the following values: • `All` • `Document` • `Membership` If you don't specify this parameter when you run the cmdlet, this parameter value defaults to `ALL`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what

			changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SiteMailboxDiagnostics

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-07*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-SiteMailboxDiagnostics** cmdlet to view important event-related data for each site mailbox. This information can be used to troubleshoot site mailbox issues.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-SiteMailboxDiagnostics -Identity <RecipientIdParameter> [-
BypassOwnerCheck <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-Confirm [<SwitchParameter>]] [-SendMeEmail
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example returns the event information for the site mailbox by using its display name Marketing Events 2013.

```
Get-SiteMailboxDiagnostics -BypassOwnerCheck -Identity
"Marketing Events 2013"
```

EXAMPLE 2

This example returns the event information for the Marketing Events 2013 site mailbox and sends an email to the primary SMTP address of the user running this command.

```
Get-SiteMailboxDiagnostics -BypassOwnerCheck -Identity
events2013@contoso.com -SendMeEmail
```

Detailed Description

If you aren't a member or owner of the site mailbox that you want to view the diagnostics information for, you must use the *BypassOwnerCheck* parameter when running this cmdlet. If you aren't a member or owner of the site mailbox and you run this cmdlet without using the *BypassOwnerCheck* parameter, the command fails with an "object not found" error.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Site mailboxes" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *Identity* parameter specifies the site mailbox. You can use the following values: • Alias • Display name • *Domain\Account* • SMTP address • Distinguished name (DN) • Object GUID

			• User principal name (UPN) • LegacyExchangeDN
BypassOwnerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassOwnerCheck* switch is used by administrators who aren't members or owners of the site mailbox. If you aren't a member or owner of the site mailbox and you run this cmdlet without using the *BypassOwnerCheck* parameter, the command fails with an "object not found" error.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
SendMeEmail	Optional	System.Management.Automation.SwitchPar	The *SendMeEmail* parameter specifies that

		ameter	the diagnostic information is sent to the primary SMTP email address for whichever user is logged into Remote PowerShell.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SiteMailboxProvisioningPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-SiteMailboxProvisioningPolicy** cmdlet to view information about site mailbox provisioning policies.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-SiteMailboxProvisioningPolicy [-Identity <MailboxPolicyIdParameter>]
[-DomainController <Fqdn>] [-IgnoreDehydratedFlag <SwitchParameter>] [-
Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example returns the default information about all site mailbox provisioning policies in your organization.

```
Get-SiteMailboxProvisioningPolicy
```

EXAMPLE 2

This example returns detailed information about the provisioning policy with the identity SM_NewPolicy

```
Get-SiteMailboxProvisioningPolicy -Identity SM_NewPolicy |
Format-List
```

EXAMPLE 3

This example returns all policies in your organization, but only displays the IsDefault information to identify which policy is the default policy.

```
Get-SiteMailboxProvisioningPolicy | Format-List IsDefault
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Site mailbox provisioning policy" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the identity of the site mailbox provisioning policy.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-SiteMailboxProvisioningPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-SiteMailboxProvisioningPolicy** cmdlet to create a provisioning policy for site mailboxes that configures the send and receive quotas and allows you to set the default provisioning policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-SiteMailboxProvisioningPolicy -Name <String> [-AliasPrefix <String>]
[-Confirm [<SwitchParameter>]] [-DefaultAliasPrefixEnabled <$true |
$false>] [-DomainController <Fqdn>] [-IgnoreDehydratedFlag
<SwitchParameter>] [-IsDefault <SwitchParameter>] [-IssueWarningQuota
<ByteQuantifiedSize>] [-MaxReceiveSize <ByteQuantifiedSize>] [-
Organization <OrganizationIdParameter>] [-ProhibitSendReceiveQuota
<ByteQuantifiedSize>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the default provisioning policy SM_ProvisioningPolicy that has the following settings:

- The warning quota for the site mailboxes is 9 GB.
- The site mailboxes are prohibited from receiving messages when the mailbox size reaches 10 GB.
- The maximum size of email messages that can be sent to site mailboxes is 50 MB.

```
New-SiteMailboxProvisioningPolicy -Name
SM_ProvisioningPolicy -IsDefault -IssueWarningQuota 9GB -
ProhibitSendReceiveQuota 10GB -MaxReceiveSize 50MB
```

EXAMPLE 2

This example creates the default provisioning policy SM_DefaultPolicy that uses the defaults for send and receive quotas. If you don't explicitly specify the *IssueWarningQuota* and *ProhibitSendReceiveQuota* parameters, the command uses the default values.

```
New-SiteMailboxProvisioningPolicy -Name SM_DefaultPolicy -
IsDefault
```

EXAMPLE 3

This example creates the default provisioning policy SM_DefaultPolicy and sets the *AliasPrefix* to **Project**. When site mailboxes are created, they are prepended with the prefix **Project-**.

Note:

By default, the *DefaultAliasPrefixEnabled* parameter is set to `$true`, all on-premises site mailboxes are created with the prefix **SM-**, and all cloud-based site mailboxes are created with the prefix **SMO-**. The *AliasPrefix* parameter takes precedence over the *DefaultAliasPrefixEnabled* parameter.

```
New-SiteMailboxProvisioningPolicy -Name SM_DefaultPolicy -
IsDefault -AliasPrefix Project
```

Detailed Description

You can create multiple site mailbox provisioning policies, but only the default policy is followed when users create site mailboxes. When Exchange is installed, a site mailbox provisioning policy is created named Default. When creating a site mailbox provisioning policy, you can set the send and receive quotas. If you don't explicitly state the quotas, the cmdlet uses the default values. The following are the default values.

- *IssueWarningQuota* 4.5 gigabytes (GB)
- *ProhibitSendReceiveQuota* 5 GB
- *MaxReceiveSize* 36 megabytes (MB)

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Site mailboxes" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	This parameter is available only in on-premises Exchange 2013. The *Name* parameter specifies the name of the provisioning policy that you are creating.

AliasPrefix	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *AliasPrefix* parameter allows you to configure a custom prefix that you want added to site mailbox aliases. If the *AliasPrefix* parameter is set to a valid, non-null string, each new site mailbox will have that string prepended to the alias. If the *AliasPrefix* parameter is set to `$null` and the *DefaultAliasPrefixEnabled* parameter is set to `$true`, the default prefix is used. If the *AliasPrefix* parameter is set to `$null` and the *DefaultAliasPrefixEnabled* parameter is set to `$false`, no prefix is used.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -

			`Confirm:$False`. You must include a colon (`:`) in the syntax.
DefaultAliasPrefixEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *DefaultAliasPrefixEnabled* parameter specifies whether all new site mailboxes will have **SM-** prepended to the alias if the site mailbox is in an on-premises organization or will have **SMO-** prepended to the alias if the site mailbox is in an Exchange Online or Office365 organization. However, if a value has been specified for the *AliasPrefix* parameter, that value will be used even if this parameter is set to `$true`. For example, if the *DefaultAliasPrefixEnabled* parameter is set to `$true`, the *AliasPrefix* parameter is set to `$null`, and an on-premises site mailbox is created named

			BugBash_2013, the alias for that site mailbox will be SM-BugBash_2013. If the *AliasPrefix* parameter is `$null` and the *DefaultAliasPrefixEnabled* parameter is `$false`, no prefix will be added for new site mailboxes. This parameter accepts `$true` or `$false`. The default value is `$true`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IsDefault	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IsDefault* parameter specifies that the site

			mailbox provisioning policy is the default policy. You can have multiple policies, but only the default policy is followed when users create site mailboxes.
IssueWarningQuota	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is available only in on-premises Exchange 2013. The *IssueWarningQuota* parameter specifies the site mailbox size that triggers a warning message to the site mailbox. The default value is 4.5 GB. When you enter the value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from `1B` through `unlimited`. If you enter a value of `unlimited`, no size limit is

			imposed on the site mailbox.
MaxReceiveSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is available only in on-premises Exchange 2013. The *MaxReceiveSize* parameter specifies the maximum size of email messages that can be received by the site mailbox. The default value is 36 MB. When you enter the value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from `1B` through `unlimited`. If you enter a value of `unlimited`, no size limit is imposed on the site mailbox.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.Orga	The *Organization* parameter is reserved for

		nizationIdParameter	internal Microsoft use.
ProhibitSendReceiveQuota	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is available only in on-premises Exchange 2013. The *ProhibitSendReceiveQuota* parameter specifies the size at which the site mailbox can no longer send or receive messages. The default value is 5 GB. When you enter the value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from `1B` through `unlimited`. If you enter a value of `unlimited`, no size limit is imposed on the site mailbox.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it

			would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-SiteMailboxProvisioningPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Remove-SiteMailboxProvisioningPolicy** cmdlet to delete a site mailbox provisioning policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-SiteMailboxProvisioningPolicy -Identity <MailboxPolicyIdParameter>
[-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the site mailbox policy that was created when you installed Microsoft Exchange.

⚠**Warning:**
You must first create and designate a default policy before you can remove the policy named Default.

```
Remove-SiteMailboxProvisioningPolicy -Identity Default
```

Detailed Description

You can't delete the default site mailbox provisioning policy. You need to create a default policy by using the New-SiteMailboxProvisioningPolicy cmdlet or assign another provisioning policy as the default by using the Set-SiteMailboxProvisioningPolicy cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Site mailbox provisioning policy" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the identity of the site mailbox provisioning policy that you want to delete.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't

			have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-SiteMailboxProvisioningPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Sharing and collaboration cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-SiteMailboxProvisioningPolicy** cmdlet to modify an existing site mailbox provisioning policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-SiteMailboxProvisioningPolicy -Identity <MailboxPolicyIdParameter> [-
AliasPrefix <String>] [-Confirm [<SwitchParameter>]] [-
DefaultAliasPrefixEnabled <$true | $false>] [-DomainController <Fqdn>] [-
IgnoreDehydratedFlag <SwitchParameter>] [-IsDefault <SwitchParameter>] [-
IssueWarningQuota <ByteQuantifiedSize>] [-MaxReceiveSize
<ByteQuantifiedSize>] [-Name <String>] [-ProhibitSendReceiveQuota
<ByteQuantifiedSize>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example changes the site mailbox provisioning policy named Default to allow the maximum size of email messages that can be received by the site mailbox to 25 MB. When you install Exchange, a provisioning policy is created with the name Default.

```
Set-SiteMailboxProvisioningPolicy -Identity Default -
MaxReceiveSize 25MB
```

EXAMPLE 2

This example changes the warning quota to 9.5 GB and the prohibit send and receive quota to 10 GB.

```
Set-SiteMailboxProvisioningPolicy -Identity Default -
IssueWarningQuota 9GB -ProhibitSendReceiveQuota 10GB
```

EXAMPLE 3

This example changes the default provisioning policy SM_DefaultPolicy and sets the *AliasPrefix*

parameter to `Project`. When site mailboxes are created, they are prepended with the prefix **Project-**.

Note:
By default, the *DefaultAliasPrefixEnabled* parameter is set to `$true` and all on-premises site mailboxes are created with the prefix **SM-** and all cloud-based site mailboxes are created with the prefix **SMO-**. The *AliasPrefix* parameter takes precedence over the *DefaultAliasPrefixEnabled* parameter.

```
Set-SiteMailboxProvisioningPolicy -Identity
SM_DefaultPolicy -AliasPrefix Project
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Site mailbox provisioning policy" entry in the Sharing and collaboration permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the identity of the site mailbox provisioning policy that you want to edit.
AliasPrefix	Optional	System.String	The *AliasPrefix* parameter allows you to configure a custom prefix that you want added to site mailbox aliases. If the *AliasPrefix* parameter is set to a valid, non-null string, each new site mailbox will have that string prepended to the

			alias. If the *AliasPrefix* parameter is set to `$null` and the *DefaultAliasPrefixEnabled* parameter is set to `$true`, the default prefix will be used. If *AliasPrefix* parameter is set to `$null` and the *DefaultAliasPrefixEnabled* parameter is set to `$false`, no prefix will be used.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DefaultAliasPrefixEnabled	Optional	System.Boolean	The *DefaultAliasPrefixEnabled* parameter specifies whether all new site mailboxes will have **SM-** prepended to the alias if the site mailbox is in an on-premises organization or will have **SMO-**

			prepended to the alias if the site mailbox is in an Exchange Online or Office365 organization. However, if a value has been specified for the *AliasPrefix* parameter, that value will be used even if this parameter is set to `$true`. For example, if the *DefaultAliasPrefixEnabled* parameter is set to `$true`, the *AliasPrefix* parameter is set to `$null`, and an on-premises site mailbox is created named BugBash_2013, the alias for that site mailbox will be SM-BugBash_2013. If the *AliasPrefix* parameter is `$null` and the *DefaultAliasPrefixEnabled* parameter is `$false`, no prefix will be added for new site mailboxes. This parameter accepts `$true` or `$false`. The default value is `$true`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDehydratedFlag	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IsDefault	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IsDefault* switch specifies that the site mailbox provisioning policy is the default policy. You can have multiple policies, but only the default policy is followed when users create site mailboxes.
IssueWarningQuota	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is available only in on-premises Exchange 2013. The *IssueWarningQuota* parameter specifies the site mailbox size that triggers a warning message to the site

			mailbox. The default value is 4.5 gigabytes (GB). When you enter the value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from `1 B` to `unlimited`. If you enter a value of `unlimited`, no size limit is imposed on the site mailbox.
MaxReceiveSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is available only in on-premises Exchange 2013. The *MaxReceiveSize* parameter specifies the maximum size of email messages that can be received by the site mailbox. The default value is 36 MB. When you enter the value, qualify the value with one of the following units: • B (bytes)

			• KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from `1 B` to `unlimited`. If you enter a value of `unlimited`, no size limit is imposed on the site mailbox.
Name	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *Name* parameter allows you to change the name of the site mailbox provisioning policy.
ProhibitSendReceiveQuota	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	This parameter is available only in on-premises Exchange 2013. The *ProhibitSendReceiveQuota* parameter specifies the size at which the site mailbox can no longer send or receive messages. The default value is 5 GB. When you enter the value, qualify the value with one

			of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The valid input range for this parameter is from `1 B` to `unlimited`. If you enter a value of `unlimited`, no size limit is imposed on the site mailbox.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Unified Messaging cmdlets

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-08*

UM auto attendant cmdlets

New-UMAutoAttendant

Remove-UMAutoAttendant

Get-UMAutoAttendant

Set-UMAutoAttendant

Enable-UMAutoAttendant

Disable-UMAutoAttendant

UM call answering rules cmdlets

New-UMCallAnsweringRule

Remove-UMCallAnsweringRule

Get-UMCallAnsweringRule

Set-UMCallAnsweringRule

Enable-UMCallAnsweringRule

Disable-UMCallAnsweringRule

UM call data and summary report cmdlets

Export-UMCallDataRecord

Get-UMCallDataRecord

Get-UMCallSummaryReport

Client Access server (UM Call Router service) cmdlets

Get-UMCallRouterSettings

Set-UMCallRouterSettings

UM dial plan cmdlets

New-UMDialplan

Remove-UMDialplan

Get-UMDialplan

Set-UMDialplan

UM hunt group cmdlets

New-UMHuntGroup

Remove-UMHuntGroup

Get-UMHuntGroup

UM IP gateway cmdlets

New-UMIPGateway

Remove-UMIPGateway

Get-UMIPGateway

Set-UMIPGateway

Enable-UMIPGateway

Disable-UMIPGateway

UM mailbox cmdlets

Enable-UMMailbox

Disable-UMMailbox

Get-UMMailbox

Set-UMMailbox

UM mailbox PIN cmdlets

Get-UMMailboxPIN

Set-UMMailboxPIN

UM mailbox policy cmdlets

New-UMMailboxPolicy

Remove-UMMailboxPolicy

Get-UMMailboxPolicy

Set-UMMailboxPolicy

UM prompt management cmdlets

Import-UMPrompt

Export-UMPrompt

Mailbox server (UM service) cmdlets

Enable-UMService

Disable-UMService

Get-UMService

Set-UMService

UM troubleshooting and monitoring cmdlets

Get-UMActiveCalls

Test-UMConnectivity

Test-ExchangeUMCallFlow

Note:
You download the **Test-ExchangeUMCallFlow** cmdlet (the UM Troubleshooting Tool) from the Microsoft Download Center. For more information, see Unified Messaging Troubleshooting Tool.

Get-UMActiveCalls

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-UMActiveCalls** cmdlet to return information about the calls that are active and being processed by the Mailbox server running the Microsoft Exchange Unified Messaging service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMActiveCalls [-Server <ServerIdParameter>] <COMMON PARAMETERS>

Get-UMActiveCalls -InstanceServer <UMServer> <COMMON PARAMETERS>

Get-UMActiveCalls -DialPlan <UMDialPlanIdParameter> <COMMON PARAMETERS>

Get-UMActiveCalls -IPGateway <UMIPGatewayIdParameter> <COMMON PARAMETERS>

COMMON PARAMETERS: [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example displays the details of all active calls on the local Mailbox server.

```
Get-UMActiveCalls
```

EXAMPLE 2

This example displays the details of all active calls on the Mailbox server MyUMServer.

```
Get-UMActiveCalls -Server MyUMServer
```

EXAMPLE 3

This example displays the details of all active calls being processed by the UM IP gateway MyUMIPGateway.

```
Get-UMActiveCalls -IPGateway MyUMIPGateway
```

EXAMPLE 4

This example displays a list of active calls associated with the UM dial plan MyUMDialPlan.

```
Get-UMActiveCalls -DialPlan MyUMDialPlan
```

Detailed Description

The **Get-UMActiveCalls** cmdlet returns information about the active calls being processed. If the **Get-UMActiveCalls** cmdlet specifies either the UM dial plan or UM IP gateway, it looks in Active Directory to determine which Mailbox server running the Microsoft Exchange Unified Messaging service must be contacted. If the Mailbox server is specified at a command prompt, the **Get-UMActiveCalls** cmdlet returns the active calls being processed by the server specified.

Note:
When a Mailbox server is process cycling, the **Get-UMActiveCalls** cmdlet doesn't return a list of all calls for both the discontinued process and the active process. It returns the active calls only for the new process.

After this task is completed, you can see the list of active calls being processed by a Mailbox server.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Server (UM service)" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
DialPlan	Required	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *DialPlan* parameter specifies the UM dial plan for which you want to retrieve active calls.
InstanceServer	Required	Microsoft.Exchange.Data.Directory.Management.UMServer	The *InstanceServer* parameter specifies the Mailbox server running the Microsoft Exchange Unified Messaging service for which you want to retrieve active

			calls.
IPGateway	Required	Microsoft.Exchange.Configuration.Tasks.UMIPGatewayIdParameter	The *IPGateway* parameter specifies the UM IP gateway for which you want to retrieve active calls.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the Mailbox server running the Microsoft Exchange Unified Messaging service for which you want to retrieve active calls.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-UMAutoAttendant

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-UMAutoAttendant** cmdlet to disable an existing Unified Messaging (UM) auto attendant that's enabled.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-UMAutoAttendant -Identity <UMAutoAttendantIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the UM auto attendant MyUMAutoAttendant.

```
Disable-UMAutoAttendant -Identity MyUMAutoAttendant
```

Detailed Description

The **Disable-UMAutoAttendant** cmdlet disables an existing UM auto attendant that's currently enabled. The **Disable-UMAutoAttendant** cmdlet disables the UM auto attendant by modifying its status variable. The **Disable-UMAutoAttendant** cmdlet can't disable the UM auto attendant if it's linked or associated to the UM hunt group associated with the default UM dial plan.

After this task is completed, the UM auto attendant is disabled and won't accept incoming calls.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMAutoAttendantIdParameter	The *Identity* parameter specifies the identifier for the UM auto attendant that's being disabled.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-UMAutoAttendant

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-UMAutoAttendant** cmdlet to enable an existing Unified Messaging (UM) auto attendant that's disabled.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-UMAutoAttendant -Identity <UMAutoAttendantIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the UM auto attendant MyUMAutoAttendant.

```
Enable-UMAutoAttendant -Identity MyUMAutoAttendant
```

Detailed Description

The **Enable-UMAutoAttendant** cmdlet enables the UM auto attendant by modifying its status variable. When you create a UM auto attendant, it isn't enabled by default. For the auto attendant to answer incoming calls, you must first enable it. After this task is completed, the UM auto attendant answers incoming calls.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMAutoAttendantIdParameter	The *Identity* parameter specifies the identifier for the UM auto attendant being enabled.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value

			with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMAutoAttendant

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-UMAutoAttendant** cmdlet to retrieve the properties and the values for a Unified Messaging (UM) auto attendant.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMAutoAttendant [-Identity <UMAutoAttendantIdParameter>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>] [-
UMDialPlan <UMDialPlanIdParameter>]
```

Examples

EXAMPLE 1

This example returns a formatted list of all UM auto attendants in the Active Directory forest.

```
Get-UMAutoAttendant | Format-List
```

EXAMPLE 2

This example displays the properties of the UM auto attendant MyUMAutoAttendant.

```
Get-UMAutoAttendant -Identity MyUMAutoAttendant
```

EXAMPLE 3

This examples displays all the UM auto attendants associated with the UM dial plan MyUMDialPlan.

```
Get-UMAutoAttendant -UMDialPlan MyUMDialPlan
```

Detailed Description

The **Get-UMAutoAttendant** cmdlet retrieves the properties for a single UM auto attendant or for a list of UM auto attendants.

After this task is completed, if no parameter is supplied with the cmdlet, the cmdlet returns all UM auto attendants in the Active Directory forest. Or, if the UM dial plan ID is supplied but no name is supplied, the cmdlet returns all UM auto attendants linked to the UM dial plan.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.UMAutoAttendantIdParameter	The *Identity* parameter specifies the identifier for the UM auto attendant that's being viewed. This is the directory object ID for the UM auto attendant.

Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
UMDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies whether to display all the UM auto attendants that are associated with the UM dial plan that's specified.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-UMAutoAttendant

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-UMAutoAttendant** cmdlet to create a Unified Messaging (UM) auto attendant.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-UMAutoAttendant -Name <String> -UMDialPlan <UMDialPlanIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
DTMFFallbackAutoAttendant <UMAutoAttendantIdParameter>] [-Organization
```

```
<OrganizationIdParameter>] [-PilotIdentifierList <MultiValuedProperty>] [-
SharedUMDialPlan <SwitchParameter>] [-SpeechEnabled <$true | $false>] [-
Status <Enabled | Disabled>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the UM auto attendant MyUMAutoAttendant that can accept incoming calls using the extension number 55000 but isn't speech-enabled.

```
New-UMAutoAttendant -Name MyUMAutoAttendant -UMDialPlan
MyUMDialPlan -PilotIdentifierList 55000 -Status Enabled
```

EXAMPLE 2

This example creates the speech-enabled UM auto attendant MyUMAutoAttendant using the extension numbers 56000 and 56100 that can accept incoming calls.

```
New-UMAutoAttendant -Name MyUMAutoAttendant -UMDialPlan
MyUMDialPlan -PilotIdentifierList 56000,56100 -
SpeechEnabled $true -Status Enabled
```

Detailed Description

The **New-UMAutoAttendant** cmdlet creates one or more UM auto attendants. UM auto attendants have a forest-wide scope in the configuration container in Active Directory. When you create a UM auto attendant, the auto attendant isn’t speech-enabled or able to answer incoming calls by default. The auto attendant is linked to a single UM dial plan that contains a list of extension numbers. Linking the UM auto attendant to the UM dial plan enables the associated Mailbox servers to answer incoming calls using the UM auto attendant.

After this task is completed, a UM auto attendant is created.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the display

			name for the UM auto attendant. The display name for the UM auto attendant can contain as many as 64 characters.
UMDialPlan	Required	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the UM dial plan identifier for the UM dial plan to be associated with this UM auto attendant.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
DTMFFallbackAutoAttendant	Optional	Microsoft.Exchange.Configuration.Tasks.UMAutoAttendantIdParameter	The *DTMFFallbackAutoAttendant* parameter specifies a secondary UM auto attendant. A secondary UM auto attendant can be used only if the *SpeechEnabled* parameter is set to `$true`.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
PilotIdentifierList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *PilotIdentifierList* parameter specifies a list of one or more pilot numbers. Pilot numbers route incoming calls to Mailbox servers. The calls are then answered by the UM auto attendant.
SharedUMDialPlan	Optional	System.Management.Automation.SwitchPara	The *SharedUMDialPlan* parameter specifies

		meter	whether the auto attendant being created is linked or associated with a dial plan outside the scope of the organization. If you specify this parameter, the auto attendant can be linked with another tenant's dial plan. This parameter is used during tenant provisioning and is only used in a data center.
SpeechEnabled	Optional	System.Boolean	The *SpeechEnabled* parameter specifies whether the UM auto attendant is speech-enabled. The default value is `$true`. If this parameter is omitted, or if the value is `$false`, the UM auto attendant isn't speech-enabled.
Status	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.StatusEnum	The *Status* parameter specifies whether the UM auto attendant being created will be enabled. If this parameter isn't

			supplied, the UM auto attendant is created but left in a disabled state.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-UMAutoAttendant

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Remove-UMAutoAttendant** cmdlet to delete a Unified Messaging (UM) auto attendant.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-UMAutoAttendant -Identity <UMAutoAttendantIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the UM auto attendant MyUMAutoAttendant.

```
Remove-UMAutoAttendant -Identity MyUMAutoAttendant
```

Detailed Description

The **Remove-UMAutoAttendant** cmdlet deletes an existing UM auto attendant from Active Directory. The **Remove-UMAutoAttendant** cmdlet deletes the UM auto attendant and also deletes instances of the UM auto attendant from any associated UM dial plans. When the UM auto attendant is deleted, incoming telephone calls to the configured extensions are no longer answered by the UM auto attendant.

After this task is completed, the UM auto attendant is removed from Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMAutoAttendantIdParameter	The *Identity* parameter specifies the identifier for the UM auto attendant being deleted. This is the directory object ID for the UM auto attendant.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of

			those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-UMAutoAttendant

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-UMAutoAttendant** cmdlet to modify an existing Unified Messaging (UM) auto attendant.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-UMAutoAttendant -Identity <UMAutoAttendantIdParameter> [-
AfterHoursKeyMapping <MultiValuedProperty>] [-AfterHoursKeyMappingEnabled
<$true | $false>] [-AfterHoursMainMenuCustomPromptEnabled <$true |
$false>] [-AfterHoursMainMenuCustomPromptFilename <String>] [-
AfterHoursTransferToOperatorEnabled <$true | $false>] [-
AfterHoursWelcomeGreetingEnabled <$true | $false>] [-
AfterHoursWelcomeGreetingFilename <String>] [-AllowDialPlanSubscribers
<$true | $false>] [-AllowedInCountryOrRegionGroups <MultiValuedProperty>]
[-AllowedInternationalGroups <MultiValuedProperty>] [-AllowExtensions
<$true | $false>] [-BusinessHoursKeyMapping <MultiValuedProperty>] [-
BusinessHoursKeyMappingEnabled <$true | $false>] [-
BusinessHoursMainMenuCustomPromptEnabled <$true | $false>] [-
BusinessHoursMainMenuCustomPromptFilename <String>] [-
BusinessHoursSchedule <ScheduleInterval[]>] [-
BusinessHoursTransferToOperatorEnabled <$true | $false>] [-
BusinessHoursWelcomeGreetingEnabled <$true | $false>] [-
BusinessHoursWelcomeGreetingFilename <String>] [-BusinessLocation
<String>] [-BusinessName <String>] [-CallSomeoneEnabled <$true | $false>]
[-Confirm [<SwitchParameter>]] [-ContactAddressList
```

```
<AddressListIdParameter>] [-ContactRecipientContainer
<OrganizationalUnitIdParameter>] [-ContactScope <DialPlan |
GlobalAddressList | AddressList>] [-DefaultMailbox <MailboxIdParameter>]
[-DomainController <Fqdn>] [-DTMFFallbackAutoAttendant
<UMAutoAttendantIdParameter>] [-ForceUpgrade <SwitchParameter>] [-
ForwardCallsToDefaultMailbox <$true | $false>] [-HolidaySchedule
<MultiValuedProperty>] [-InfoAnnouncementEnabled <True | False |
Uninterruptible>] [-InfoAnnouncementFilename <String>] [-Language
<UMLanguage>] [-MatchedNameSelectionMethod <Title | Department | Location
| None | PromptForAlias | InheritFromDialPlan>] [-Name <String>] [-
NameLookupEnabled <$true | $false>] [-OperatorExtension <String>] [-
PilotIdentifierList <MultiValuedProperty>] [-SendVoiceMsgEnabled <$true |
$false>] [-SpeechEnabled <$true | $false>] [-StarOutToDialPlanEnabled
<$true | $false>] [-TimeZone <String>] [-TimeZoneName <UMTimeZone>] [-
WeekStartDay <Sunday | Monday | Tuesday | Wednesday | Thursday | Friday |
Saturday>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures the UM auto attendant MySpeechEnabledAA to fall back to the MyDTMFAA, sets the operator's extension to 50100, enables transfers to this extension number after business hours and enables a caller to press the * button on a telephone keypad to get to the Outlook Voice Access welcome greeting when a UM auto attendant menu is being played.

```
Set-UMAutoAttendant -Identity MySpeechEnabledAA -
DTMFFallbackAutoAttendant MyDTMFAA -OperatorExtension 50100
-AfterHoursTransferToOperatorEnabled $true -
StaroutToDialPlanEnabled $true
```

EXAMPLE 2

This example configures the UM auto attendant MyUMAutoAttendant that has business hours configured to be 10:45 to 13:15 (Sunday), 09:00 to 17:00 (Monday), and 09:00 to 16:30 (Saturday) and holiday times and their associated greetings configured to be "`New Year`" on January 2, 2013, and "`Building Closed for Construction`" from April 24, 2013 through April 28, 2013.

```
Set-UMAutoAttendant -Identity MyUMAutoAttendant -
BusinessHoursSchedule 0.10:45-0.13:15,1.09:00-
1.17:00,6.09:00-6.16:30 -HolidaySchedule "New
Year,newyrgrt.wav,1/2/2013","Building Closed for
Construction,construction.wav,4/24/2013,4/28/2013"
```

EXAMPLE 3

This example configures the UM auto attendant MyAutoAttendant and enables business hours key mappings so that when callers press 1, they're forwarded to another UM auto attendant named SalesAutoAttendant. When they press 2, they're forwarded to extension number 12345 for Support, and when they press 3, they're sent to another auto attendant that plays an audio file.

```
Set-UMAutoAttendant -Identity MyAutoAttendant -
BusinessHoursKeyMappingEnabled $true -
BusinessHoursKeyMapping
"1,Sales,,SalesAutoAttendant","2,Support,12345","3,Directio
ns,,,directions.wav"
```

Detailed Description

The **Set-UMAutoAttendant** cmdlet changes or modifies the settings of an existing UM auto attendant. By default, some UM auto attendant parameters are required and are created. However, after a UM auto attendant is created, not all properties for the UM auto attendant are writable. Therefore, some values for the UM auto attendant can't be changed or modified unless the UM auto attendant is deleted and a new UM auto attendant is created.

After this task is completed, the parameters and values specified are configured on the UM auto attendant.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM auto attendants" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMAutoAttendantIdParameter	The *Identity* parameter specifies the identifier for the UM auto attendant being viewed. This parameter is the directory object ID for the UM auto attendant.
AfterHoursKeyMapping	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AfterHoursKeyMapping* parameter specifies the key mappings to be used for after business hours for the UM auto attendant.

			A key mapping is defined as an entry in a table that has as many as 9 entries. The 0 key is reserved for a transfer to the operator. The following is an example for a custom table that has two entries: "Sales, 77899","Service, 78990". The default value is `disabled`.
AfterHoursKeyMappingEnabled	Optional	System.Boolean	The *AfterHoursKeyMappingEnabled* parameter specifies whether to enable or disable key mappings for after business hours for the UM auto attendant. A key mapping is defined as an entry in a table that has as many as 9 entries. The 0 key is reserved for a transfer to the operator. The following is an example for a custom table that has two entries: "Sales, 77899","Service, 78990".
AfterHoursMainMenuCustomPromptEnable	Optional	System.Boolean	The *AfterHoursMainMenuCust*

d			*omPromptEnabled* parameter specifies whether the after business hours custom main menu is enabled. The default value is `disabled`.
AfterHoursMainMenuCustomPromptFilename	Optional	System.String	The *AfterHoursMainMenuCustomPromptFilename* parameter specifies the .wav file to be used for the after business hours custom main menu prompt.
AfterHoursTransferToOperatorEnabled	Optional	System.Boolean	The *AfterHoursTransferToOperatorEnabled* parameter specifies whether to allow calls to be transferred to the operator's extension number after business hours.
AfterHoursWelcomeGreetingEnabled	Optional	System.Boolean	The *AfterHoursWelcomeGreetingEnabled* parameter specifies whether the after hours greeting is enabled. The system default audio is used if this parameter is set to `disabled`. The default value is `disabled`.

AfterHoursWelcomeGreetingFilename	Optional	System.String	The *AfterHoursWelcomeGreetingFilename* parameter specifies the .wav file to be used for the after hours greeting message.
AllowDialPlanSubscribers	Optional	System.Boolean	The *AllowDialPlanSubscribers* parameter specifies whether to allow the dial plan subscribers to dial numbers that are resolved to a subscriber in the same dial plan. The default value is `$true`.
AllowedInCountryOrRegionGroups	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AllowedInCountryOrRegionGroups* parameter specifies the list of in-country/region dial group names allowed. The names must match group names defined in the dial plan. The string must have fewer than 128 characters.
AllowedInternationalGroups	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AllowedInternationalGroups* parameter specifies the list of international dial group names allowed. The names must match group

			names defined in the dial plan. The string must have fewer than 128 characters.
AllowExtensions	Optional	System.Boolean	The *AllowExtensions* parameter specifies whether callers can make calls to extensions that have the same number of digits as the number specified on the dial plan object. The default value is `$false`.
BusinessHoursKeyMapping	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BusinessHoursKeyMapping* parameter specifies the key mappings for business hours for the UM auto attendant. A key mapping is defined as an entry in a table that has as many as 9 entries. The 0 key is reserved for a transfer to the operator. The following is an example for a custom table that has two entries: "Sales, 77899","Service, 78990". The default is `disabled`.
BusinessHoursKeyMappingEnabled	Optional	System.Boolean	The *BusinessHoursKeyMappin*

			gEnabled parameter specifies whether the custom menus for business hours are enabled or disabled. The default value is `disabled`.
BusinessHoursMainMenuCustomPromptEnabled	Optional	System.Boolean	The *BusinessHoursMainMenuCustomPromptEnabled* parameter specifies whether the business hours custom main menu prompt is enabled. The default value is `disabled`.
BusinessHoursMainMenuCustomPromptFilename	Optional	System.String	The *BusinessHoursMainMenuCustomPromptFilename* parameter specifies the .wav file to be used for the business hours custom main menu prompt.
BusinessHoursSchedule	Optional	Microsoft.Exchange.Common.ScheduleInterval[]	The *BusinessHoursSchedule* parameter specifies the hours the business is open.
BusinessHoursTransferToOperatorEnabled	Optional	System.Boolean	The *BusinessHoursTransferToOperatorEnabled* parameter specifies whether to allow call

			transfers to the operator's extension number during business hours.
BusinessHoursWelcomeGreetingEnabled	Optional	System.Boolean	The *BusinessHoursWelcomeGreetingEnabled* parameter specifies whether the custom business hours greeting is enabled. The system default audio is used if this parameter is set to `disabled`. The default value is `disabled`.
BusinessHoursWelcomeGreetingFilename	Optional	System.String	The *BusinessHoursWelcomeGreetingFilename* parameter specifies the .wav file to be used for the welcome message.
BusinessLocation	Optional	System.String	The *BusinessLocation* parameter specifies what the Mailbox server should read to the caller who selected the business location option on a UM auto attendant menu.
BusinessName	Optional	System.String	The *BusinessName* parameter specifies the name of the company or organization being used to generate the UM auto

			attendant welcome greeting for callers.
CallSomeoneEnabled	Optional	System.Boolean	The *CallSomeoneEnabled* parameter specifies whether the Call Someone feature is enabled. The default value is `$true`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ContactAddressList	Optional	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *ContactAddressList* parameter specifies the identity of the address list. If the *ContactScope* parameter is set to `AddressList`, this parameter defines the scope for directory searches.
ContactRecipientContainer	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *ContactRecipientContainer* parameter specifies the name or identity of the container used for

			directory searches.
ContactScope	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DialScopeEnum	The *ContactScope* parameter specifies the scope of the directory search given to callers when they access the UM auto attendant and specify a user's name.
DefaultMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *DefaultMailbox* parameter specifies the mailbox to be used with the *ForwardCallsToDefaultMailbox* parameter. The mailbox specified must be UM-enabled and associated with the same UM dial plan as the UM auto attendant.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
DTMFFallbackAutoAtt	Optional	Microsoft.Exchange.Co	The

endant		nfiguration.Tasks.UMAutoAttendantIdParameter	*DTMFFallbackAutoAttendant* parameter specifies the dual tone multi-frequency (DTMF) auto attendant used if the speech-enabled auto attendant is unavailable. If the *SpeechEnabled* parameter is set to `$true`, this auto attendant must have an associated DTMF auto attendant to use as the fallback auto attendant.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* switch specifies whether you're prompted for confirmation before a UM auto attendant object is upgraded.
ForwardCallsToDefaultMailbox	Optional	System.Boolean	The *ForwardCallsToDefaultMailbox* parameter specifies whether incoming calls received by the UM auto attendant are forwarded. The default is `$false`. When this is set to `$true`, all incoming calls to the UM auto attendant are transferred to the UM-enabled mailbox set by

			the *DefaultMailbox* parameter.
HolidaySchedule	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *HolidaySchedule* parameter specifies the holiday schedule for the organization. The schedule is formatted as an array of strings. Each string contains three parts: • Name, which is limited to 64 characters • File name for the audio prompt, which is in the .wav format • Day (date) of the holiday The following is an example: "Christmas, Christmas.wav, 12/25/2013".
InfoAnnouncementEnabled	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.InfoAnnouncementEnabledEnum	The *InfoAnnouncementEnabled* parameter specifies whether to enable the informational greeting. The default setting is `$true`.
InfoAnnouncementFilename	Optional	System.String	The *InfoAnnouncementFilename* parameter specifies

			the .wav file to be used for the informational announcement.
Language	Optional	Microsoft.Exchange.Data.UMLanguage	The *Language* parameter specifies the language used by the UM auto attendant. This language is selected from the list of available dial plan languages.
MatchedNameSelectionMethod	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AutoAttendantDisambiguationFieldEnum	The *MatchedNameSelectionMethod* parameter specifies the selection to use to differentiate between users who have names that match the touchtone or speech input. This setting can be set to the following: • Department • Title • Location • None • Prompt for alias • Inherited from UM dial plan
Name	Optional	System.String	The *Name* parameter specifies the display name to be used for the UM auto attendant. This name is limited to 64 characters.

NameLookupEnabled	Optional	System.Boolean	The *NameLookupEnabled* parameter specifies whether to allow callers to perform directory lookups by dialing the name or by speaking the name. This parameter can prevent callers from connecting to unknown extensions.
OperatorExtension	Optional	System.String	The *OperatorExtension* parameter specifies the extension number of the operator. If this parameter isn't specified, the dial plan operator is used. If the dial plan operator isn't specified, the feature isn't enabled.
PilotIdentifierList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *PilotIdentifierList* parameter specifies a list of one or more pilot numbers. Pilot numbers are used to route incoming calls to Mailbox servers. The calls are then answered by the UM auto attendant.
SendVoiceMsgEnabled	Optional	System.Boolean	The *SendVoiceMsgEnabled* parameter specifies whether to allow the Send

			Message feature.
SpeechEnabled	Optional	System.Boolean	The *SpeechEnabled* parameter specifies whether the auto attendant is speech-enabled. The default setting on the UM auto attendant is `$false`.
StarOutToDialPlanEnabled	Optional	System.Boolean	The *StarOutToDialPlanEnabled* parameter specifies whether a caller can press the * button on a telephone keypad to get to the Outlook Voice Access welcome greeting when a UM auto attendant menu is played. The default setting is `$false`.
TimeZone	Optional	System.String	The *Timezone* parameter specifies the time zone used with the auto attendant. The default time zone is the time zone setting on the server.
TimeZoneName	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UMTimeZone	The *TimeZoneName* parameter specifies all or part of a Microsoft Windows time zone display name. The string is

			compared to the display names in the local system registry to determine a simple *contains* match. An error is returned if the time zone name isn't correct.
WeekStartDay	Optional	System.DayOfWeek	The *WeekStartDay* parameter specifies the starting day of the week. The valid values for this parameter are `Sunday`, `Monday`, `Tuesday`, `Wednesday`, `Thursday`, `Friday`, and `Saturday`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-UMCallAnsweringRule

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-UMCallAnsweringRule** cmdlet to disable a call answering rule that has been created within a UM-enabled mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-UMCallAnsweringRule -Identity <UMCallAnsweringRuleIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Mailbox
<MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the call answering rule MyUMCallAnsweringRule in the mailbox for Tony Smith.

```
Disable -UMCallAnsweringRule -Identity
MyUMCallAnsweringRule -Mailbox tonysmith
```

EXAMPLE 2

This example uses the *WhatIf* switch to test whether the call answering rule MyUMCallAnsweringRule in the mailbox for Tony Smith is ready to be disabled and if there are any errors within the command.

```
Disable -UMCallAnsweringRule -Identity
MyUMCallAnsweringRule -Mailbox tonysmith -WhatIf
```

EXAMPLE 3

This example disables the call answering rule MyUMCallAnsweringRule in the mailbox for Tony Smith and prompts the user logged on to confirm that they're disabling the call answering rule.

```
Disable-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
-Mailbox tonysmith -Confirm
```

Detailed Description

The **Disable-UMCallAnsweringRule** cmdlet disables the call answering rule by modifying its status variable. Disabling a call answering rule prevents it from being retrieved and processed when an incoming call is received. With this cmdlet, you can disable an existing call answering rule that's enabled.

When the call answering rule is created, you should disable the call answering rule when you're setting up conditions and actions. This prevents the call answering rule from being processed when an incoming call is received until you've correctly configured the call answering rule. After this task is completed, the cmdlet sets the parameters and the values specified.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM call answering rules entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMCallAnsweringRuleIdParameter	The *Identity* parameter specifies the UM call answering rule in a UM-enabled mailbox that's to be disabled.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing

			continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the UM-enabled mailbox that contains the UM call answering rule. The default is the user's mailbox running the cmdlet.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes

			would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-UMCallAnsweringRule

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-UMCallAnsweringRule** cmdlet to enable a call answering rule that has been created within a UM-enabled mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-UMCallAnsweringRule -Identity <UMCallAnsweringRuleIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Mailbox
<MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the call answering rule MyUMCallAnsweringRule in the mailbox for Tony

Smith.

```
Enable-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
-Mailbox tonysmith
```

EXAMPLE 2

The example uses the *WhatIf* switch to test whether the call answering rule MyUMCallAnsweringRule in the mailbox for Tony Smith is ready to be enabled and if there are any errors within the command.

```
Enable-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
-Mailbox tonysmith -WhatIf
```

EXAMPLE 3

This example enables the call answering rule MyUMCallAnsweringRule in the mailbox for Tony Smith and prompts the logged-on user to confirm that the call answering rule is to be enabled.

```
Enable-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
-Mailbox tonysmith -Confirm
```

Detailed Description

The **Enable-UMCallAnsweringRule** cmdlet enables the call answering rule by modifying its status variable. When the call answering rule is created, it's enabled. This cmdlet allows you to enable a previously disabled call answering rule. Enabling a call answering rule enables the cmdlet to retrieve the call answering rule including the conditions and actions for a specified call answering rule.

After this task is completed, the cmdlet sets the parameters and values specified. When you enable a call answering rule, the call answering rule is processed when an incoming call is received.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM call answering rules" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMC	The *Identity* parameter specifies the UM call

		allAnsweringRuleIdParameter	answering rule in a UM-enabled mailbox that's to be enabled.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the UM-enabled mailbox that contains the UM call answering rule. The

			default is the user's mailbox running the cmdlet.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMCallAnsweringRule

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some

parameters and settings may be exclusive to one environment or the other.

Use the **Get-UMCallAnsweringRule** cmdlet to view the properties of a Unified Messaging (UM) call answering rule that has been created within a UM-enabled mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMCallAnsweringRule [-Identity <UMCallAnsweringRuleIdParameter>] [-
DomainController <Fqdn>] [-Mailbox <MailboxIdParameter>]
```

Examples

EXAMPLE 1

This example returns a formatted list of call answering rules in a user's UM-enabled mailbox.

```
Get-UMCallAnsweringRule -Identity MyUMCallAnsweringRule -
Mailbox tonysmith | Format-List
```

EXAMPLE 2

This example displays the properties of the call answering rule MyUMCallAnsweringRule.

```
Get-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
```

Detailed Description

The **Get-UMCallAnsweringRule** cmdlet enables you to view the properties of a call answering rule that has been created in a UM-enabled user's mailbox. It allows you to retrieve the properties for a single call answering rule or a list of call answering rules in a UM-enabled user's mailbox.

After this task is completed, the cmdlet returns the parameters and the values specified.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM call answering rules" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.UMCallAnsweringRuleIdParameter	The *Identity* parameter specifies the identifier for a call answering rule being viewed.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the UM-enabled mailbox that contains the UM call answering rule. The default is the user's mailbox running the cmdlet.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-UMCallAnsweringRule

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-UMCallAnsweringRule** cmdlet to create a call answering rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-UMCallAnsweringRule -Name <String> [-CallerIds <MultiValuedProperty>]
[-CallersCanInterruptGreeting <$true | $false>] [-CheckAutomaticReplies
<$true | $false>] [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-ExtensionsDialed <MultiValuedProperty>] [-KeyMappings
<MultiValuedProperty>] [-Mailbox <MailboxIdParameter>] [-Organization
<OrganizationIdParameter>] [-Priority <Int32>] [-ScheduleStatus <Int32>]
[-TimeOfDay <TimeOfDay>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the call answering rule MyCallAnsweringRule in the mailbox for tonysmith with the priority of 2.

```
New-UMCallAnsweringRule -Mailbox tonysmith -Name
MyCallAnsweringRule -Priority 2
```

EXAMPLE 2

This example creates the following actions on the call answering rule MyCallAnsweringRule in the mailbox for tonysmith:

- Sets the call answering rule to two caller IDs.
- Sets the priority of the call answering rule to 2.
- Sets the call answering rule to allow callers to interrupt the greeting.

```
New-UMCallAnsweringRule -Name MyCallAnsweringRule -
CallerIds "1,4255550100,,","1,4255550123,," -Priority 2 -
CallersCanInterruptGreeting $true -Mailbox tonysmith
```

EXAMPLE 3

This example creates the call answering rule MyCallAnsweringRule in the mailbox for tonysmith that sets the free/busy status to Out of Office and sets the priority to 2.

```
New-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith@contoso.com -ScheduleStatus 0x8
```

EXAMPLE 4

This example creates the call answering rule MyCallAnsweringRule in the mailbox tonysmith and performs the following actions:

Sets the priority of the call answering rule to 2.

Creates key mappings for the call answering rule.

If the caller reaches the voice mail for the user and the status of the user is set to Busy, the caller can:

- Press the 1 key and be transferred to a receptionist at extension 45678.
- Press the 2 key and the Find Me feature will be used for urgent issues and ring extension 23456 first, and then 45671.

```
New-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith -ScheduleStatus 0x4 - -KeyMappings
"1,1,Receptionist,,,,,45678,","5,2,Urgent
Issues,23456,23,45671,50,,"
```

EXAMPLE 5

This example creates the call answering rule MyCallAnsweringRule in the mailbox for tonysmith and performs the following actions:

- Sets the priority of the call answering rule to 2.
- If the caller reaches voice mail during working hours, the caller is asked to call back later.

```
New-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith -TimeOfDay "1,0,,"
```

EXAMPLE 6

This example creates the call answering rule MyCallAnsweringRule in the mailbox for tonysmith with a custom period for the time of day and performs the following actions:

- Sets the priority of the call answering rule to 2.
- If the caller reaches voice mail and the time is between 8:00 A.M. and 12:00 P.M. on Tuesday, ask the caller to call back later.

```
New-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith -TimeOfDay "3,4,8:00,12:00"
```

Detailed Description

The **New-UMCallAnsweringRule** cmdlet creates a Unified Messaging (UM) call answering rule stored in a UM-enabled user's mailbox. You can run the cmdlet and create a call answering rule of the user that's logged on or use the *Mailbox* parameter to specify the mailbox where you want the call answering rule to be created. You can use the **New-UMCallAnsweringRule** cmdlet to specify the following conditions:

- Who the incoming call is from
- Time of day
- Calendar free/busy status
- Whether automatic replies are turned on for email

You can also specify the following actions:

- Find me
- Transfer the caller to someone else
- Leave a voice message

After this task is completed, the cmdlet sets the parameters and the values specified.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM call answering rules" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name of the Unified Messaging (UM) call answering rule or Call Answering Rule ID being modified. The call answering ID or name must be unique per the user's UM-enabled mailbox. The name or ID for the call answering rule can contain up to 255 characters.
CallerIds	Optional	Microsoft.Exchange.Da	The *CallerIds* parameter

		ta.MultiValuedProperty	specifies an entry for the "If the Caller is" condition. Each entry for this parameter can contain a phone number, an Active Directory contact, a personal contact, or the personal Contacts folder. The parameter can contain 50 phone numbers or contact entries and no more than one entry for specifying the default Contacts folder. If the *CallerIds* parameter doesn't contain a condition, the condition isn't set and is ignored. The default value is `$null`.
CallersCanInterruptGreeting	Optional	System.Boolean	The *CallersCanInterruptGreeting* parameter specifies whether a caller can interrupt the voice mail greeting while it's being played. The default is `$null`.
CheckAutomaticReplies	Optional	System.Boolean	The *CheckAutomaticReplies* parameter specifies an entry for the "If My Automatic Replies are

			Enabled" condition. The default is `$false`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtensionsDialed	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionsDialed* parameter specifies an entry for the "If the Caller Dials" condition. Each entry must be unique per call answering rule. Each extension must correspond to existing

			extension numbers assigned to UM-enabled users. The default is `$null`.
KeyMappings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *KeyMappings* parameter specifies a key mapping entry for a call answering rule. The key mappings are those menu options offered to callers if the call answering rule is set to `$true`. You can configure a maximum of 10 entries. None of the defined key mappings can overlap. The default is `$null`.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the UM-enabled mailbox where the call answering rule is created. The default is the user's mailbox that's running the cmdlet.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
Priority	Optional	System.Int32	The *Priority* parameter specifies the order that the call answering rule will be evaluated against other existing call

			answering rules. Call answering rules are processed in order of increasing priority values. The priority must be unique between all call answering rules in the UM-enabled mailbox. The priority on the call answering rule must be between 1 (highest) and 9 (lowest). The default is 9.
ScheduleStatus	Optional	System.Int32	The *ScheduleStatus* parameter specifies an entry for the "If my Schedule show that I am" condition. Users can specify their free/busy status to be checked. This parameter can be set from 0 through 15 and is interpreted as a 4-bit mask that represents the calendar status including Free, Tentative, Busy, and Out of Office. The following settings can be used to set the schedule status: • None = 0x0 • Free = 0x1 • Tentative = 0x2 • Busy = 0x4

			• OutOfOffice = 0x8 The default setting is `$null`.
TimeOfDay	Optional	Microsoft.Exchange.Data.TimeOfDay	The *TimeOfDay* parameter specifies an entry for the "If the Call Arrives During" condition for the call answering rule. You can specify working hours, non-working hours, or custom hours. The default is `$null`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-UMCallAnsweringRule

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-UMCallAnsweringRule** cmdlet to remove an existing Unified Messaging (UM) call answering rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-UMCallAnsweringRule -Identity <UMCallAnsweringRuleIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Mailbox
<MailboxIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the call answering rule MyUMCallAnsweringRule from a user's mailbox. The user's mailbox is the mailbox of the user running the cmdlet.

```
Remove-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
```

EXAMPLE 2

This example removes the call answering rule MyUMCallAnsweringRule from the mailbox of tonysmith.

```
Remove-UMCallAnsweringRule -Identity MyUMCallAnsweringRule
-Mailbox tonysmith
```

Detailed Description

The **Remove-UMCallAnsweringRule** cmdlet removes an existing UM call answering rule that has been created and stored in a UM-enabled user's mailbox. When you remove an existing call answering rule, all of the remaining call answering rules are still processed in order of their priority.

After this task is completed, the cmdlet sets the parameters and the values specified.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM call answering rules" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMCallAnsweringRuleIdParameter	The *Identity* parameter specifies the identifier for a call answering rule being removed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this

			configuration change to Active Directory.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the UM-enabled mailbox where the call answering rule is changed. The default is the user's mailbox running the cmdlet.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-UMCallAnsweringRule

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-05-05*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-UMCallAnsweringRule** cmdlet to change properties of an existing UM call answering rule.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-UMCallAnsweringRule -Identity <UMCallAnsweringRuleIdParameter> [-
CallerIds <MultiValuedProperty>] [-CallersCanInterruptGreeting <$true |
$false>] [-CheckAutomaticReplies <$true | $false>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ExtensionsDialed
<MultiValuedProperty>] [-KeyMappings <MultiValuedProperty>] [-Mailbox
<MailboxIdParameter>] [-Name <String>] [-Priority <Int32>] [-
ScheduleStatus <Int32>] [-TimeOfDay <TimeOfDay>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the priority to 2 on the existing call answering rule MyCallAnsweringRule that exists in the mailbox for tonysmith.

```
Set-UMCallAnsweringRule -Mailbox tonysmith -Name
MyCallAnsweringRule -Priority 2
```

EXAMPLE 2

This example performs the following actions on the call answering rule MyCallAnsweringRule in the mailbox for tonysmith:

- Sets the call answering rule to two caller IDs.
- Sets the priority of the call answering rule to 2.
- Sets the call answering rule to allow callers to interrupt the greeting.

```
Set-UMCallAnsweringRule -Name MyCallAnsweringRule -
CallerIds "1,4255550100,,","1,4255550123,," -Priority 2 -
CallersCanInterruptGreeting $true -Mailbox tonysmith
```

EXAMPLE 3

This example changes the free/busy status to Out of Office on the call answering rule MyCallAnsweringRule in the mailbox for tonysmith and sets the priority to 2.

```
Set-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith@contoso.com -ScheduleStatus 0x8
```

EXAMPLE 4

This example performs the following actions on the call answering rule MyCallAnsweringRule in the mailbox tonysmith:

- Sets the priority of the call answering rule to 2.
- Creates key mappings for the call answering rule.
- If the caller reaches the voice mail for the user and the status of the user is set to Busy, the caller can:
- Press the 1 key and be transferred to a receptionist at extension 45678.
- Press the 2 key and the Find Me feature will be used for urgent issues and ring extension 23456 first, and then 45671.

```
Set-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith -ScheduleStatus 0x4 -KeyMappings
"1,1,Receptionist,,,,,45678,","5,2,Urgent
Issues,23456,23,45671,50,,"
```

EXAMPLE 5

This example performs the following actions on the call answering rule MyCallAnsweringRule in the mailbox for tonysmith:

- Sets the priority of the call answering rule to 2.
- If the caller reaches voice mail during working hours, the caller is asked to call back later.

```
Set-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith -TimeOfDay "1,0,,"
```

EXAMPLE 6

This example sets a custom period for the time of day on the call answering rule MyCallAnsweringRule in the mailbox for tonysmith and performs the following actions:

- Sets the priority of the call answering rule to 2.
- If the caller reaches voice mail and the time is between 8:00 A.M. and 12:00 P.M. on Tuesday, ask the caller to call back later.

```
Set-UMCallAnsweringRule -Name MyCallAnsweringRule -Priority
2 -Mailbox tonysmith -TimeOfDay "3,4,8:00,12:00"
```

Detailed Description

The *Set-UMCallAnsweringRule* cmdlet changes the properties of an existing UM call answering rule stored in a UM enabled user's mailbox. You can use the **Set-UMCallAnsweringRule** cmdlet to specify the following conditions:

- Who the incoming call is from
- Time of day
- Calendar free/busy status
- Whether automatic replies are turned on for email

You can also specify the following actions:

- Find me
- Transfer the caller to someone else
- Leave a voice message

After this task is completed, the cmdlet sets the parameters and the values specified.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM call answering rules" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMCallAnsweringRuleIdParameter	The *Identity* parameter specifies the identifier for a call answering rule being changed.
CallerIds	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *CallerIds* parameter specifies an entry for the "If the Caller is" condition. Each entry for this parameter can contain a phone number, an Active Directory contact, a personal contact, or the personal Contacts folder.

			The parameter can contain 50 phone numbers or contact entries and no more than one entry for specifying the default Contacts folder. If the *CallerIds* parameter doesn't contain a condition, the condition isn't set and is ignored. The default value is `$null`.
CallersCanInterruptGreeting	Optional	System.Boolean	The *CallersCanInterruptGreeting* parameter specifies whether a caller can interrupt the voice mail greeting while it's being played. The default is `$null`.
CheckAutomaticReplies	Optional	System.Boolean	The *CheckAutomaticReplies* parameter specifies an entry for the "If My Automatic Replies are Enabled" condition. The default is `$false`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExtensionsDialed	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionsDialed* parameter specifies an entry for the "If the Caller Dials" condition. Each entry must be unique per call answering rule. Each extension must correspond to existing extension numbers assigned to UM-enabled users. The default is `$null`.
KeyMappings	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *KeyMappings* parameter specifies a key mapping entry for a call answering rule. The key

			mappings are those menu options offered to callers if the call answering rule is set to `$true`. You can configure a maximum of 10 entries. None of the defined key mappings can overlap. The default is `$null`.
Mailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the UM-enabled mailbox where the call answering rule will be changed. The default is the user's mailbox that's running the cmdlet.
Name	Optional	System.String	The *Name* parameter specifies the name of the UM call answering rule or Call Answering Rule ID being modified. The call answering ID or name must be unique per the user's UM-enabled mailbox. The name or ID for the call answering rule can contain up to 255 characters.
Priority	Optional	System.Int32	The *Priority* parameter specified the order that the call answering rule

			will be evaluated against other existing call answering rules. Call answering rules are processed in order of increasing priority values. The priority must be unique between all call answering rules in the UM-enabled mailbox. The priority on the call answering rule must be between 1 (highest) and 9 (lowest). The default is 9.
ScheduleStatus	Optional	System.Int32	The *ScheduleStatus* parameter specifies an entry for the "If my Schedule show that I am" condition. Users can specify their free/busy status to be checked. This parameter can be set from 0 through 15 and is interpreted as a 4-bit mask that represents the calendar status including Free, Tentative, Busy, and Out of Office. The following settings can be used to set the schedule status: • None = 0x0 • Free = 0x1

			• Tentative = 0x2 • Busy = 0x4 • OutOfOffice = 0x8 The default setting is `$null`.
TimeOfDay	Optional	Microsoft.Exchange.Data.TimeOfDay	The *TimeOfDay* parameter specifies an entry for the "If the Call Arrives During" condition for the call answering rule. You can specify working hours, non-working hours, or custom hours. The default is `$null`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-UMCallDataRecord

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Export-UMCallDataRecord** cmdlet to export Unified Messaging (UM) call data records for UM dial plans and UM IP gateways for a date that you've specified.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Export-UMCallDataRecord -ClientStream <Stream> -Date <ExDateTime> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-UMDialPlan <UMDialPlanIdParameter>] [-
UMIPGateway <UMIPGatewayIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example returns all Unified Messaging call data records on February 1, 2012, and exports them to a CSV file.

```
Export-UMCallDataRecord -Date 02/01/12
```

EXAMPLE 2

This example exports all Unified Messaging call data records for February 1, 2012, for the UM dial plan MyUMDialPlan.

```
Export-UMCallDataRecord -Date 02/01/12 -UMDialPlan
MyUMDialPlan
```

Detailed Description

The **Export-UMCallDataRecord** cmdlet exports Unified Messaging call data records for a specified date to a comma-separated value (CSV) file. You can filter call data records for specific UM dial

plans or UM IP gateways. However, if you don't specify a UM IP gateway, all call data records are returned.

Note:
The **Export-UMCallDataRecord** cmdlet is available when you're using the Exchange Administration Center. You can't use the cmdlet from the Exchange Management Shell.

After this task is completed, a report is generated that contains Unified Messaging call data records.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM call data and summary reports" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
ClientStream	Required	System.IO.Stream	The *ClientStream* parameter specifies the .NET stream to use to output for the Unified Messaging call data records.
Date	Required	Microsoft.Exchange.ExchangeSystem.ExDateTime	The *Date* parameter specifies the date of Unified Messaging call data records to retrieve. If there are no call records for the date specified, the report will be empty.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
UMDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the UM dial plan to export statistics for. If you don't specify a UM dial plan, statistics include all UM dial plans in the organization.
UMIPGateway	Optional	Microsoft.Exchange.Co	The *UMIPGateway*

		nfiguration.Tasks.UMIPGatewayIdParameter	parameter specifies the UM IP gateway to export statistics for. If you don't specify a gateway, statistics include all UM IP gateways in the selected UM dial plan, or if a UM dial plan isn't selected, statistics include all UM IP gateways in the organization.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMCallDataRecord

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-UMCallDataRecord** cmdlet to display Unified Messaging (UM) call data records for a specific UM-enabled user's mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMCallDataRecord -Mailbox <MailboxIdParameter> [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example displays the UM call data records for the UM-enabled user Tony.

```
Get-UMCallDataRecord -Mailbox tony@contoso.com
```

Detailed Description

The **Get-UMCallDataRecord** cmdlet displays UM call data records for the last 90 days for a UM-enabled mailbox that you specify. Each UM call data record provides detailed information about all calls either placed to or received by the specified user. The following table details each of the properties returned:

Property Name	Description
Date	Date and time of the Mailbox server running the Microsoft Exchange Unified Messaging service that took the call in the Coordinated Universal Time (UTC) format.

Duration	Total duration of this call. For Find Me calls, this will always be zero because the call is being transferred and won't be handled by Unified Messaging any longer.
AudioCodec	Audio codec used for the call including G.711 or Group System Mobile (GSM).
DialPlan	Name of the UM dial plan handling the call.
CallType	Type of the call (localized in the user's language): • Call Answering Voice Message • Call Answering Missed Call • Auto Attendant • Subscriber Access • Fax • PlayOnPhone • Find Me • None • UnauthenticatedPilotNumber • PromptProvisioning
CallingNumber	Phone number or address of the caller.
CalledNumber	Phone number or address of the intended recipient of the call.
Gateway	Fully qualified domain name (FQDN) of the UM IP gateway handling the call.
Network MOS (NMOS)	Mean opinion score for the network performance.
NMOSDegradation	Total NMOS degradation, which is how far the NMOS reported value was from its top value for the corresponding audio codec.

PercentagePacketLoss	Percentage that reflects the average network packet loss during the call.
Jitter	Average jitter of the network.
RoundTripMilliseconds	Round trip time for Real Time Control Protocol (RTCP) statistics in milliseconds.
BurstLossDurationMilliseconds	Average duration of packet loss during bursts during the call.

After this task is completed, a report is generated that contains UM call data records for a specific UM-enabled mailbox.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM call data and summary reports" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Mailbox	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Mailbox* parameter specifies the UM-enabled mailbox that UM call data records are displayed.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMCallRouterSettings

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-UMCallRouterSettings** cmdlet to retrieve the properties for a Client Access server that runs the Microsoft Exchange Unified Messaging Call Router service and returns a list of available Client Access servers from Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMCallRouterSettings [-DomainController <Fqdn>] [-Server
<ServerIdParameter>]
```

Examples

EXAMPLE 1

This example displays a list of all the Client Access servers running the Microsoft Exchange Unified

Messaging Call Router service in the Active Directory forest.

```
Get-UMCallRouterSettings
```

EXAMPLE 2

This example displays a formatted list of properties for the Client Access server MyUMCallRouter running the Microsoft Exchange Unified Messaging Call Router service.

```
Get-UMCallRouterSettings -Server MyUMCallRouter | Format-
List
```

Detailed Description

The **Get-UMCallRouterSettings** cmdlet retrieves the properties for a Client Access server that runs the Microsoft Exchange Unified Messaging Call Router service and returns a list of available Client Access servers from Active Directory. When the cmdlet is used for a single Client Access server, it returns the UM call router properties including `MaxCalls`, `MaxFaxCalls`, and `UMDialPlans`. The properties and their values for the Client Access server are stored in the Unified Messaging section of the Exchange Server configuration object in Active Directory.

After this task is completed, the cmdlet returns the parameters and the values specified.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access Server (UM call router service)" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.Serve	The *Server* parameter specifies the name of

		rIdParameter	the Client Access server that runs the Microsoft Exchange Unified Messaging Call Router service that's viewed. This parameter specifies the directory object ID for the UM call router.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-UMCallRouterSettings

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-UMCallRouterSettings** cmdlet to set specific properties on a Client Access server running the Microsoft Exchange Unified Messaging Call Router service. This cmdlet can be used to set individual parameters for a specified Client Access server.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-UMCallRouterSettings [-Confirm [<SwitchParameter>]] [-DialPlans
<MultiValuedProperty>] [-DomainController <Fqdn>] [-ExternalHostFqdn
<UMSmartHost>] [-ExternalServiceFqdn <UMSmartHost>] [-IPAddressFamily
<IPv4Only | IPv6Only | Any>] [-IPAddressFamilyConfigurable <$true |
$false>] [-MaxCallsAllowed <Int32>] [-Server <ServerIdParameter>] [-
SipTcpListeningPort <Int32>] [-SipTlsListeningPort <Int32>] [-
UMForwardingAddressTemplate <String>] [-UMPodRedirectTemplate <String>] [-
```

```
UMStartupMode <TCP | TLS | Dual>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the following configuration on the Client Access server MyUMCallRouter:

- Adds the Client Access server to the UM SIP dial plan MySIPDialPlan.
- Enables the Microsoft Exchange Unified Messaging Call Router service on the Client Access server to accept both IPv4 and IPv6 data packets.
- Sets the maximum number of incoming voice, fax, auto attendant, and Outlook Voice Access calls to 150.
- Enables the Microsoft Exchange Unified Messaging Call Router service to start up using TLS mode.

```
Set-UMCallRouterSettings -DialPlans MySIPDialPlan -
IPAddressFamily Any -Server
MyUMCallRouter.northwindtraders.com -UMStartupMode TLS
```

EXAMPLE 2

This example removes the Client Access server UMCallRouter001 from all UM SIP dial plans.

```
Set-UMCallRouterSettings -DialPlans $null -Server
UMCallRouter001.contoso.com
```

Detailed Description

The **Set-UMCallRouterSettings** cmdlet sets specific properties on a Client Access server running the Microsoft Exchange Unified Messaging Call Router service. This cmdlet can be used to set individual Unified Messaging parameters for a specified Client Access server.

After this task is completed, the cmdlet sets the parameters and the values specified.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Client Access Server (UM call router service)" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Confirm	Optional	System.Management.A	The *Confirm* switch

		utomation.SwitchParameter	causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DialPlans	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *DialPlans* parameter specifies the dial plan used by the Microsoft Exchange Unified Messaging Call Router service on a Client Access server. The Client Access server only needs to be associated with a UM dial plan if Microsoft Office Communications Server 2007 R2, Lync Server 2010, or Lync Server 15 is used in your organization. To remove a Client Access server from a dial plan, use `$null`. The default is no dial plans assigned. To enter multiple

			values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...`. To add or remove one or more values without affecting any existing entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExternalHostFqdn	Optional	Microsoft.Exchange.Data.UMSmartHost	This parameter is reserved for internal Microsoft use.
ExternalServiceFqdn	Optional	Microsoft.Exchange.Data.UMSmartHost	This parameter is reserved for internal

			Microsoft use.
IPAddressFamily	Optional	Microsoft.Exchange.Data.Directory.IPAddressFamily	The *IPAddressFamily* parameter specifies whether the UM IP gateway will use Internet Protocol version 4 (IPv4), IPv6, or both to communicate. If set to `IPv4Only`, the UM IP gateway only uses IPv4 to communicate. If set to `IPv6Only`, the UM IP gateway only uses IPv6. If set to `Any`, IPv6 is used first, and then if necessary, it falls back to IPv4. The default is `IPv4Only`.
IPAddressFamilyConfigurable	Optional	System.Boolean	The *IPAddressFamilyConfigurable* parameter specifies whether you're able to set the *IPAddressFamily* parameter to `IPv6Only` or `Any`. The default is `$true`.
MaxCallsAllowed	Optional	System.Int32	The *MaxCallsAllowed* parameter will be removed in future

			versions of the product.
Server	Optional	Microsoft.Exchange.Configuration.Tasks.ServerIdParameter	The *Server* parameter specifies the name of the Client Access server that runs the Microsoft Exchange Unified Messaging Call Router service that will be changed. This parameter specifies the directory object ID for the Client Access server.
SipTcpListeningPort	Optional	System.Int32	The *SipTcpListeningPort* parameter specifies the TCP port used by the Client Access server running the Microsoft Exchange Unified Messaging Call Router service to receive incoming calls. This TCP port is used by a Client Access server when a UM dial plan isn't configured to use SIP Secured or Secured mode. The default is port 5060.
SipTlsListeningPort	Optional	System.Int32	The *SipTlsListeningPort* parameter specifies the

			Transport Layer Security (TLS) port used by a Client Access server running the Microsoft Exchange Unified Messaging Call Router service to receive incoming calls. This TLS port is used by a Client Access server when a UM dial plan is configured to use SIP Secured or Secured mode. The default is port 5061.
UMForwardingAddressTemplate	Optional	System.String	This parameter is reserved for internal Microsoft use.
UMPodRedirectTemplate	Optional	System.String	This parameter is reserved for internal Microsoft use.
UMStartupMode	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UMStartupMode	The *UMStartupMode* parameter specifies whether the Client Access server running the Microsoft Exchange Unified Messaging Call Router service starts up in TCP, TLS, or Dual mode. If the Client Access server isn't

			associated with any UM dial plans or is being added to UM dial plans that have different security settings, you should choose `Dual` mode. In `Dual` mode, the Client Access server can listen on ports 5060 and 5061 at the same time. If the startup mode is changed, the Microsoft Exchange Unified Messaging Call Router service must be restarted.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMCallSummaryReport

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-UMCallSummaryReport** cmdlet to return statistics about all calls received or placed by Mailbox server running the Microsoft Exchange Unified Messaging service in an organization.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMCallSummaryReport -GroupBy <Day | Month | Total> [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>] [-UMDialPlan
<UMDialPlanIdParameter>] [-UMIPGateway <UMIPGatewayIdParameter>]
```

Examples

EXAMPLE 1

This example displays the statistics for all calls received or placed by Mailbox servers in the organization.

```
Get-UMCallSummaryReport -GroupBy Total
```

EXAMPLE 2

This example displays the statistics for all calls received or placed by Mailbox servers in the organization over the last 12 months.

```
Get-UMCallSummaryReport -GroupBy Month
```

EXAMPLE 3

This example displays the statistics for all calls received or placed by Mailbox servers in the organization over the last 90 days.

```
Get-UMCallSummaryReport -GroupBy Day
```

EXAMPLE 4

This example displays the statistics for calls received or placed by Mailbox servers for the UM dial plan MyUMDialPlan.

```
Get-UMCallSummaryReport -GroupBy Month -UMDialplan
MyUMDialPlan
```

Detailed Description

The **Get-UMCallSummaryReport** cmdlet displays the aggregated statistics about all calls received or placed by Mailbox servers running the Microsoft Exchange Unified Messaging service in an organization including voice messages, missed calls, subscriber access, auto attendant, or fax calls. The data returned by running this cmdlet includes audio quality metrics for the sample calls such as the following:

Metrics	Description
Date	Date in which all calls associated with the selected UM IP gateway and UM dial plan have been grouped, as per the following: • If the *GroupBy* parameter is set to `All`, this column has the value `---`. • If the *GroupBy* parameter is set to `Month`, the date is in the format *MMM/YY* (for example, Jan/12), where *MMM* is the first three letters of the month and *YY* is the last two digits of the corresponding year. • If the *GroupBy* parameter is set to `Date`, the date is in the format *MM/DD/YY* (for example, 01/23/12), where *MM* is the corresponding two digits of the month, *DD* is the corresponding two digits of the day, and *YY* is the last two digits of the corresponding year.

Voice Message	Percentage of incoming calls answered by Unified Messaging on behalf of users in which callers left a voice message.
Missed Calls	Percentage of incoming calls answered by Unified Messaging on behalf of users in which the callers didn't leave a voice message resulting in a missed call notification.
Outlook Voice Access	Percentage of incoming calls in which users authenticate to Unified Messaging to access their email, calendars, and voice messages.
Outbound	Percentage of calls placed or transferred by Unified Messaging on behalf of authenticated or unauthenticated users, which can be one of the following: • Find Me • Play On Phone • Play On Phone Greetings
Automated Attendant	Percentage of incoming calls that were answered by auto attendants.
Fax	Percentage of incoming calls that were redirected to a fax partner.
Other	Percentage of any other incoming or placed calls that don't fall in any of the previous categories. This is provided to allow different types of calls that might be provided in the future to be accounted for as well. This category includes unauthenticated calls made to pilot numbers.
Failed Or Rejected	Percentage of calls that either failed or were rejected by the Mailbox server for that organization.

Audio Quality	Overall audio quality for the selected period of time for the organization/user using the following scale: • Greater than 4.50 = Excellent • From 3.5 through 4.49 = Good • From 2.5 through 3.49 = Average • From 1.50 through 2.49 =Poor • Up to 1.49 = Bad
Total Calls	If the UM IP gateway is selected, this is the total number of calls grouped for the selected UM IP gateway for the corresponding date. If the UM dial plan control is selected, this is the total number of calls grouped for the selected UM dial plan for the corresponding date. If the user is selected, this column has the total number of calls for the user.
Network MOS (NMOS)	Average NMOS for the specific UM dial plan or UM IP gateway.
NMOS Degradation	NMOS degradation for the specific UM dial plan or UM IP gateway.
Jitter	Average jitter for the specific UM dial plan or UM IP gateway.
Packet loss	Average packet loss for the specific UM dial plan or UM IP gateway.
Round Trip	Round trip time (in milliseconds) for the selected UM dial plan or UM IP gateway.
Burst loss Duration	Average duration of packet loss during bursts of losses for the selected UM dial plan or UM IP gateway.

Number of samples	Number of calls sampled, when calculating the averages. A sample refers to any call data record that contains at least one of the audio quality metrics.

After this task is completed, a summary report will be displayed for all calls that are processed by Mailbox servers running the Microsoft Exchange Unified Messaging service in your organization.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM call data and summary reports" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
GroupBy	Required	Microsoft.Exchange.Data.GroupBy	The *GroupBy* parameter displays the daily statistics for the last 90 days, monthly statistics for the last 12 months, or a summary of all call statistics for your Mailbox servers running the Microsoft Exchange Unified Messaging service in your organization.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that retrieves data from Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
UMDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the Unified Messaging (UM) dial plan to show statistics for. If you don't specify a dial plan, statistics are included for all dial plans in the organization.
UMIPGateway	Optional	Microsoft.Exchange.Configuration.Tasks.UMIPGatewayIdParameter	The *UMIPGateway* parameter specifies the UM IP gateway to show statistics for. If you don't specify a UM IP gateway, statistics are included for all UM IP gateways for a selected dial plan, or, if no dial plan is selected, results will be returned for all UM IP gateways in the organization.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-UMConnectivity

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Test-UMConnectivity** cmdlet to test the operation of a Mailbox server computer running the Microsoft Exchange Unified Messaging service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-UMConnectivity -Phone <String> -PIN <String> -TUILogon <$true |
$false> -UMDialPlan <UMDialPlanIdParameter> [-CertificateThumbprint
<String>] [-ListenPort <Int32>] [-MediaSecured <$true | $false>] [-
MonitoringContext <$true | $false>] [-RemotePort <Int32>] [-Secured <$true
| $false>] [-Timeout <Int32>] <COMMON PARAMETERS>


Test-UMConnectivity [-CallRouter <SwitchParameter>] [-
CertificateThumbprint <String>] [-ListenPort <Int32>] [-MediaSecured
<$true | $false>] [-MonitoringContext <$true | $false>] [-RemotePort
<Int32>] [-Secured <$true | $false>] [-Timeout <Int32>] <COMMON
PARAMETERS>


Test-UMConnectivity -Phone <String> -UMIPGateway <UMIPGatewayIdParameter>
[-CertificateThumbprint <String>] [-DiagDtmfDurationInMilisecs <Int32>] [-
DiagDtmfSequence <String>] [-DiagInitialSilenceInMilisecs <Int32>] [-
DiagInterDtmfDiffGapInMilisecs <String>] [-DiagInterDtmfGapInMilisecs
<Int32>] [-From <String>] [-ListenPort <Int32>] [-MediaSecured <$true |
$false>] [-MonitoringContext <$true | $false>] [-Secured <$true | $false>]
[-Timeout <Int32>] <COMMON PARAMETERS>


Test-UMConnectivity -TUILogonAll <$true | $false> [-CertificateThumbprint
<String>] [-ListenPort <Int32>] [-MediaSecured <$true | $false>] [-
MonitoringContext <$true | $false>] [-RemotePort <Int32>] [-Secured <$true
| $false>] [-Timeout <Int32>] <COMMON PARAMETERS>


Test-UMConnectivity -ResetPIN <$true | $false> [-MonitoringContext <$true
| $false>] <COMMON PARAMETERS>
```

```
COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example performs connectivity and operational tests on the local Mailbox server, and then displays the Voice over IP (VoIP) connectivity information.

```
Test-UMConnectivity
```

EXAMPLE 2

This example tests the ability of the local Mailbox server to use an unsecured TCP connection instead of a secured mutual TLS connection to place a call through the UM IP gateway MyUMIPGateway by using the telephone number 56780.

```
Test-UMConnectivity -UMIPGateway MyUMIPGateway -Phone 56780
-Secured $false
```

EXAMPLE 3

This example tests a SIP dial plan by using a SIP URI. This example can be used in an environment that includes Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server.

```
Test-UMConnectivity -Phone
sip:sipdp.contoso.com@contoso.com -UMIPGateway
MyUMIPGateway -Secured $true -From sip:user1@contoso.com -
MediaSecured $true
```

Detailed Description

Two diagnostic tests are designed to test the operation of the Microsoft Exchange Server 2013 Mailbox server software (mode 1) and the operation of the whole system that includes the connected telephony components (mode 2).

The **Test-UMConnectivity** cmdlet can be used to test the operation of a Mailbox server and related connected telephony equipment. When you run this cmdlet and include the *UMIPGateway* parameter, the Mailbox server tests the full end-to-end operation of the Unified Messaging system. This test includes the telephony components connected to the Mailbox server, such as IP gateways, Private Branch eXchanges (PBXs), and cabling. If the *UMIPGateway* parameter isn't specified, the Mailbox server tests only the operation of the Unified Messaging components that are installed and configured on the server.

When you run this cmdlet in an on-premises Unified Messaging deployment, you need to create a UM IP gateway object for the computer or server that the cmdlet is testing. When you create the UM IP gateway object, you must configure it with a fully qualified domain name (FQDN) and that FQDN must match the name of the computer running this cmdlet.

After this task is complete, the cmdlet will have tested the operation of the Mailbox server and related telephony components.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM server" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Phone	Required	System.String	The *Phone* parameter specifies the telephone number or Session Initiation Protocol (SIP) Uniform Resource Identifier (URI) used when the test call is redirected. The extension number should be configured in the PBX to forward calls to the UM hunt group.
PIN	Required	System.String	The *PIN* parameter specifies the PIN associated with the UM-enabled mailbox.
ResetPIN	Required	System.Boolean	The *ResetPIN* parameter specifies whether to generate or regenerate a new PIN

			for all the test mailboxes in the current site.
TUILogon	Required	System.Boolean	The *TUILogon* parameter specifies whether the cmdlet tries to log on to one or more UM-enabled mailboxes. The mailboxes must be UM-enabled and associated with the UM dial plan to which the Mailbox server running the Microsoft Exchange Unified Messaging service belongs. The default setting is `$false`.
TUILogonAll	Required	System.Boolean	The *TUILogonAll* parameter specifies whether to try to connect to all test mailboxes in the current Active Directory site. The default setting is `$false`. The accounts that are tested must be generated by calling the New-TestCasConnectivityUser.ps1 script, and the

			corresponding mailboxes must be UM-enabled. Otherwise, no action is taken.
UMDialPlan	Required	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the UM dial plan to be tested. This parameter must be used with the *TUILogon* parameter.
UMIPGateway	Required	Microsoft.Exchange.Configuration.Tasks.UMIPGatewayIdParameter	The *UMIPGateway* parameter specifies the name of the UM IP gateway or IP PBX to use for the outgoing test call.
CallRouter	Optional	System.Management.Automation.SwitchParameter	The *CallRouter* parameter specifies the Client Access server that runs the Microsoft Exchange Unified Messaging Call Router service used for testing the functionality of Unified Messaging.
CertificateThumbprint	Optional	System.String	The *CertificateThumbprint* parameter specifies the certificate thumbprint used for SIP Secured

			and Secured mode.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DiagDtmfDurationInMilisecs	Optional	System.Int32	The *DiagDtmfDurationInMilisecs* parameter specifies the duration of each digit sent.
DiagDtmfSequence	Optional	System.String	The *DiagDtmfSequence* parameter specifies the sequence of digits sent.
DiagInitialSilenceInMilisecs	Optional	System.Int32	The *DiagInitialSilenceInMilisecs* parameter specifies the time period in milliseconds that the cmdlet pauses before the digit sequence is sent.
DiagInterDtmfDiffGapInMilisecs	Optional	System.String	The *DiagInterDtmfDiffGapI*

			nMilisecs parameter specifies whether to customize the time between the digits in the diagnostic sequence. This is a comma-delimited list that can accept null entries. This should include multiple values.
DiagInterDtmfGapInMilisecs	Optional	System.Int32	The *DiagInterDtmfGapInMilisecs* parameter specifies the time in milliseconds between each digit sent in the digit sequence. This is a single value.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
From	Optional	System.String	The *From* parameter specifies the SIP URI or SIP address that the call originated from. It's used only when you use the *Phone*

			parameter. The default setting is the SIP URI specified when you use the *Phone* parameter.
ListenPort	Optional	System.Int32	The *ListenPort* parameter specifies the IP port number on which to listen. If not specified, IP port 9000 is used.
MediaSecured	Optional	System.Boolean	The *MediaSecured* parameter specifies whether to use Secure RTP or RTP (unsecured) mode.
MonitoringContext	Optional	System.Boolean	The *MonitoringContext* parameter includes or excludes the associated monitoring events and performance counters in the results. Valid input for this parameter is `$true` or `$false`. The default value is `$false`. If you specify the value `$true`, the monitoring events and performance counters are included in the command results. Typically, you

			include the monitoring events and performance counters in the results when the output is passed to Microsoft System Center Operations Manager 2007 or System Center 2012 - Operations Manager.
RemotePort	Optional	System.Int32	The *RemotePort* parameter specifies the port used for the call. If not specified, the default is port 5060 for Transmission Control Protocol (TCP) and 5061 for mutual Transport Layer Security (TLS).
Secured	Optional	System.Boolean	The *Secured* parameter specifies whether the test is run in SIP Secured mode.
Timeout	Optional	System.Int32	The *Timeout* parameter specifies the length of time in seconds to wait for the test operation to finish. The default is 600 seconds. You can't set this parameter with

			a value of less than 60 seconds. However, we recommend that you always configure this parameter with a value of 60 seconds or more. The maximum value for this parameter is 1800 seconds.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMDialplan

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-UMDialplan** cmdlet to display the properties of a single Unified Messaging (UM) dial plan or to return a list of all UM dial plans associated with Mailbox servers running the Microsoft Exchange Unified Messaging service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMDialPlan [-Identity <UMDialPlanIdParameter>] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example displays a list of all the UM dial plans in the Active Directory forest.

```
Get-UMDialplan
```

EXAMPLE 2

This example displays a formatted list of properties for the UM dial plan MyUMDialPlan.

```
Get-UMDialplan -Identity MyUMDialPlan | Format-List
```

Detailed Description

The **Get-UMDialplan** cmdlet displays all properties for a UM dial plan.

After this task is completed, when you specify the *Identity* parameter, you can view the values set. When the **Get-UMDialplan** cmdlet is run, if no parameter is supplied, the cmdlet returns all UM dial plans in the Active Directory forest.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *Identity* parameter specifies the UM dial plan ID.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-UMDialplan

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-UMDialplan** cmdlet to create a Unified Messaging (UM) dial plan to establish a link between UM IP gateways, UM hunt groups, and Mailbox servers to enable communication between Unified Messaging components.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-UMDialPlan -CountryOrRegionCode <String> -Name <String> -
NumberOfDigitsInExtension <Int32> [-AccessTelephoneNumbers
<MultiValuedProperty>] [-Confirm [<SwitchParameter>]] [-DefaultLanguage
<UMLanguage>] [-DefaultOutboundCallingLineId <String>] [-DomainController
<Fqdn>] [-FaxEnabled <$true | $false>] [-GenerateUMMailboxPolicy <$true |
$false>] [-GlobalCallRoutingScheme <None | E164 | GatewayGuid | Reserved1
| Reserved2 | Reserved3>] [-Organization <OrganizationIdParameter>] [-
SipResourceIdentifierRequired <$true | $false>] [-SubscriberType
<Enterprise | Consumer>] [-URIType <TelExtn | E164 | SipName>] [-
VoIPSecurity <SIPSecured | Unsecured | Secured>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the UM dial plan MyUMDialPlan that uses four-digit extension numbers.

```
New-UMDialplan -Name MyUMDialPlan -
NumberOfDigitsInExtension 4
```

EXAMPLE 2

This example creates the UM dial plan MyUMDialPlan that uses five-digit extension numbers that support SIP URIs.

```
New-UMDialplan -Name MyUMDialPlan -URIType SipName -
NumberOfDigitsInExtension 5
```

EXAMPLE 3

This example creates the unsecured UM dial plan MyUMDialPlan that supports E.164 numbers and that uses five-digit extension numbers.

```
New-UMDialplan -Name MyUMDialPlan -URIType E164 -
NumberOfDigitsInExtension 5 -VoIPSecurity Unsecured
```

Detailed Description

The **New-UMDialplan** cmdlet creates a UM dial plan in Active Directory. A UM dial plan object has an organization-wide scope and contains all configuration information related to a telephony dial plan. A UM dial plan is a required component for establishing Unified Messaging communications with Microsoft Exchange Server 2013. When you create a UM dial plan, an understanding of telephony configurations and the implications of adding to or modifying a UM configuration is required.

Note:
After the new UM dial plan is created, a UM IP gateway and a Mailbox server must be associated with the UM dial plan to enable Unified Messaging operations.

After this task is completed, a new UM dial plan is created.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
CountryOrRegionCode	Required	System.String	The *CountryOrRegionCode* parameter specifies the country or region code that precedes a telephone number used to place calls from other countries or regions to the country or region in which the UM dial plan is located. For example, 1 is the code used for North

			America, and 44 is the code used for the United Kingdom.
Name	Required	System.String	The *Name* parameter specifies the display name of the UM dial plan. This description is presented to the administrator when a user is enabled for Unified Messaging. The UM dial plan name field can contain as many as 64 characters.
NumberOfDigitsInExtension	Required	System.Int32	The *NumberOfDigitsInExtension* parameter specifies the fixed number of digits in an extension number. The range for this parameter is from 1 through 20 digits.
AccessTelephoneNumbers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AccessTelephoneNumbers* parameter specifies the telephone number or numbers used for subscriber access. These numbers are sometimes referred to

			as pilot or pilot ID numbers. The telephone number is limited to 32 characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DefaultLanguage	Optional	Microsoft.Exchange.Data.UMLanguage	The *DefaultLanguage* parameter specifies the default language of the system. This default language is selected from the list of available languages. If there are no other UM language packs installed, the default value is en-US.
DefaultOutboundCallingLineId	Optional	System.String	The *DefaultOutboundCallingLineId* parameter specifies the phone

			number that a Mailbox server would use as the calling line ID when placing an outbound call. By default, this is set to `$null` and only the extension number of the UM-enabled user that places the outbound call is used. This parameter is reserved for internal Microsoft use.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
FaxEnabled	Optional	System.Boolean	The *FaxEnabled* parameter specifies whether the Mailbox servers associated with the dial plan answer and process incoming fax calls. The default is

			`$true`.
GenerateUMMailboxPolicy	Optional	System.Boolean	The *GenerateUMMailboxPolicy* parameter specifies whether a default UM mailbox policy is created when the UM dial plan is created. The default setting is to create a UM mailbox when the UM dial plan is created.
GlobalCallRoutingScheme	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UMGlobalCallRoutingScheme	The *GlobalCallRoutingScheme* parameter specifies whether UM-enabled users and auto attendant numbers should be included in the global routing database. If the setting is `E164`, the numbers are provisioned in the global routing database.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
SipResourceIdentifierRequired	Optional	System.Boolean	The *SipResourceIdentifierRe*

			quired parameter specifies whether the Session Initiation Protocol (SIP) resource identifier is required to be specified when mailboxes are UM-enabled and associated with the dial plan. The default is `$false` but it can only be set to `$true` if the Uniform Resource Identifier (URI) type of the dial plan is E.164.
SubscriberType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UMSubscriberType	The *SubscriberType* parameter specifies either `Consumer` or `Enterprise` as the type of dial plan. Enterprise dial plans are most likely to be used in a single organization. Consumer dial plans are used in hosted environments and can represent dial plans that may belong to different tenants.
URIType	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UMUriType	The *URIType* parameter specifies the URI type to be sent and received

			with SIP messages from the Private Branch eXchange (PBX).
VoIPSecurity	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UMVoIPSecurityType	This parameter is available only in on-premises Exchange 2013. The *VoIPSecurity* parameter specifies whether the signaling channel is encrypted using mutual Transport Layer Security (TLS). The default setting is `Unsecured`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-UMDialplan

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-UMDialplan** cmdlet to delete an existing Unified Messaging (UM) dial plan.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-UMDialPlan -Identity <UMDialPlanIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the existing UM dial plan MyUMDialPlan.

```
Remove-UMDialplan -Identity MyUMDialPlan
```

Detailed Description

The **Remove-UMDialplan** cmdlet deletes an existing UM dial plan from Active Directory. Make sure the UM dial plan isn't being used by other UM objects such as UM mailbox policies or UM IP gateways. When you delete an existing UM dial plan, the cmdlet verifies that the specified UM dial plan isn't referenced by a Mailbox server, UM IP gateway, or UM mailbox policies. The only benefit gained from deleting an obsolete UM dial plan is to reuse the name or perform general Active Directory housekeeping.

After this task is completed, the UM dial plan is removed from Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *Identity* parameter specifies the identifier for the UM dial plan being deleted. This is the directory object ID for the UM dial plan.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-UMDialplan

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-UMDialplan** cmdlet to set various properties on a Unified Messaging (UM) dial plan.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-UMDialPlan -Identity <UMDialPlanIdParameter> [-AccessTelephoneNumbers
<MultiValuedProperty>] [-AllowDialPlanSubscribers <$true | $false>] [-
AllowedInCountryOrRegionGroups <MultiValuedProperty>] [-
AllowedInternationalGroups <MultiValuedProperty>] [-AllowExtensions <$true
| $false>] [-AllowHeuristicADCallingLineIdResolution <$true | $false>] [-
AudioCodec <G711 | Wma | Gsm | Mp3>] [-AutomaticSpeechRecognitionEnabled
<$true | $false>] [-CallAnsweringRulesEnabled <$true | $false>] [-
CallSomeoneEnabled <$true | $false>] [-ConfiguredInCountryOrRegionGroups
<MultiValuedProperty>] [-ConfiguredInternationalGroups
<MultiValuedProperty>] [-Confirm [<SwitchParameter>]] [-ContactAddressList
<AddressListIdParameter>] [-ContactRecipientContainer
<OrganizationalUnitIdParameter>] [-ContactScope <DialPlan |
GlobalAddressList | Extension | AutoAttendantLink | AddressList>] [-
CountryOrRegionCode <String>] [-DefaultLanguage <UMLanguage>] [-
DefaultOutboundCallingLineId <String>] [-DialByNamePrimary <LastFirst |
FirstLast | SMTPAddress>] [-DialByNameSecondary <LastFirst | FirstLast |
SMTPAddress | None>] [-DomainController <Fqdn>] [-
EquivalentDialPlanPhoneContexts <MultiValuedProperty>] [-Extension
<String>] [-FaxEnabled <$true | $false>] [-ForceUpgrade <SwitchParameter>]
[-InCountryOrRegionNumberFormat <NumberFormat>] [-InfoAnnouncementEnabled
<True | False | Uninterruptible>] [-InfoAnnouncementFilename <String>] [-
InputFailuresBeforeDisconnect <Int32>] [-InternationalAccessCode <String>]
[-InternationalNumberFormat <NumberFormat>] [-LegacyPromptPublishingPoint
<String>] [-LogonFailuresBeforeDisconnect <Int32>] [-
MatchedNameSelectionMethod <Title | Department | Location | None |
PromptForAlias>] [-MaxCallDuration <Int32>] [-MaxRecordingDuration
<Int32>] [-Name <String>] [-NationalNumberPrefix <String>] [-
NumberingPlanFormats <MultiValuedProperty>] [-OperatorExtension <String>]
[-OutsideLineAccessCode <String>] [-PilotIdentifierList
<MultiValuedProperty>] [-RecordingIdleTimeout <Int32>] [-
SendVoiceMsgEnabled <$true | $false>] [-TUIPromptEditingEnabled <$true |
$false>] [-UMAutoAttendant <UMAutoAttendantIdParameter>] [-VoIPSecurity
<SIPSecured | Unsecured | Secured>] [-WelcomeGreetingEnabled <$true |
$false>] [-WelcomeGreetingFilename <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures the UM dial plan MyDialPlan to use 9 for the outside line access code.

```
Set-UMDialplan -Identity MyDialPlan -OutsideLineAccessCode
9
```

EXAMPLE 2

This example configures the UM dial plan MyDialPlan to use a welcome greeting.

```
Set-UMDialplan -Identity MyDialPlan -WelcomeGreetingEnabled
$true -WelcomeGreetingFilename welcome.wav
```

EXAMPLE 3

This example configures the UM dial plan MyDialPlan with dialing rules.

```
$csv=import-csv "C:\MyInCountryGroups.csv"
Set-UMDialPlan -Identity MyDialPlan -
ConfiguredInCountryOrRegionGroups $csv
Set-UMDialPlan -Identity MyDialPlan -
AllowedInCountryOrRegionGroups "local, long distance"
```

Detailed Description

The **Set-UMDialplan** cmdlet changes or modifies the properties of an existing UM dial plan. Some UM dial plan properties are required and are created by default. However, in some cases, after the UM dial plan is created, not all properties for the UM dial plan are writable. Therefore, some of the properties can't be changed unless the existing UM dial plan is deleted and a new one is created.

Important:
UM dial plans are important to the operation of Unified Messaging. Modifications to an existing UM dial plan should be performed by an administrator who understands the implications of changes to UM dial plans.

After this task is completed, the parameters and values specified are configured on the UM dial plan.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM dial plans" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *Identity* parameter specifies the UM dial plan ID. This parameter is the directory object identifier for the UM dial plan. This parameter is used to link mailboxes and Mailbox and Client Access servers to dial plans.
AccessTelephoneNum	Optional	Microsoft.Exchange.Da	The

bers		ta.MultiValuedProperty	*AccessTelephoneNumbers* parameter specifies a single valid voice mail pilot number or a list of valid voice mail pilot numbers. This list is presented to you when a user is being enabled for Unified Messaging.
AllowDialPlanSubscribers	Optional	System.Boolean	The *AllowDialPlanSubscribers* parameter specifies whether to allow subscribers dial numbers that resolve to a subscriber within the same dial plan. The default value is `$true`.
AllowedInCountryOrRegionGroups	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AllowedInCountryOrRegionGroups* parameter specifies the list of in-country/region names from the same dial group that can be dialed. The name of the allowed in-country/region group must match the group name specified in the UM dial plan.
AllowedInternational	Optional	Microsoft.Exchange.Da	The

Groups		ta.MultiValuedProperty	*AllowedInternationalGroups* parameter specifies the list of international dial group names allowed. The international dial group name must match the group name specified in the dial plan.
AllowExtensions	Optional	System.Boolean	The *AllowExtensions* parameter specifies whether to allow calls to dial plan extensions. The default value is `$false`.
AllowHeuristicADCallingLineIdResolution	Optional	System.Boolean	The *AllowHeuristicADCallingLineIdResolution* parameter specifies whether to allow calling line ID resolution using telephone number fields that may be configured in Active Directory. When this parameter is set to `$true`, the telephone numbers such as those defined in the **Mobile** or **Home** telephone number fields in Active Directory are used. Setting this parameter to `$true` allows for resolution of calling IDs for both UM-enabled

			and non-UM-enabled users. The default is `$true`. You may want to set this parameter to `$false` if the telephone numbers for users aren't in a standard format. If the telephone numbers aren't in a standard format, the Mailbox server may not be able to correctly resolve the caller ID to a name of a user in a consistent manner.
AudioCodec	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AudioCodecEnum	The *AudioCodec* parameter specifies the audio codec used for recording. `Mp3` is the default setting.
AutomaticSpeechRecognitionEnabled	Optional	System.Boolean	The *AutomaticSpeechRecognitionEnabled* parameter specifies whether Automatic Speech Recognition (ASR) is enabled for users who are members of the dial plan.
CallAnsweringRulesEnabled	Optional	System.Boolean	The *CallAnsweringRulesEnabled* parameter specifies whether Call Answering

			Rules are enabled for UM-enabled users associated with the UM dial plan.
CallSomeoneEnabled	Optional	System.Boolean	The *CallSomeoneEnabled* parameter specifies whether the Call Someone feature is enabled.
ConfiguredInCountryOrRegionGroups	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConfiguredInCountryOrRegionGroups* parameter specifies the in-country groups that can be used. Each string consists of four parts: • Group name (up to 32 characters) • AllowedNumberString • DialNumberString • TextComment
ConfiguredInternationalGroups	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConfiguredInternationalGroups* parameter specifies the international groups that can be used. Each string consists of four parts: • Group name (up to 32 characters) • AllowedNumberString • DialNumberString • TextComment

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
ContactAddressList	Optional	Microsoft.Exchange.Configuration.Tasks.AddressListIdParameter	The *ContactAddressList* parameter specifies the identity of the address list. If the *ContactScope* parameter is set to `AddressList`, this parameter defines the scope for directory searches.
ContactRecipientContainer	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *ContactRecipientContainer* parameter specifies the name or identity of the container used for directory searches.
ContactScope	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.CallSomeoneScopeEnum	The *ContactScope* parameter specifies the scope of the directory search provided to callers when they access the UM dial plan and specify a

			user's name.
CountryOrRegionCode	Optional	System.String	The *CountryOrRegionCode* parameter specifies the country or region code that precedes a telephone number used to place calls from other countries or regions to the country or region in which the UM dial plan is located. For example, 1 is the code used for North America, and 44 is the code used for the United Kingdom.
DefaultLanguage	Optional	Microsoft.Exchange.Data.UMLanguage	The *DefaultLanguage* parameter specifies the default language of the system. This default language is selected from the list of available languages. The default value is `U.S. English`.
DefaultOutboundCallingLineId	Optional	System.String	The *DefaultOutboundCallingLineId* parameter specifies the phone number that a Mailbox server would use as the calling line ID when placing an outbound call. By default, this is set to

			`$null` and only the extension number of the UM-enabled user that places the outbound call is used. This parameter is reserved for internal Microsoft use.
DialByNamePrimary	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DialByNamePrimaryEnum	The *DialByNamePrimary* parameter specifies that the Dial by Name lookup key is to be created from the specified source. The default value is `LastFirst`.
DialByNameSecondary	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DialByNameSecondaryEnum	The *DialByNameSecondary* parameter specifies that the secondary Dial by Name lookup key is to be created from the specified source. The default value is `SMTPAddress`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active

			Directory.
EquivalentDialPlanPhoneContexts	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *EquivalentDialPlanPhoneContexts* parameter specifies the name of an equivalency dial plan. This parameter can be used when two UM dial plans exist but are in different forests or when a Private Branch eXchange (PBX) numbering plan spans two UM dial plans. Adding the name of the equivalency dial plan allows name lookups using a caller ID to search in the user's dial plan but then also search for a name for the calling line ID in any equivalent dial plans that are configured.
Extension	Optional	System.String	The *Extension* parameter specifies the extension number used by the Call Someone feature when a call is transferred.
FaxEnabled	Optional	System.Boolean	The *FaxEnabled* parameter specifies whether the Mailbox servers associated with

			the UM dial plan answers and processes incoming fax calls. The default value is `$true`.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* switch specifies whether you're prompted for confirmation before a UM dial plan object is upgraded.
InCountryOrRegionNumberFormat	Optional	Microsoft.Exchange.Data.NumberFormat	The *InCountryOrRegionNumberFormat* parameter specifies the prefix string to use and the number of digits to take from the directory. This number is used when dialing into this dial plan from inside the same country or region code.
InfoAnnouncementEnabled	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.InfoAnnouncementEnabledEnum	The *InfoAnnouncementEnabled* parameter specifies whether an informational announcement is enabled. This parameter can be set to `True`, `False`, or `Uninterruptible`. The default value is `False`.
InfoAnnouncementFil	Optional	System.String	The

ename			*InfoAnnouncementFilename* parameter specifies the audio file name for an informational announcement.
InputFailuresBeforeDisconnect	Optional	System.Int32	The *InputFailuresBeforeDisconnect* parameter specifies the number of sequential user input errors allowed before the call is disconnected. The default value is 3.
InternationalAccessCode	Optional	System.String	The *InternationalAccessCode* parameter specifies the code that precedes a telephone number to dial international calls. For example, 011 is the code used to call the United States.
InternationalNumberFormat	Optional	Microsoft.Exchange.Data.NumberFormat	The *InternationalNumberFormat* parameter specifies the prefix string to use and the number of digits to take from the directory, when dialing into this dial plan from a different country code.

LegacyPromptPublishingPoint	Optional	System.String	The *LegacyPromptPublishingPoint* parameter specifies the location of the prompt publishing point for Microsoft Exchange Server 2007 Unified Messaging servers. In coexistence scenarios, this parameter is used when Exchange 2007 and Exchange Server 2010 Unified Messaging servers are added to the same Exchange 2010 UM dial plan.
LogonFailuresBeforeDisconnect	Optional	System.Int32	The *LogonFailuresBeforeDisconnect* parameter specifies the number of sequential unsuccessful logon attempts that can be made before the call is disconnected. The default value is 3.
MatchedNameSelectionMethod	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DisambiguationFieldEnum	The *MatchedNameSelectionMethod* parameter specifies the selection to use to differentiate between users who have names that match the touchtone

			or speech input. This setting can be set to the following: • `Title` • `Department` • `Location` • `None` • `PromptForAlias`
MaxCallDuration	Optional	System.Int32	The *MaxCallDuration* parameter specifies the maximum length of time that a call can last before it's interrupted and the call is dropped. The default value is `30` minutes.
MaxRecordingDuration	Optional	System.Int32	The *MaxRecordingDuration* parameter specifies the maximum length of time that messages can be recorded. This includes all kinds of calls. The default is `20` minutes. The value of this setting can be from 1 through 100. Setting this value too low can cause long voice messages to be disconnected before they are completed. Setting this value too high lets users save lengthy voice messages in their Inboxes.
Name	Optional	System.String	The *Name* parameter

			specifies the display name to use for the UM dial plan. This name is limited to 64 characters.
NationalNumberPrefix	Optional	System.String	The *NationalNumberPrefix* parameter specifies the dialing code that precedes a telephone number when placing calls from one local area to another within a specific country or region. For example, 1 is the code used within North America, and 0 is the code used within the United Kingdom.
NumberingPlanFormats	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *NumberingPlanFormats* parameter specifies one or more phone number masks that can be used for resolving caller ID to names of users in Active Directory.
OperatorExtension	Optional	System.String	The *OperatorExtension* parameter specifies the extension number of the operator. If this parameter isn't specified, the Do Not Allow Transfer to the

			Operator feature is unavailable.
OutsideLineAccessCode	Optional	System.String	The *OutsideLineAccessCode* parameter specifies the code that precedes a telephone number to dial an external in-country telephone number. This code is also referred to as a trunk access code. The default value is 9.
PilotIdentifierList	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *PilotIdentifierList* parameter specifies the pilot numbers configured on the dial plan. A single dial plan can have multiple pilot numbers. The pilot numbers must be in the E.164 format.
RecordingIdleTimeout	Optional	System.Int32	The *RecordingIdleTimeout* parameter specifies the length of time that a caller can be silent when recording a voice message before the recording is ended. The default value is 5 seconds.
SendVoiceMsgEnabled	Optional	System.Boolean	The *SendVoiceMsgEnabled* parameter specifies

			whether the Send Message feature is enabled.
TUIPromptEditingEnabled	Optional	System.Boolean	The *TUIPromptEditingEnabled* parameter specifies whether authorized users are permitted to record UM dial plan or automated attendant prompts by using the Telephone User Interface (TUI). The default setting is `$false`.
UMAutoAttendant	Optional	Microsoft.Exchange.Configuration.Tasks.UMAutoAttendantIdParameter	The *UMAutoAttendant* parameter specifies the auto attendant run when the caller presses the star (*) key. If this parameter is specified, it overrides the Call Someone feature.
VoIPSecurity	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UMVoIPSecurityType	This parameter is available only in on-premises Exchange 2013. The *VoIPSecurity* parameter specifies whether the Voice over IP (VoIP) traffic is encrypted or that the signaling channel or the signaling and the media channels

			are encrypted by using mutual Transport Layer Security (TLS). The default setting is `Unsecured`.
WelcomeGreetingEnabled	Optional	System.Boolean	The *WelcomeGreetingEnabled* parameter specifies whether a custom welcome greeting is enabled. The default value is `$false`.
WelcomeGreetingFilename	Optional	System.String	The *WelcomeGreetingFilename* parameter specifies the audio file name for the welcome greeting.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input

Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMHuntGroup

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-UMHuntGroup** cmdlet to display the properties and values for an existing Unified Messaging (UM) hunt group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMHuntGroup [-Identity <UMHuntGroupIdParameter>] [-DomainController
<Fqdn>] [-Organization <OrganizationIdParameter>] [-UMDialPlan
<UMDialPlanIdParameter>]
```

Examples

EXAMPLE 1

This example displays all the UM hunt groups in the Active Directory forest.

```
Get-UMHuntGroup
```

EXAMPLE 2

This example displays the details of the UM hunt group MyUMHuntGroup in a formatted list.

```
Get-UMHuntGroup -Identity MyUMIPGateway\MyUMHuntGroup |
Format-List
```

EXAMPLE 3

This example displays all of the UM hunt groups associated with the UM dial plan MyUMDialPlan.

```
Get-UMHuntGroup -UMDialPlan MyUMDialPlan
```

Detailed Description

The **Get-UMHuntGroup** cmdlet retrieves the properties for a single UM hunt group or a list of UM hunt groups. When you're using the **Get-UMHuntGroup** cmdlet, you can't only enter the name of the UM hunt group. You must also include the name of the UM IP gateway associated with the UM hunt group, for example, `Get-UMHuntGroup -Identity MyUMIPGateway\MyUMHuntGroup1`.

After this task is completed, if the *Identity* parameter is specified, the properties for the UM hunt group are returned. If neither the *Identity* nor the *UMDialPlan* parameter is specified, all UM hunt groups in the Active Directory forest are returned.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM hunt groups" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.UMHuntGroupIdParameter	The *Identity* parameter specifies the identifier for the UM hunt group being viewed. This is the directory object ID

			for the UM hunt group.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
UMDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the UM dial plan associated with a UM hunt group.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-UMHuntGroup

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-UMHuntGroup** cmdlet to create a Unified Messaging (UM) hunt group used to link incoming calls to a specific UM dial plan.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-UMHuntGroup -Name <String> -UMDialPlan <UMDialPlanIdParameter> -
UMIPGateway <UMIPGatewayIdParameter> [-Confirm [<SwitchParameter>]] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>] [-
PilotIdentifier <String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the UM hunt group MyUMHuntGroup that has a pilot identifier of 12345.

```
New-UMHuntGroup -Name MyUMHuntGroup -PilotIdentifier 12345
-UMDialPlan MyUMDialPlan -UMIPGateway MyUMIPGateway
```

EXAMPLE 2

This example creates the UM hunt group MyUMHuntGroup that has multiple pilot identifiers.

```
New-UMHuntGroup -Name MyUMHuntGroup -PilotIdentifier
5551234,55555 -UMDialPlan MyUMDialPlan -UMIPGateway
MyUMIPGateway
```

Detailed Description

The **New-UMHuntGroup** cmdlet creates a UM hunt group in Active Directory. Running this cmdlet enables all Mailbox servers associated with UM dial plans to communicate with an IP gateway. A UM hunt group must be created to allow communication between a UM IP gateway and a UM dial plan.

After this task is completed, a new UM hunt group is created.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM hunt groups" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the UM hunt group name used for display purposes. This string can contain as many as 64 characters, and it must be unique.

UMDialPlan	Required	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the UM dial plan used with the UM hunt group.
UMIPGateway	Required	Microsoft.Exchange.Configuration.Tasks.UMIPGatewayIdParameter	The *UMIPGateway* parameter specifies the UM IP gateway to be associated with the UM hunt group.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change

			to Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
PilotIdentifier	Optional	System.String	The *PilotIdentifier* parameter specifies the number string used to uniquely identify the pilot access number for the specified IP gateway. This number must match the subscriber access number configured in the UM dial plan.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-UMHuntGroup

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-UMHuntGroup** cmdlet to remove and delete an existing Unified Messaging (UM) hunt group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-UMHuntGroup -Identity <UMHuntGroupIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the UM hunt group MyUMHuntGroup.

```
Remove-UMHuntGroup -Identity MyUMHuntGroup
```

Detailed Description

The **Remove-UMHuntGroup** cmdlet deletes an existing UM hunt group from Active Directory. When the **Remove-UMHuntGroup** cmdlet is used, the UM hunt group is removed from the UM IP gateway, and then deleted from Active Directory. If the operation leaves the UM IP gateway without any remaining configured UM hunt groups, the IP gateway can't handle or process Unified Messaging calls.

After this task is completed, the UM hunt group is removed from Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM hunt groups" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMHuntGroupIdParameter	The *Identity* parameter specifies the identifier for the UM hunt group being deleted. This is the directory object ID for the UM hunt group object.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-UMIPGateway

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-UMIPGateway** cmdlet to disable a Unified Messaging (UM) IP gateway.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-UMIPGateway -Identity <UMIPGatewayIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Immediate <$true |
$false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables the UM IP gateway MyUMIPGateway and stops it from accepting incoming calls from the IP gateway.

```
Disable-UMIPGateway -Identity MyUMIPGateway
```

EXAMPLE 2

This example disables the UM IP gateway MyUMIPGateway and disconnects all current calls immediately.

```
Disable-UMIPGateway -Identity MyUMIPGateway -Immediate
$true
```

Detailed Description

The status variable for a UM IP gateway can be used to enable or disable call answering destined for the IP gateway. The **Disable-UMIPGateway** cmdlet disables a UM IP gateway in Active Directory by modifying its status variable. After this task is completed, the UM IP gateway no longer answers incoming calls or makes outgoing calls.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.UMIPGatewayIdParameter	specifies the identifier for the UM IP gateway being disabled. This is the directory object ID for the UM IP gateway.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Immediate	Optional	System.Boolean	The *Immediate* parameter specifies whether the Mailbox

			server running the Microsoft Exchange Unified Messaging service drops incoming calls associated with this UM IP gateway immediately or waits for the current calls to finish processing.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-UMIPGateway

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-UMIPGateway** cmdlet to enable a Unified Messaging (UM) IP gateway.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-UMIPGateway -Identity <UMIPGatewayIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the UM IP gateway MyUMIPGateway.

```
Enable-UMIPGateway -Identity MyUMIPGateway
```

Detailed Description

The status variable for a UM IP gateway can be used to enable or disable call answering destined for the IP gateway. The **Enable-UMIPGateway** cmdlet enables a UM IP gateway in Active Directory by modifying its status variable.

After this task is completed, the UM IP gateway answers incoming calls and makes outgoing calls through the IP gateway or IP Private Branch eXchange (PBX).

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.UMIPGatewayIdParameter	specifies the identifier for the UM IP gateway being enabled. This parameter is the directory object ID for the UM IP gateway.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchPara	The *WhatIf* switch instructs the command

		meter	to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMIPGateway

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-UMIPGateway** cmdlet to return a list of properties and values for a specified Unified Messaging (UM) IP gateway or a list of UM IP gateways.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMIPGateway [-Identity <UMIPGatewayIdParameter>] [-DomainController
<Fqdn>] [-IncludeSimulator <SwitchParameter>] [-Organization
```

```
<OrganizationIdParameter>]
```

Examples

EXAMPLE 1

This example displays a formatted list of all the UM IP gateways in the Active Directory forest.

```
Get-UMIPGateway | Format-List
```

EXAMPLE 2

This example displays the properties for the UM IP gateway MyUMIPGateway.

```
Get-UMIPGateway -Identity MyUMIPGateway
```

EXAMPLE 3

This example displays all the UM IP gateways including IP gateway simulators in the Active Directory forest.

```
Get-UMIPGateway -IncludeSimulator $true
```

Detailed Description

The **Get-UMIPGateway** cmdlet displays the properties and values for a specified UM IP gateway, such as the display name, IP address, status, and outgoing calls settings. If no parameter is specified, all UM IP gateways in the Active Directory forest are returned.

When you're using the **Get-UMIPGateway** cmdlet, you can't enter the IP address configured on the UM IP gateway. You must use the name of the UM IP gateway. The name specified with the *Identity* parameter of the **Get-UMIPGateway** cmdlet can be the same as or different from the host name of the UM IP gateway, for example, **Get-UMIPGateway** *MyUMIPGateway*.

After this task is completed, you can view the list of properties and values for a specific UM IP gateway. Or, if the *Identity* parameter isn't used, the cmdlet returns a list of all UM IP gateways in the forest.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.UMIPGatewayIdParameter	The *Identity* parameter specifies the identifier for the UM IP gateway being viewed. This parameter is the directory object ID for the UM IP gateway.
IncludeSimulator	Optional	System.Management.Automation.SwitchParameter	The *IncludeSimulator* switch retrieves the simulator of the UM IP gateway being viewed. A simulator allows a client to connect to the Mailbox server.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-UMIPGateway

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-UMIPGateway** cmdlet to create a Unified Messaging (UM) IP gateway. A UM IP gateway is used to connect Unified Messaging servers to an IP gateway or a Session Initiation Protocol (SIP)–enabled IP Private Branch eXchange (PBX).

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-UMIPGateway -Address <UMSmartHost> -Name <String> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-GlobalCallRoutingScheme
<None | E164 | GatewayGuid | Reserved1 | Reserved2 | Reserved3>] [-
IPAddressFamily <IPv4Only | IPv6Only | Any>] [-Organization
<OrganizationIdParameter>] [-UMDialPlan <UMDialPlanIdParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the UM IP gateway MyUMIPGateway that enables a Mailbox server to start accepting calls from an IP gateway with an IP address of 10.10.10.1.

```
New-UMIPGateway -Identity MyUMIPGateway -Address 10.10.10.1
```

EXAMPLE 2

This example creates the UM IP gateway MyUMIPGateway and prevents it from accepting incoming calls and outgoing calls, sets an IPv6 address, and allows the UM IP gateway to use IPv4 and IPV6 addresses.

```
New-UMIPGateway -Identity MyUMIPGateway -Address
fe80::39bd:88f7:6969:d223%11 -IPAddressFamily Any -Status
Disabled -OutcallsAllowed $false
```

EXAMPLE 3

This example creates the UM IP gateway MyUMIPGateway that enables a Mailbox server to start accepting calls from an IP gateway with an FQDN of MyUMIPGateway.contoso.com.

```
New-UMIPGateway -Identity MyUMIPGateway -Address
"MyUMIPGateway.contoso.com"
```

Detailed Description

The **New-UMIPGateway** cmdlet creates a UM IP gateway. A UM IP gateway has organization-wide scope and references a single physical IP gateway. The UM IP gateway that's created is used to establish a connection to an IP gateway or a SIP-enabled IP PBX. After this task is completed, a new UM IP gateway is created.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Address	Required	Microsoft.Exchange.Data.UMSmartHost	The *Address* parameter specifies the IP address configured on the IP gateway or SIP-enabled IP PBX.
Name	Required	System.String	The *Name* parameter specifies the display name for the UM IP gateway. The name for the new UM IP gateway can contain up to 64

			characters.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
GlobalCallRoutingScheme	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UMGlobalCallRoutingScheme	The *GlobalCallRoutingScheme* parameter specifies whether the IP gateway can accept calls for UM-enabled users and auto attendant

			numbers included in the global routing database. If the value is set to `E164`, the IP gateway accepts the call.
IPAddressFamily	Optional	Microsoft.Exchange.Data.Directory.IPAddressFamily	The *IPAddressFamily* parameter specifies whether the UM IP gateway uses Internet Protocol version 4 (IPv4), IPv6, or both to communicate. If set to `IPv4Only`, the UM IP gateway only uses IPv4 to communicate. If set to `IPv6Only`, the UM IP gateway only uses IPv6. If set to `Any`, IPv6 will be used first, and then, if necessary, it will fallback to IPv4. The default is `IPv4Only`.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
UMDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the UM dial plan to be associated with the UM

			IP gateway.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-UMIPGateway

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-UMIPGateway** cmdlet to delete a Unified Messaging (UM) IP gateway.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-UMIPGateway -Identity <UMIPGatewayIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the UM IP gateway MyUMIPGateway.

```
Remove-UMIPGateway -Identity MyUMIPGateway
```

Detailed Description

The **Remove-UMIPGateway** cmdlet deletes a specified UM IP gateway. After the UM IP gateway is deleted, Mailbox servers no longer accept new call requests from the IP gateway.

Important:
The **Remove-UMIPGateway** cmdlet should be run only by an administrator who fully understands the implications of disabling communication with a Session Initiation Protocol (SIP)–enabled IP Private Branch eXchange (PBX) or IP gateway.

After this task is completed, the UM IP gateway is removed from Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMIPGatewayIdParameter	The *Identity* parameter specifies the identifier for the UM IP gateway being deleted.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-UMIPGateway

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-UMIPGateway** cmdlet to modify the configuration settings for a single Unified Messaging (UM) IP gateway or to return a list of configuration settings that can be modified on a specified UM IP gateway.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-UMIPGateway -Identity <UMIPGatewayIdParameter> [-Address
<UMSmartHost>] [-Confirm [<SwitchParameter>]] [-
DelayedSourcePartyInfoEnabled <$true | $false>] [-DomainController <Fqdn>]
[-ForceUpgrade <SwitchParameter>] [-IPAddressFamily <IPv4Only | IPv6Only |
Any>] [-MessageWaitingIndicatorAllowed <$true | $false>] [-Name <String>]
[-OutcallsAllowed <$true | $false>] [-Port <Int32>] [-Simulator <$true |
$false>] [-Status <Enabled | Disabled | NoNewCalls>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example modifies the IP address of the UM IP gateway MyUMIPGateway.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address 10.10.10.1
```

EXAMPLE 2

This example prevents the UM IP gateway from accepting incoming calls and prevents outgoing

calls.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address 10.10.10.1
-Status Disabled -OutcallsAllowed $false
```

EXAMPLE 3

This example prevents the UM IP gateway MyUMIPGateway from accepting incoming calls and outgoing calls, sets an IPv6 address, and allows the UM IP gateway to use IPv4 and IPv6 addresses.

```
Set-UMIPGateway -Identity MyUMIPGateway -Address
fe80::39bd:88f7:6969:d223%11 -IPAddressFamily Any -Status
Disabled -OutcallsAllowed $false
```

EXAMPLE 4

This example enables the UM IP gateway to function as an IP gateway simulator and can be used with the **Test-UMConnectivity** cmdlet.

```
Set-UMIPGateway -Identity MyUMIPGateway -Simulator $true
```

Detailed Description

The **Set-UMIPGateway** cmdlet modifies configuration settings for a specific UM IP gateway, for example, the IP address to the IP gateway. These modifications include allowing outgoing calls and controlling communications with a Session Initiation Protocol (SIP)–enabled IP Private Branch eXchange (PBX) or IP gateway.

Important:
It's possible that modifications to the UM IP gateway settings may disrupt communication between Mailbox servers and the SIP-enabled IP PBX or IP gateway. Modifications to a UM IP gateway should be performed only by an administrator who fully understands the implications of making configuration changes to the UM IP gateway.

After this task is completed, the parameters and values specified are configured on the UM IP gateway.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM IP gateways" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMIPGatewayIdParameter	The *Identity* parameter specifies the identifier for the UM IP gateway being modified. This parameter is the directory object ID for the UM IP gateway.
Address	Optional	Microsoft.Exchange.Data.UMSmartHost	The *Address* parameter specifies the IP address or the fully qualified domain name (FQDN) configured on the UM IP gateway or SIP-enabled IP PBX. An FQDN is required if the UM dial plan associated with the UM IP gateway is operating in SIP Secured or Secured mode. If an FQDN is used, verify that the Domain Name System (DNS) has been configured correctly.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't

			have to specify a value with the *Confirm* switch.
DelayedSourcePartyInfoEnabled	Optional	System.Boolean	The *DelayedSourcePartyInfoEnabled* parameter specifies whether Unified Messaging should delay the process of accepting an inbound call from the Voice over IP (VoIP) gateway if the corresponding SIP INVITE of the call contains no calling party and diversion information.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ForceUpgrade	Optional	System.Management.A	This parameter is

		utomation.SwitchParameter	available only in on-premises Exchange 2013. The *ForceUpgrade* switch specifies whether you're prompted for confirmation before a UM IP gateway object is upgraded.
IPAddressFamily	Optional	Microsoft.Exchange.Data.Directory.IPAddressFamily	The *IPAddressFamily* parameter specifies whether the UM IP gateway will use Internet Protocol version 4 (IPv4), IPv6, or both to communicate. If set to `IPv4Only`, the UM IP gateway will only use IPv4 to communicate. If set to `IPv6Only`, the UM IP gateway will only use IPv6. If set to `Any`, IPv6 will be used first, and then if necessary, it will fall back to IPv4. The default is `IPv4Only`.
MessageWaitingIndicatorAllowed	Optional	System.Boolean	The *MessageWaitingIndicatorAllowed* parameter specifies whether to

			enable the UM IP gateway to allow SIP NOTIFY messages to be sent to users associated with a UM dial plan and the UM IP gateway. The default value is `$true`.
Name	Optional	System.String	The *Name* parameter specifies the display name for the UM IP gateway. This display name is limited to 64 characters.
OutcallsAllowed	Optional	System.Boolean	The *OutcallsAllowed* parameter specifies whether to allow this UM IP gateway to be used for outgoing calls. This doesn't govern call transfers.
Port	Optional	System.Int32	The *Port* parameter specifies the IP port on which the IP gateway or IP PBX is listening. By default, it's port 5060. The range for this parameter is from 0 through 65535.
Simulator	Optional	System.Boolean	The *Simulator* parameter specifies the simulator used for the

			UM IP gateway being viewed. A simulator allows a client to connect to the Mailbox server.
Status	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.GatewayStatus	The *Status* parameter specifies whether to enable or disable the UM IP gateway.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-UMMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Disable-UMMailbox** cmdlet to disable Unified Messaging (UM) for a UM-enabled recipient.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-UMMailbox -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreDefaultScope
<SwitchParameter>] [-KeepProperties <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables Unified Messaging on the mailbox for tonysmith@contoso.com.

```
Disable-UMMailbox -Identity tonysmith@contoso.com
```

Detailed Description

The **Disable-UMMailbox** cmdlet disables Unified Messaging for a Microsoft Exchange Server 2013 recipient who's currently UM-enabled. When the task is complete, Mailbox servers running the Microsoft Exchange Unified Messaging service no longer handle calls for the extension number associated with the mailbox. You can continue to use the Exchange mailbox for all other operations unrelated to Unified Messaging.

After this task is completed, the user is disabled for Unified Messaging and can't use the voice mail features found in Unified Messaging any longer.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the user to disable for Unified Messaging. The variables for this parameter include the following: • ADObjectID • GUID • Distinguished Name (DN) • Domain\Account • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically.

			• You can only use the distinguished name for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
KeepProperties	Optional	System.Boolean	The *KeepProperties* parameter specifies whether the mailbox and directory resident properties should be retained. If this parameter isn't included, the user's UM properties are retained.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-UMMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Enable-UMMailbox** cmdlet to enable Unified Messaging (UM) for an existing mail-enabled user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-UMMailbox -Identity <MailboxIdParameter> -UMMailboxPolicy
<MailboxPolicyIdParameter> [-AutomaticSpeechRecognitionEnabled <$true |
$false>] [-Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
Extensions <MultiValuedProperty>] [-IgnoreDefaultScope <SwitchParameter>]
[-NotifyEmail <String>] [-PilotNumber <String>] [-Pin <String>] [-
PinExpired <$true | $false>] [-SIPResourceIdentifier <String>] [-
ValidateOnly <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables Unified Messaging on the mailbox for tonysmith@contoso.com, sets the extension and PIN for the user that must be changed when the user logs on to Outlook Voice Access, assigns the UM mailbox policy MyUMMailboxPolicy to the user's mailbox, and then sends an email message that contains the Unified Messaging welcome information to administrator@contoso.com.

```
Enable-UMMailbox -Identity tonysmith@contoso.com -
UMMailboxPolicy MyUMMailboxPolicy -Extensions 51234 -PIN
5643892 -NotifyEmail administrator@contoso.com -PINExpired
$true
```

EXAMPLE 2

This example enables Unified Messaging on a SIP-enabled mailbox for tonysmith@contoso.com, associates the UM mailbox policy MyUMMailboxPolicy, and sets the extension number, SIP resource identifier, and PIN for the user that must be changed when the user logs on to Outlook Voice Access, and then sends an email message that contains the Unified Messaging welcome information to tonysmith@contoso.com.

```
Enable-UMMailbox -Identity tonysmith@contoso.com -
UMMailboxPolicy MyUMMailboxPolicy -Extensions 51234 -PIN
5643892 -SIPResourceIdentifier "tonysmith@contoso.com" -
PINExpired $true
```

Detailed Description

The **Enable-UMMailbox** cmdlet enables Unified Messaging for an existing mail-enabled Microsoft Exchange Server 2013 user. When the mail-enabled user is enabled for Unified Messaging, the settings from a UM mailbox policy are applied to the user. After the user is enabled for Unified Messaging, the user can use the UM features included with Exchange 2013.

After this task is completed, the user is enabled for Unified Messaging and can use the voice mail features included with Exchange 2013.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the user to enable for Unified Messaging. The values for this parameter include the following: • ADObjectID • GUID • Distinguished name (DN) • *Domain\Account*

			• user principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
UMMailboxPolicy	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *UMMailboxPolicy* parameter specifies the name of the UM mailbox policy to be associated with the user.
AutomaticSpeechRecognitionEnabled	Optional	System.Boolean	The *AutomaticSpeechRecognitionEnabled* parameter enables Automatic Speech Recognition (ASR) to be used with the UM mailbox. ASR is only available if the user's specified language preference is installed.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Da	This parameter is available only in on-

		ta.Fqdn	premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Extensions	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Extensions* parameter specifies the extension number for the user. Either a single extension number or an array of telephone number extensions can be specified. The user's extension must be unique to the UM dial plan. If this parameter isn't included, a default telephone number value from Active Directory is used. If you're enabling a user for Unified Messaging using a Session Initiation Protocol (SIP) Uniform Resource Identifier (URI) or E.164 dial plan, you must also specify the *SIPResourceIdentifier* parameter.

IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the distinguished name for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.

NotifyEmail	Optional	System.String	The *NotifyEmail* parameter specifies the email address to which the server sends the email message that contains the Unified Messaging welcome information. By default, the message is sent to the SMTP address of the enabled user.
PilotNumber	Optional	System.String	The *PilotNumber* parameter specifies the subscriber access number users can dial to gain access to their mailboxes. The default value is the subscriber access number specified on a dial plan object in Active Directory.
Pin	Optional	System.String	The *PIN* parameter specifies the value for the initial PIN to be used with the UM mailbox. The PIN is checked against the UM mailbox policy rules. The PIN value must be from 4 through 24 numeric characters. If no PIN is specified, a PIN generated by the system is sent to the user. The PIN generated by the system

			contains six numeric characters, which is the default.
PinExpired	Optional	System.Boolean	The *PINExpired* parameter specifies whether the PIN is treated as expired. If this parameter is supplied and is set to `$false`, users aren't required to reset their PIN the next time they log on. If the PIN isn't supplied, the PIN is treated as expired and users are prompted to reset their PIN the next time they log on.
SIPResourceIdentifier	Optional	System.String	The *SIPResourceIdentifier* parameter specifies the SIP address or E.164 address for the user. This property is compared to the URI type defined on the UM dial plan.
ValidateOnly	Optional	System.Management.Automation.SwitchParameter	The *ValidateOnly* switch tells the cmdlet to evaluate the conditions and requirements necessary to perform the operation and then reports whether the operation will succeed or

			fail. No changes are made when the *ValidateOnly* switch is used.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-UMMailbox** cmdlet to display the Unified Messaging (UM) properties for a UM-enabled recipient.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMMailbox [-Anr <String>] <COMMON PARAMETERS>

Get-UMMailbox [-Identity <MailboxIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-Filter <String>]
[-IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example displays a list of all the UM-enabled mailboxes in the Active Directory forest in a formatted list.

```
Get-UMMailbox | Format-List
```

EXAMPLE 2

This example displays the UM mailbox properties for tonysmith@contoso.com.

```
Get-UMMailbox -Identity tonysmith@contoso.com
```

Detailed Description

The **Get-UMMailbox** cmdlet retrieves the Unified Messaging properties for a single UM mailbox. It can also return a list of UM-enabled mailboxes.

After this task is completed, you can see the properties and values configured on a single UM mailbox. Or, if the *Identity* parameter isn't used, the cmdlet returns a list of all the UM-enabled mailboxes in a forest.

Note:
When running in an environment with Exchange Server 2007 and Exchange Server 2013 where the user's mailbox is located on an Exchange 2007 Mailbox server, running the **Get-UMMailbox** cmdlet on an Exchange 2013 server won't work correctly. To resolve the issue, run the **Get-UMMailbox** cmdlet from an Exchange 2007 server or a computer running the Exchange 2007 administrative tools.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not

included in the permissions assigned to you. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	The *AccountPartition*parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects that have an attribute that matches that string. The following default attributes are searched: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the user name and password to use to access Active Directory.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter filters the results returned by the cmdlet.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the user to enable for Unified Messaging. The values for this parameter include the following: • ADObjectID • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchPar	This parameter is available only in on-premises Exchange 2013.

		ameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the distinguished name for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters

			together. • You can't use the *Credential* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	This parameter is available only in on-premises Exchange 2013. The *OrganizationalUnit* parameter specifies an organizational unit (OU) and is used to limit the results. If you use this parameter, you retrieve only mailboxes in the container you specify. You can use the OU or the domain name. If you use the OU, you must specify the canonical name of the OU.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's

			domain. If you've set the recipient scope to include all recipients in the forest and if you don't use this parameter, the user information may be read from a global catalog whose information is outdated. If you do use this parameter, multiple reads may be necessary to get the information. By default, the recipient scope is set to the domain that hosts your Exchange servers.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies a maximum number of results to be returned. Otherwise, a maximum number isn't specified, and the command returns all results.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. You can sort by only one attribute at a time. You can sort by the following attributes: • **Alias**

			• **DisplayName** • **Name** The results are sorted in ascending order.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-UMMailbox

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-UMMailbox** cmdlet to set the Unified Messaging (UM) properties for a user who is currently UM-enabled.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-UMMailbox -Identity <MailboxIdParameter> [-AirSyncNumbers
<MultiValuedProperty>] [-AllowUMCallsFromNonUsers <None | SearchEnabled>]
[-AnonymousCallersCanLeaveMessages <$true | $false>] [-
AutomaticSpeechRecognitionEnabled <$true | $false>] [-
CallAnsweringAudioCodec <G711 | Wma | Gsm | Mp3>] [-
CallAnsweringRulesEnabled <$true | $false>] [-Confirm [<SwitchParameter>]]
[-DomainController <Fqdn>] [-FaxEnabled <$true | $false>] [-
IgnoreDefaultScope <SwitchParameter>] [-ImListMigrationCompleted <$true |
$false>] [-MissedCallNotificationEnabled <$true | $false>] [-Name
<String>] [-OperatorNumber <String>] [-PhoneNumber <String>] [-
PhoneProviderId <String>] [-PinlessAccessToVoiceMailEnabled <$true |
$false>] [-PlayOnPhoneEnabled <$true | $false>] [-SubscriberAccessEnabled
<$true | $false>] [-TUIAccessToCalendarEnabled <$true | $false>] [-
TUIAccessToEmailEnabled <$true | $false>] [-UMMailboxPolicy
<MailboxPolicyIdParameter>] [-UMSMSNotificationOption <None | VoiceMail |
VoiceMailAndMissedCalls>] [-VerifyGlobalRoutingEntry <SwitchParameter>] [-
VoiceMailAnalysisEnabled <$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example configures a UM-enabled user, tony@contoso.com with the following settings:

- Changes the call answering audio codec to Wma
- Disables call answering rules
- Prevents him from receiving incoming faxes
- Enables voice mail notifications but not missed call notifications using text messaging

```
Set-UMMailbox -Identity tony@contoso.com -
CallAnsweringAudioCodec Wma -CallAnsweringRulesEnabled
$false -FaxEnabled $false -UMSMSNotificationOption
VoiceMail
```

EXAMPLE 2

This example prevents the user tony@contoso.com from accessing his calendar and email when he's using Outlook Voice Access.

```
Set-UMMailbox -Identity tony@contoso.com -
TUIAccessToCalendarEnabled $false -TUIAccessToEmailEnabled
$false
```

Detailed Description

The **Set-UMMailbox** cmdlet sets UM properties associated with a user who has been UM-enabled. Many of the UM properties for the user are stored on the user's mailbox, and other UM properties for the user are stored in Active Directory.

After this task is completed, the parameters and values specified are configured on the UM mailbox.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Co	The *Identity* parameter

		nfiguration.Tasks.MailboxIdParameter	specifies the user to enable for Unified Messaging. The values for this parameter include the following: • ADObjectID • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
AirSyncNumbers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AirSyncNumbers* parameter specifies whether to register a mobile phone number with a hosted voice mail service. Each UM mailbox can have up to three numbers defined, and numbers must be in E.164 format.
AllowUMCallsFromNonUsers	Optional	Microsoft.Exchange.Data.Directory.Recipient.AllowUMCallsFromNonUsersFlags	The *AllowUMCallsFromNonUsers* parameter specifies whether to exclude the mailbox from directory searches.
AnonymousCallersCa	Optional	System.Boolean	The

nLeaveMessages			*AnonymousCallersCanLeaveMessages* parameter specifies whether diverted calls without a caller ID are allowed to leave a message.
AutomaticSpeechRecognitionEnabled	Optional	System.Boolean	The *AutomaticSpeechRecognitionEnabled* parameter specifies whether users can use Automatic Speech Recognition (ASR) when they log on to their mailbox. This parameter can only be set to `$true` if there is ASR support for the language selected by the user in Microsoft Office Outlook Web App **Options**.
CallAnsweringAudioCodec	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.AudioCodecEnum	The *CallAnsweringAudioCodec* parameter specifies the audio codec used to encode voice mail messages left for the user. The audio codec used is the audio codec set on the UM dial plan. The default value is Mp3.
CallAnsweringRulesEn	Optional	System.Boolean	The

abled			*CallAnsweringRulesEnabled* parameter specifies whether users can configure or set up Call Answering Rules for their accounts. The default value is `$true`.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
FaxEnabled	Optional	System.Boolean	The *FaxEnabled* parameter specifies whether a user is allowed

			to receive incoming faxes.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the distinguished name for the *Identity* parameter. Other forms of identification, such as

			alias or GUID, aren't accepted.
ImListMigrationCompleted	Optional	System.Boolean	The *ImListMigrationCompleted* parameter specifies whether a UM-enabled user's Microsoft Lync 2013 contact list has been successfully migrated from a user's Exchange mailbox to Lync 2013 servers and that the user's Microsoft Exchange Server 2013 mailbox can be migrated back to Exchange Server 2010. For Lync 2013, the contact list for a Lync user can be stored in an Exchange 2013 user's mailbox called a Unified Contact Store (UCS). By storing the contact list in a user's mailbox, it allows applications to show a consistent, up-to-date list of the user's contact list. If a user's mailbox is being migrated from Exchange 2013 to Exchange 2010, and the user is in UCS mode, the user's contact list must first be moved

			from their Exchange 2013 mailbox to Lync 2013 to preserve the user's contact list. This is the case because Exchange 2010 doesn't support the Exchange Web Services (EWS) methods used to support the UCS feature in Lync 2013. A setting of `$false` indicates that a Lync user's contacts haven't been migrated. The default is `$false`.
MissedCallNotificationEnabled	Optional	System.Boolean	The *MissedCallNotificationEnabled* parameter specifies whether to send missed call notifications. **⚠Warning:** When you're integrating Unified Messaging and Lync Server, missed call notifications aren't available to users who have a mailbox located on an Exchange 2007 or Exchange 2010 Mailbox server. A missed call notification is generated when a user disconnects before the call is sent to a Mailbox server.
Name	Optional	System.String	The *Name* parameter

			specifies the display name for the user. The display name is limited to 64 characters.
OperatorNumber	Optional	System.String	The *OperatorNumber* parameter specifies the string of digits for the personal operator.
PhoneNumber	Optional	System.String	This parameter is available only in the cloud-based service. The *PhoneNumber* parameter specifies whether to assign a phone number to a UM-enabled user. This is only used for E.164 consumer dial plans.
PhoneProviderId	Optional	System.String	This parameter is available only in the cloud-based service. The *PhoneProviderId* parameter specifies the user's phone number and mobile service provider. This information is used to provide custom call forwarding and cancelling voice mail setup instructions based on the mobile phone provider.

PinlessAccessToVoiceMailEnabled	Optional	System.Boolean	The *PinlessAccessToVoiceMailEnabled* parameter specifies whether UM-enabled users are required to use a PIN to access their voice mail. A PIN is still required to access email and the calendar. The default value is `$false`.
PlayOnPhoneEnabled	Optional	System.Boolean	The *PlayOnPhoneEnabled* parameter specifies whether a user can use the Play on Phone feature to listen to voice messages. The default value is `$true`.
SubscriberAccessEnabled	Optional	System.Boolean	The *SubscriberAccessEnabled* parameter specifies whether the users are allowed subscriber access to their individual mailboxes. If it's set to `$true`, after users are authenticated, they're able to retrieve voice mail over the telephone. The default value is `$true`.
TUIAccessToCalendarE	Optional	System.Boolean	The

nabled			*TUIAccessToCalendarEnabled* parameter specifies whether UM-enabled users can access and manage their individual calendar using the Microsoft Outlook Voice Access telephone user interface (TUI) or touchtone interface. The default value is `$true`.
TUIAccessToEmailEnabled	Optional	System.Boolean	The *TUIAccessToEmailEnabled* parameter specifies whether users can access their individual email messages over the telephone. The default value is `$true`.
UMMailboxPolicy	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *UMMailboxPolicy* parameter specifies the UM mailbox policy associated with the UM-enabled user's mailbox.
UMSMSNotificationOption	Optional	Microsoft.Exchange.Data.Directory.Recipient.UMSMSNotificationOptions	The *UMSMSNotificationOption* parameter specifies whether a UM-enabled user gets SMS or text messaging notifications for voice mail only, voice

			mail and missed calls, or no notifications. The values for this parameter are: `VoiceMail`, `VoiceMailAndMissedCalls`, and `None`. The default value is `None`.
VerifyGlobalRoutingEntry	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *VerifyGlobalRoutingEntry* parameter specifies the phone number has been successfully registered in the global routing database.
VoiceMailAnalysisEnabled	Optional	System.Boolean	The *VoiceMailAnalysisEnabled* parameter specifies whether a copy of each voice mail left for a UM-enabled user will be forwarded to Microsoft for analysis and improvement of speech recognition features.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMMailboxPIN

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-UMMailboxPIN** cmdlet to return information from a Unified Messaging (UM)-enabled user's mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMMailboxPin [-Identity <MailboxIdParameter>] [-Credential
<PSCredential>] [-DomainController <Fqdn>] [-IgnoreDefaultScope
<SwitchParameter>] [-IgnoreErrors <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-ReadFromDomainController <SwitchParameter>]
[-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example displays the UM mailbox PIN-related properties for all UM-enabled users.

```
Get-UMMailboxPIN
```

EXAMPLE 2

This example displays the UM mailbox PIN-related properties for tonysmith@contoso.com.

```
Get-UMMailboxPIN -Identity tonysmith@contoso.com
```

Detailed Description

The **Get-UMMailboxPIN** cmdlet returns information calculated from the PIN data stored in encrypted form in the user's mailbox. This cmdlet also shows whether the mailbox or user access has been locked out.

After this task is completed, you can view information on a user's mailbox.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the identifier that can be used to retrieve information about the mailbox. The values for this parameter include the following: • ADObjectID • GUID • Distinguished name (DN) • *Domain\Account* • User principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the

			command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *Organization* and *Identity* parameters together. • You can't use the *Credential* parameter.

IgnoreErrors	Optional	System.Management.Automation.SwitchParameter	The *IgnoreErrors* switch specifies whether errors that may occur when running this cmdlet are written as warnings.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you've set the recipient scope to include all recipients in the forest and if you don't use this parameter, the user information may be read from a global catalog whose information is outdated. If you use this parameter, multiple reads may be necessary to get the information. By default, the recipient scope is set to the domain that hosts your Exchange

			servers.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies a maximum number of results to be returned. If a maximum number isn't specified, the cmdlet returns all results.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-UMMailboxPIN

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-UMMailboxPIN** cmdlet to reset the PIN for a Unified Messaging (UM)–enabled mailbox.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-UMMailboxPIN -Identity <MailboxIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreDefaultScope
<SwitchParameter>] [-LockedOut <$true | $false>] [-NotifyEmail <String>]
[-Pin <String>] [-PinExpired <$true | $false>] [-SendEmail <$true |
$false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example resets the PIN on the UM-enabled mailbox for tonysmith@contoso.com.

```
Set-UMMailboxPIN -Identity tonysmith@contoso.com
```

EXAMPLE 2

This example resets the initial PIN to 1985848 on the UM-enabled mailbox for tonysmith@contoso.com, and then sets the PIN as expired so that the user will be asked to change the PIN the next time the user logs on.

```
Set-UMMailboxPIN -Identity tonysmith@contoso.com -PIN
1985848 -PinExpired $true
```

EXAMPLE 3

This example locks the UM-enabled mailbox for tonysmith@contoso.com to prevent the user from accessing the mailbox.

```
Set-UMMailboxPIN -Identity tonysmith@contoso.com -LockedOut
$true
```

EXAMPLE 4

This example unlocks the UM-enabled mailbox for tonysmith@contoso.com and allows the user access to the mailbox.

```
Set-UMMailboxPIN -Identity tonysmith@contoso.com -LockedOut
$false
```

Detailed Description

The **Set-UMMailboxPIN** cmdlet is used when a UM-enabled user has been locked out of a mailbox because either the user tried to log on by using an incorrect PIN multiple times or because the user has forgotten the PIN. You can use this cmdlet to set the user's PIN. The new PIN must comply with the PIN policy rules specified in the user's mailbox policy. The new PIN is sent to the user in an email message, or sent to an alternative email address. You can control whether the user must reset the PIN at logon and if the mailbox will continue to be locked.

After this task is completed, the PIN on a UM-enabled mailbox is set.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM

mailboxes" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	The *Identity* parameter specifies the UM-enabled user PIN being set. The values for this parameter include the following: • ADObjectID • GUID • Distinguished name (DN) • Domain\Account • user principal name (UPN) • LegacyExchangeDN • SmtpAddress • Alias
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global

			catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
LockedOut	Optional	System.Boolean	The *LockedOut* parameter specifies whether the mailbox will continue to be locked. If set to `$true`, the mailbox is marked as locked out. By default, if this parameter is omitted or set to `$false`, the command clears the locked-out status on the mailbox.
NotifyEmail	Optional	System.String	The *NotifyEmail* parameter specifies the email address to which the server sends the email message that contains the PIN reset information. By default, the message is sent to the SMTP address of the enabled user.
Pin	Optional	System.String	The *Pin* parameter specifies a new PIN for use with the mailbox. The PIN is checked against the

			PIN rules defined in the Unified Messaging mailbox policy. If the PIN isn't supplied, the command generates a new PIN for the mailbox and includes it in an email message sent to the user.
PinExpired	Optional	System.Boolean	The *PINExpired* parameter specifies whether the PIN is treated as expired. If this parameter is supplied and is set to `$false`, the user isn't required to reset the PIN the next time that the user logs on. If the PIN isn't supplied, the PIN is treated as expired and the user is prompted to reset the PIN the next time that the user logs on.
SendEmail	Optional	System.Boolean	The *SendEmail* parameter specifies whether to send a PIN to the user in an email message. The default is `$true`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-UMMailboxPolicy** cmdlet to display the properties and values of a Unified Messaging (UM) mailbox policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMMailboxPolicy [-Identity <MailboxPolicyIdParameter>] [-
DomainController <Fqdn>] [-Organization <OrganizationIdParameter>] [-
UMDialPlan <UMDialPlanIdParameter>]
```

Examples

EXAMPLE 1

This example returns a formatted list of all UM mailbox policies in the Active Directory forest.

```
Get-UMMailboxPolicy | Format-List
```

EXAMPLE 2

This example returns the properties and values for the UM mailbox policy MyUMMailboxPolicy.

```
Get-UMMailboxPolicy -Identity MyUMMailboxPolicy
```

EXAMPLE 3

This examples displays all the UM mailbox policies associated with the UM dial plan MyUMDialPlan.

```
Get-UMMailboxPolicy -UMDialPlan MyUMDialPlan
```

Detailed Description

The **Get-UMMailboxPolicy** cmdlet retrieves the configuration properties and values for a UM mailbox policy or returns a list of UM mailbox policies.

After this task is completed, if the *Identity* parameter is supplied, the properties and values for the specified UM mailbox policy object are returned. If no parameter is specified at the command prompt, all UM mailbox policies in the Active Directory forest are returned.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the

			domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the identifier for the UM mailbox policy being viewed. This is the directory object ID for the UM mailbox policy.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
UMDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies that all UM mailbox policies associated with the UM dial plan are displayed.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-UMMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-UMMailboxPolicy** cmdlet to create a Unified Messaging (UM) mailbox policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-UMMailboxPolicy -Name <String> -UMDialPlan <UMDialPlanIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-Organization
<OrganizationIdParameter>] [-SharedUMDialPlan <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the UM mailbox policy MyUMMailboxPolicy associated with the UM dial plan MyUMDialPlan.

```
New-UMMailboxPolicy -Name MyUMMailboxPolicy -UMDialPlan
MyUMDialPlan
```

Detailed Description

The **New-UMMailboxPolicy** cmdlet creates a UM mailbox policy that has organization-wide scope. UM mailbox policies provide a set of policy values to be applied to UM-enabled users associated with a particular UM dial plan. UM mailbox policies are directly associated with UM dial plans. Therefore, the settings contained within a UM mailbox policy apply only to UM-enabled users of the UM dial plan that the UM mailbox policy is associated with. You can also use the **New-UMMailboxPolicy** cmdlet to create a UM mailbox policy template that can be used to create additional UM mailbox policies.

After this task is completed, a new UM mailbox policy is created.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the display name for the UM mailbox policy. The name for the UM mailbox policy can contain as many as 64 characters.
UMDialPlan	Required	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the identifier for the UM dial plan to be associated with the UM mailbox policy. This parameter is the directory object ID for the UM dial plan.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Dat	This parameter is

		a.Fqdn	available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
SharedUMDialPlan	Optional	System.Management.Automation.SwitchParameter	The *SharedUMDialPlan* switch specifies whether the new UM mailbox policy being created is linked or associated with a dial plan outside the scope of the organization. If you specify this parameter, the UM mailbox policy can be linked with another tenant's dial plan. This parameter is used during tenant provisioning and is only used in a data

			center.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-UMMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-UMMailboxPolicy** cmdlet to delete a Unified Messaging (UM) mailbox policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-UMMailboxPolicy -Identity <MailboxPolicyIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the UM mailbox policy MyUMMailboxPolicy.

```
Remove-UMMailboxPolicy -Identity MyUMMailboxPolicy
```

Detailed Description

The **Remove-UMMailboxPolicy** cmdlet deletes or removes a UM mailbox policy. If the UM mailbox policy is deleted from Active Directory, the UM mailbox policy can't be used when configuring UM-enabled users. The UM mailbox policy can't be deleted if the UM mailbox policy is referenced by any UM-enabled mailboxes.

After this task is completed, the UM mailbox policy is removed from Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the identifier for the UM mailbox policy being deleted. This parameter is the directory object ID for the UM mailbox policy.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf*

			switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-UMMailboxPolicy

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-UMMailboxPolicy** cmdlet to modify a Unified Messaging (UM) mailbox policy.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-UMMailboxPolicy -Identity <MailboxPolicyIdParameter> [-
AllowAutomaticSpeechRecognition <$true | $false>] [-
AllowCallAnsweringRules <$true | $false>] [-AllowCommonPatterns <$true |
$false>] [-AllowDialPlanSubscribers <$true | $false>] [-
AllowedInCountryOrRegionGroups <MultiValuedProperty>] [-
AllowedInternationalGroups <MultiValuedProperty>] [-AllowExtensions <$true
| $false>] [-AllowFax <$true | $false>] [-AllowMessageWaitingIndicator
<$true | $false>] [-AllowMissedCallNotifications <$true | $false>] [-
AllowPinlessVoiceMailAccess <$true | $false>] [-AllowPlayOnPhone <$true |
$false>] [-AllowSMSNotification <$true | $false>] [-AllowSubscriberAccess
<$true | $false>] [-AllowTUIAccessToCalendar <$true | $false>] [-
AllowTUIAccessToDirectory <$true | $false>] [-AllowTUIAccessToEmail <$true
| $false>] [-AllowTUIAccessToPersonalContacts <$true | $false>] [-
AllowVoiceMailAnalysis <$true | $false>] [-AllowVoiceMailPreview <$true |
$false>] [-AllowVoiceNotification <$true | $false>] [-
AllowVoiceResponseToOtherMessageTypes <$true | $false>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-FaxMessageText <String>]
[-FaxServerURI <String>] [-ForceUpgrade <SwitchParameter>] [-
InformCallerOfVoiceMailAnalysis <$true | $false>] [-
LogonFailuresBeforePINReset <Unlimited>] [-MaxGreetingDuration <Int32>] [-
MaxLogonAttempts <Unlimited>] [-MinPINLength <Int32>] [-Name <String>] [-
PINHistoryCount <Int32>] [-PINLifetime <Unlimited>] [-
ProtectAuthenticatedVoiceMail <None | Private | All>] [-
ProtectedVoiceMailText <String>] [-ProtectUnauthenticatedVoiceMail <None |
Private | All>] [-RequireProtectedPlayOnPhone <$true | $false>] [-
ResetPINText <String>] [-SourceForestPolicyNames <MultiValuedProperty>] [-
UMDialPlan <UMDialPlanIdParameter>] [-UMEnabledText <String>] [-
```

```
VoiceMailPreviewPartnerAddress <SmtpAddress>] [-
VoiceMailPreviewPartnerAssignedID <String>] [-
VoiceMailPreviewPartnerMaxDeliveryDelay <Int32>] [-
VoiceMailPreviewPartnerMaxMessageDuration <Int32>] [-VoiceMailText
<String>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example sets the PIN settings for users associated with the UM mailbox policy MyUMMailboxPolicy.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
LogonFailuresBeforePINReset 8 -MaxLogonAttempts 12 -
MinPINLength 8 -PINHistoryCount 10 -PINLifetime 60 -
ResetPINText "The PIN used to allow you access to your
mailbox using Outlook Voice Access has been reset."
```

EXAMPLE 2

This example selects the in-country or region groups and international groups from those configured on the UM dial plan associated with the UM mailbox policy. UM-enabled users associated with this UM mailbox policy can place outbound calls according to the rules defined on these groups.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
AllowDialPlanSubscribers $true -
AllowedInCountryOrRegionGroups InCountry/
RegionGroup1,InCountry/RegionGroup2 -
AllowedInternationalGroups
InternationalGroup1,InternationalGroup2 -AllowExtensions
$true
```

EXAMPLE 3

This example configures the text of voice mail messages sent to UM-enabled users and the text included in an email message sent to a user who is UM-enabled.

```
Set-UMMailboxPolicy -Identity MyUMMailboxPolicy -
UMEnabledText "You have been enabled for Unified
Messaging." -VoiceMailText "You have received a voice mail
message from Microsoft Exchange 2013 Unified Messaging."
```

Detailed Description

When the **Set-UMMailboxPolicy** cmdlet is used to modify UM mailbox policy objects, you can change settings such as PIN policies, message text settings, and dialing restrictions for a single UM-enabled recipient or multiple UM-enabled recipients. UM mailbox policies are associated with UM-enabled mailboxes and can be configured to increase the level of security for UM-enabled users.

After this task is completed, the parameters and values specified are configured on the UM mailbox policy.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM mailbox policies" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailboxPolicyIdParameter	The *Identity* parameter specifies the identifier for the UM mailbox policy being modified. This is the directory object ID for the UM mailbox policy.
AllowAutomaticSpeechRecognition	Optional	System.Boolean	The *AllowAutomaticSpeechRecognition* parameter specifies whether users associated with the UM mailbox policy can use Automatic Speech Recognition (ASR). The default value is `$true`.
AllowCallAnsweringRules	Optional	System.Boolean	The *AllowCallAnsweringRule*

			s parameter specifies whether users associated with the UM mailbox policy are allowed to configure or set up Call Answering Rules for their accounts. The default value is `$true`.
AllowCommonPatterns	Optional	System.Boolean	The *AllowCommonPatterns* parameter specifies whether to allow obvious PINs. Examples of obvious PINs include subsets of the telephone number, sequential numbers, or repeated numbers. If set to `$false`, sequential and repeated numbers and the suffix of the mailbox extension are rejected. If set to `$true`, only the suffix of the mailbox extension is rejected.
AllowDialPlanSubscribers	Optional	System.Boolean	The *AllowDialPlanSubscribers* parameter specifies whether to let subscribers in a dial plan dial a number that

			resolves to another subscriber within the same dial plan. The default value is `$true`.
AllowedInCountryOrRegionGroups	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AllowedInCountryOrRegionGroups* parameter specifies whether to let subscribers dial the list of in-country/region dial group names. The names that subscribers are allowed to dial must match the group names defined in the UM dial plan. The string is limited to 128 characters.
AllowedInternationalGroups	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AllowedInternationalGroups* parameter specifies whether to let subscribers dial the list of international dial group names. The names that subscribers dial must match the group names defined in the dial plan.
AllowExtensions	Optional	System.Boolean	The *AllowExtensions* parameter specifies whether to let

			subscribers dial calls to the number of digits specified on the UM dial plan. The default value is `$true`.
AllowFax	Optional	System.Boolean	The *AllowFax* parameter specifies whether users associated with the UM mailbox policy are allowed to receive incoming faxes. The default value is `$true`.
AllowMessageWaitingIndicator	Optional	System.Boolean	The *AllowMessageWaitingIndicator* parameter specifies whether users associated with the UM mailbox policy can receive notifications that they've received a new voice mail message. The default value is `$true`.
AllowMissedCallNotifications	Optional	System.Boolean	The *AllowMissedCallNotifications* parameter specifies whether missed call notifications are enabled for users associated with the UM mailbox policy. The default value is `$true`.

			⚠Warning: When you're integrating Unified Messaging and Lync Server, missed call notifications aren't available to users who have a mailbox located on an Exchange 2007 or Exchange 2010 Mailbox server. A missed call notification is generated when a user disconnects before the call is sent to a Mailbox server.
AllowPinlessVoiceMailAccess	Optional	System.Boolean	The *AllowPinlessVoiceMailAccess* parameter specifies whether users associated with the UM mailbox policy are required to use a PIN to access their voice mail. A PIN is still required to access their email and calendar. The default value is `$false`.
AllowPlayOnPhone	Optional	System.Boolean	The *AllowPlayOnPhone* parameter specifies whether users associated with the UM mailbox policy can use the Play on Phone feature to listen to voice mail messages. The default value is `$true`.
AllowSMSNotification	Optional	System.Boolean	The

			AllowSMSNotification parameter specifies whether UM-enabled users associated with the UM mailbox policy are allowed to get SMS or text messages sent to their mobile phones. If this parameter is set to `$true`, you also want to set the **Set-UMMailbox** cmdlet *UMSMSNotificationOption* parameter for the UM-enabled user to either `voiceMail` or `VoiceMailAndMissedCalls`. The default value is `$true`.
AllowSubscriberAccess	Optional	System.Boolean	The *AllowSubscriberAccess* parameter specifies whether users associated with the UM mailbox policy are allowed subscriber access to their individual mailboxes. If this parameter is set to `$true`, after users are authenticated, they're able to retrieve voice mail over the telephone.

			The default value is `$true`.
AllowTUIAccessToCalendar	Optional	System.Boolean	The *AllowTUIAccessToCalendar* parameter specifies whether users associated with the UM mailbox policy can access their individual calendars over the telephone. The default value is `$true`.
AllowTUIAccessToDirectory	Optional	System.Boolean	The *AllowTUIAccessToDirectory* parameter specifies whether users associated with the UM mailbox policy can access the directory over the telephone. The default value is `$true`.
AllowTUIAccessToEmail	Optional	System.Boolean	The *AllowTUIAccessToEmail* parameter specifies whether users associated with the UM mailbox policy can access their individual email messages over the telephone. The default value is `$true`.

AllowTUIAccessToPersonalContacts	Optional	System.Boolean	The *AllowTUIAccessToPersonalContacts* parameter specifies whether users associated with the UM mailbox policy can access their personal contacts over the telephone. The default value is `$true`.
AllowVoiceMailAnalysis	Optional	System.Boolean	The *AllowVoiceMailAnalysis* parameter specifies whether a copy of each voice mail left for the users associated with the UM mailbox policy will be forwarded to Microsoft for analysis and improvement of our speech recognition features.
AllowVoiceMailPreview	Optional	System.Boolean	The *AllowVoiceMailPreview* parameter specifies whether users associated with the UM mailbox policy are able to receive Voice Mail Previews for call-answered messages, or have Voice Mail

			Previews provided for voice mail messages that they send to other users in their mailbox. The default value is `$true`.
AllowVoiceNotification	Optional	System.Boolean	The *AllowVoiceNotification* parameter will be removed in future versions of the product.
AllowVoiceResponseToOtherMessageTypes	Optional	System.Boolean	The *AllowVoiceResponseToOtherMessageTypes* parameter specifies whether UM-enabled users associated with the UM mailbox policy can record and attach a voice mail message when replying to email messages and calendar items.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the

			Confirm switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
FaxMessageText	Optional	System.String	The *FaxMessageText* parameter specifies the text included in the body part of fax messages. This text is limited to 512 characters.
FaxServerURI	Optional	System.String	The *FaxServerURI* parameter specifies the Session Initiation Protocol (SIP) Uniform Resource Identifier (URI) for the fax solution that serves the UM-enabled users associated with the UM mailbox policy. This fax product or fax service accepts incoming fax calls that were redirected

			from Microsoft Exchange Server 2013 Client Access and Mailbox servers and creates inbound fax messages for the UM-enabled users associated with the UM mailbox policy. Although you can enter more than one fax server URI, only one URI will be used by Client Access and Mailbox servers running UM services.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* switch specifies whether you're prompted for confirmation before a UM mailbox policy is upgraded.
InformCallerOfVoiceMailAnalysis	Optional	System.Boolean	The *InformCallerOfVoiceMailAnalysis* parameter specifies whether the callers leaving the voice mails will be informed about the possibility of their voice mails being forwarded to Microsoft for analysis.
LogonFailuresBeforePI	Optional	Microsoft.Exchange.Da	The

NReset		ta.Unlimited	*LogonFailuresBeforePINReset* parameter specifies the number of sequential unsuccessful logon attempts before the mailbox PIN is automatically reset. To disable this feature, set this parameter to `Unlimited`. If this parameter isn't set to `Unlimited`, it must be set to less than the value of the *MaxLogonAttempts* parameter. The range is from 0 through 999. The default setting is 5.
MaxGreetingDuration	Optional	System.Int32	The *MaxGreetingDuration* parameter specifies the maximum greeting length. The range is from 1 through 10 minutes. The default value is 5 minutes.
MaxLogonAttempts	Optional	Microsoft.Exchange.Data.Unlimited	The *MaxLogonAttempts* parameter specifies the number of times users can try unsuccessfully to log on, in sequence, before the UM mailboxes are locked.

			The range is from 1 through 999. The default value is 15.
MinPINLength	Optional	System.Int32	The *MinPINLength* parameter specifies the minimum number of digits required in a PIN for UM-enabled users. The range is from 4 through 24. The default value is 6.
Name	Optional	System.String	The *Name* parameter specifies the display name for the UM mailbox policy. This setting is limited to 64 characters.
PINHistoryCount	Optional	System.Int32	The *PINHistoryCount* parameter specifies the number of previous PINs that are remembered and aren't allowed during a PIN reset. This number includes the first time that the PIN was set. The range is from 1 through 20. The default value is 5.
PINLifetime	Optional	Microsoft.Exchange.Data.Unlimited	The *PINLifetime* parameter specifies the number of days until a

			new password is required. The range is from 1 through 999. The default value is 60. If you specify `Unlimited`, the users' PINs won't expire.
ProtectAuthenticatedVoiceMail	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DRMProtectionOptions	The *ProtectAuthenticatedVoiceMail* parameter specifies whether Mailbox servers that answer Outlook Voice Access calls for UM-enabled users associated with the UM mailbox policy create protected voice mail messages. The default setting is `None`. This means that no protection is applied to voice mail messages. If the value is set to `Private`, only messages marked as private are protected. If the value is set to `All`, every voice mail message is protected.
ProtectedVoiceMailText	Optional	System.String	The *ProtectedVoiceMailText* parameter specifies the text included in the body

			part of the protected voice mail messages for UM-enabled users associated with the UM mailbox policy. This text can contain up to 512 characters.
ProtectUnauthenticatedVoiceMail	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.DRMProtectionOptions	The *ProtectUnauthenticatedVoiceMail* parameter specifies whether the Mailbox servers that answer calls for UM-enabled users associated with the UM mailbox policy create protected voice mail messages. This also applies when a message is sent from a UM auto attendant to a UM-enabled user associated with the UM mailbox policy. The default setting is `None`. This means that no protection is applied to voice mail messages. If the value is set to `Private`, only messages marked as private are protected. If the value is set to `All`, every voice

			mail message is protected.
RequireProtectedPlayOnPhone	Optional	System.Boolean	The *RequireProtectedPlayOnPhone* parameter specifies whether users associated with the UM mailbox policy can only use Play on Phone for protected voice mail messages or whether users can use multimedia software to play the protected message. The default value is `$false`. When set to `$false`, users are able to use both methods to listen to protected voice mail messages.
ResetPINText	Optional	System.String	The *ResetPINText* parameter specifies the text to be included in the PIN reset email message. This text is limited to 512 characters.
SourceForestPolicyNames	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *SourceForestPolicyNames* parameter specifies the name or names of the

			corresponding UM mailbox policy objects located in the source forest during a cross-forest migration.
UMDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the UM dial plan associated with the UM mailbox policy.
UMEnabledText	Optional	System.String	The *UMEnabledText* parameter specifies the text to be included when a user is enabled for Unified Messaging. This text is limited to 512 characters.
VoiceMailPreviewPartnerAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *VoiceMailPreviewPartnerAddress* parameter specifies the SMTP address of a Voice Mail Preview partner that's contracted to provide transcription services for UM-enabled users in this UM mailbox policy. The default value is `$null`.
VoiceMailPreviewPartnerAssignedID	Optional	System.String	The *VoiceMailPreviewPartnerAssignedID* parameter

			specifies the identification string, if any, provided to the organization by the Voice Mail Preview partner that's contracted to provide transcription services for UM-enabled users in this UM mailbox policy. The default value is `$null`.
VoiceMailPreviewPartnerMaxDeliveryDelay	Optional	System.Int32	The *VoiceMailPreviewPartnerMaxDeliveryDelay* parameter specifies the number of seconds that a Mailbox server waits for a Voice Mail Preview partner system to return a message with a Voice Mail Preview. If this time is exceeded, the Mailbox server delivers the voice mail message without a preview. The default value is 1200. The minimum value for this parameter is 300.
VoiceMailPreviewPartnerMaxMessageDuration	Optional	System.Int32	The *VoiceMailPreviewPartnerMaxMessageDuration* parameter specifies the

			maximum duration, in seconds, of voice mail messages sent to the Voice Mail Preview partner that's contracted to provide transcription services for UM-enabled users in this UM mailbox policy. The default value is 180. The minimum number for this parameter is 60. This setting should be set equal to the maximum value allowed by the Voice Mail Preview partner.
VoiceMailText	Optional	System.String	The *VoiceMailText* parameter specifies the text to be included in the body part of voice mail messages. The parameter applies to call answering messages in addition to messages originated by an authenticated subscriber. This text is limited to 512 characters.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions

			that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Export-UMPrompt

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Export-UMPrompt** cmdlet to export an audio file being used as a greeting prompt for Unified Messaging (UM) dial plans and auto attendants.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Export-UMPrompt -BusinessHoursWelcomeGreetingAndMenu <SwitchParameter> -
UMAutoAttendant <UMAutoAttendantIdParameter> [-TestMenuKeyMapping
<CustomMenuKeyMapping[]>] <COMMON PARAMETERS>


Export-UMPrompt -BusinessHoursWelcomeGreeting <SwitchParameter> -
UMAutoAttendant <UMAutoAttendantIdParameter> [-TestBusinessName <String>]
<COMMON PARAMETERS>
```

```
Export-UMPrompt -AfterHoursWelcomeGreetingAndMenu <SwitchParameter> -
UMAutoAttendant <UMAutoAttendantIdParameter> [-TestMenuKeyMapping
<CustomMenuKeyMapping[]>] <COMMON PARAMETERS>


Export-UMPrompt -PromptFileName <String> -UMAutoAttendant
<UMAutoAttendantIdParameter> <COMMON PARAMETERS>


Export-UMPrompt -AfterHoursWelcomeGreeting <SwitchParameter> -
UMAutoAttendant <UMAutoAttendantIdParameter> [-TestBusinessName <String>]
<COMMON PARAMETERS>


Export-UMPrompt -BusinessHours <SwitchParameter> -UMAutoAttendant
<UMAutoAttendantIdParameter> <COMMON PARAMETERS>


Export-UMPrompt -BusinessLocation <SwitchParameter> -UMAutoAttendant
<UMAutoAttendantIdParameter> <COMMON PARAMETERS>


Export-UMPrompt -PromptFileName <String> -UMDialPlan
<UMDialPlanIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example exports the welcome greeting for the UM dial plan MyUMDialPlan and saves it as the file welcomegreeting.mp3.

```
$prompt = Export-UMPrompt -PromptFileName
"customgreeting.mp3" -UMDialPlan MyUMDialPlan
set-content -Path "d:\DialPlanPrompts\welcomegreeting.mp3"
-Value $prompt.AudioData -Encoding Byte
```

EXAMPLE 2

This example exports the business hours welcome greeting used for the UM auto attendant MyUMAutoAttendant and saves it as the file BusinessHoursWelcomeGreeting.mp3.

```
$prompt = Export-UMPrompt -BusinessHoursWelcomeGreeting -
UMAutoAttendant MyUMAutoAttendant
set-content -Path "d:\UMPrompts
\BusinessHoursWelcomeGreeting.mp3" -Value $prompt.AudioData
-Encoding Byte
```

EXAMPLE 3

This example exports a custom greeting for the UM auto attendant MyUMAutoAttendant and saves

it to the file welcomegreetingbackup.mp3.

```
Export-UMPrompt -PromptFileName "welcomegreeting.mp3" -
UMAutoAttendant MyUMAutoAttendant
set-content -Path "e:\UMPromptsBackup
\welcomegreetingbackup.mp3" -Value $prompt.AudioData -
Encoding Byte
```

EXAMPLE 4

This example exports the after hours welcome greeting for the UM auto attendant MyUMAutoAttendant, saves it as the file AfterHoursWelcomeGreeting.mp3, and uses Northwind Traders as the test business name.

```
Export-UMPrompt -AfterHoursWelcomeGreeting -UMAutoAttendant
MyUMAutoAttendant -TestBusinessName "Northwind Traders"
set-content -Path "d:\AfterHoursWelcomeGreeting.mp3" -Value
$prompt.AudioData -Encoding Byte
```

Detailed Description

The **Export-UMPrompt** cmdlet exports prompts that belong to existing UM dial plan and UM auto attendant objects. After the **Export-UMPrompt** cmdlet exports a prompt, you can save a copy of the prompt to a local drive as an audio file. You can then play the audio file using a media player.

After this task is completed, the UM prompts are displayed or saved.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
AfterHoursWelcomeGreeting	Required	System.Management.Automation.SwitchParameter	The *AfterHoursWelcomeGreeting* parameter specifies the system-generated after hours welcome greeting for

			the UM auto attendant specified.
AfterHoursWelcomeGreetingAndMenu	Required	System.Management.Automation.SwitchParameter	The *AfterHoursWelcomeGreetingAndMenu* parameter specifies the system-generated after hours welcome greeting and menu prompts for the UM auto attendant specified.
BusinessHours	Required	System.Management.Automation.SwitchParameter	The *BusinessHours* parameter specifies that the prompt to be returned is the business hours prompt of the auto attendant.
BusinessHoursWelcomeGreeting	Required	System.Management.Automation.SwitchParameter	The *BusinessHoursWelcomeGreeting* parameter specifies the system-generated business hours welcome greeting for the UM auto attendant specified.
BusinessHoursWelcomeGreetingAndMenu	Required	System.Management.Automation.SwitchParameter	The *BusinessHoursWelcomeGreetingAndMenu* parameter specifies the

			system-generated business hours welcome greeting and menu prompts for the UM auto attendant specified.
BusinessLocation	Required	System.Management.Automation.SwitchParameter	The *BusinessLocation* parameter specifies the business location greeting played for callers when the caller calls into a UM auto attendant and specifies the business location.
PromptFileName	Required	System.String	The *PromptFileName* parameter specifies the name of the custom prompt to export.
UMAutoAttendant	Required	Microsoft.Exchange.Configuration.Tasks.UMAutoAttendantIdParameter	The *UMAutoAttendant* parameter specifies the UM auto attendant ID. This parameter specifies the directory object identifier for the UM auto attendant.
UMDialPlan	Required	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the UM dial plan ID. This parameter specifies the directory object identifier for the UM

			dial plan.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
TestBusinessName	Optional	System.String	The *TestBusinessName* parameter specifies whether the business name configured on a UM auto attendant or the business name specified by this

			parameter is used to generate the welcome greeting prompt.
TestMenuKeyMapping	Optional	Microsoft.Exchange.Data.CustomMenuKeyMapping[]	The *TestMenuKeyMapping* parameter specifies whether the existing key mappings configured on a UM auto attendant or the key mapping menu specified by this parameter is used to generate the welcome greeting and menu.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Import-UMPrompt

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Import-UMPrompt** cmdlet to copy or upload a custom audio file to be used by Unified Messaging (UM) dial plans and auto attendants.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Import-UMPrompt -PromptFileName <String> -PromptFileStream <Stream> -
UMAutoAttendant <UMAutoAttendantIdParameter> <COMMON PARAMETERS>


Import-UMPrompt -PromptFileData <Byte[]> -PromptFileName <String> -
UMAutoAttendant <UMAutoAttendantIdParameter> <COMMON PARAMETERS>


Import-UMPrompt -PromptFileData <Byte[]> -PromptFileName <String> -
UMDialPlan <UMDialPlanIdParameter> <COMMON PARAMETERS>


Import-UMPrompt -PromptFileName <String> -PromptFileStream <Stream> -
UMDialPlan <UMDialPlanIdParameter> <COMMON PARAMETERS>


COMMON PARAMETERS: [-Confirm [<SwitchParameter>]] [-DomainController
<Fqdn>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example imports the welcome greeting file welcomegreeting.wav from d:\UMPrompts into the UM dial plan MyUMDialPlan.

```
[byte[]]$c = Get-content -Path "d:\UMPrompts
\welcomegreeting.wav" -Encoding Byte -ReadCount 0
Import-UMPrompt -UMDialPlan MyUMDialPlan -PromptFileName
```

```
"welcomegreeting.wav" -PromptFileData $c
```

EXAMPLE 2

This example imports the welcome greeting file welcomegreeting.wav from d:\UMPrompts into the UM auto attendant MyUMAutoAttendant.

```
[byte[]]$c = Get-content -Path "d:\UMPrompts
\welcomegreeting.wav" -Encoding Byte -ReadCount 0
Import-UMPrompt -UMAutoAttendant MyUMAutoAttendant -
PromptFileName "welcomegreeting.wav" -PromptFileData $c
```

EXAMPLE 3

This example imports the welcome greeting file AfterHoursWelcomeGreeting.wav from d:\UMPrompts into the UM auto attendant MyUMAutoAttendant.

```
[byte[]]$c = Get-content -Path "d:\UMPrompts
\AfterHoursWelcomeGreeting.wav" -Encoding Byte -ReadCount 0
Import-UMPrompt -UMAutoAttendant MyUMAutoAttendant -
PromptFileName "AfterHoursWelcomeGreeting.wav" -
PromptFileData $c
```

Detailed Description

The **Import-UMPrompt** cmdlet imports custom greeting audio files into UM dial plans and auto attendants. There are many custom greetings used by UM dial plans and auto attendants including welcome greetings for dial plans and after hours welcome greetings and menus, business hours and non-business hours welcome greetings and menus, and key mappings for UM auto attendants.

After this task is completed, the custom audio file can be used by a UM dial plan or auto attendant.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "UM mailboxes" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
PromptFileData	Required	System.Byte[]	The *PromptFileData* parameter specifies the byte array of the

			custom prompt.
PromptFileName	Required	System.String	The *PromptFileName* parameter specifies the name of the custom prompt.
PromptFileStream	Required	System.IO.Stream	The *PromptFileStream* parameter specifies whether the audio file will be uploaded or imported as an audio stream and not a byte array. The default setting is for the audio file to imported as a byte array.
UMAutoAttendant	Required	Microsoft.Exchange.Configuration.Tasks.UMAutoAttendantIdParameter	The *UMAutoAttendant* parameter specifies the UM auto attendant ID. This parameter specifies the directory object identifier for the UM auto attendant.
UMDialPlan	Required	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *UMDialPlan* parameter specifies the UM dial plan ID. This parameter specifies the directory object identifier for the UM dial plan.
Confirm	Optional	System.Management.A	The *Confirm* switch can

		utomation.SwitchParameter	be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-UMService

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-UMService** cmdlet to set the status of an Exchange 2007 or Exchange 2010 Unified Messaging server to disabled. This prevents the UM server from processing UM incoming calls.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-UMService -Identity <UMServerIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Immediate <$true |
$false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables Unified Messaging on the UM server MyUMServer but doesn't disconnect calls that are being processed.

```
Disable-UMService -Identity MyUMServer
```

EXAMPLE 2

This example disables Unified Messaging on the UM server MyUMServer and disconnects all calls being processed.

```
Disable-UMService -Identity MyUMServer -Immediate $true
```

Detailed Description

The **Disable-UMService** cmdlet sets the status of a UM server. A UM server has a logical status variable controlled using the enable and disable cmdlets. A UM server won't process any new calls unless it's in the enabled state. With this status variable, you can start or stop call processing on a UM server so the UM server can be brought online or taken offline in a controlled way.

Warning:
This cmdlet only is available for Exchange 2007 and Exchange 2010 servers running the Unified Messaging server role and is not available for Exchange 2013 Client Access and Mailbox servers.

After this task is completed, the UM server can no longer:

- Answer any incoming calls.
- Respond to Play on Phone requests from a UM server.
- Be used to manage UM-enabled mailboxes.
- Be queried when a diagnostic task is used.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Server (UM service)" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMServerIdParameter	The *Identity* parameter specifies the identifier for the UM server being disabled. This is the directory object ID for the UM server.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and

			requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Immediate	Optional	System.Boolean	The *Immediate* parameter specifies whether the UM server drops all current calls or enables current calls to finish. If this parameter is set to `$true`, all calls that are currently connected are disconnected.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the

			WhatIf switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-UMService

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-UMService** cmdlet to set the status of a Microsoft Exchange Server 2007 or Exchange Server 2010 Unified Messaging server to enabled. This setting enables the Unified Messaging server to process UM calls.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-UMService -Identity <UMServerIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables Unified Messaging on the UM server MyUMServer.

```
Enable-UMService -Identity MyUMServer
```

Detailed Description

The **Enable-UMService** cmdlet sets the status of an Exchange server running the Unified Messaging server role. A UM server has a logical status variable controlled using the enable and disable cmdlets. A UM server won't process any new calls unless it's in the enabled state. With the status variable, you can start or stop call processing on a UM server so the UM server can be brought online or taken offline in a controlled way.

After this task is completed, the UM server is available to answer incoming calls.

Warning:
This cmdlet only is available for Exchange 2007 and Exchange 2010 servers running the Unified Messaging server role, and it's not available on Exchange 2013 Client Access and Mailbox servers.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Server (UM service)" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMServerIdParameter	The *Identity* parameter specifies the identifier for the UM server being enabled. This is the directory object ID for the Mailbox server.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the

			command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-UMService

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Get-UMService** cmdlet to display the properties for a single Mailbox server that's running the Microsoft Exchange Unified Messaging service or to display a list of Mailbox servers that are running the Microsoft Exchange Unified Messaging service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-UMService [-Identity <UMServerIdParameter>] [-DomainController <Fqdn>]
```

Examples

EXAMPLE 1

This example displays a list of all the Mailbox servers in the Active Directory forest.

```
Get-UMService
```

EXAMPLE 2

This example displays a formatted list of properties for the Mailbox server MyUMServer.

```
Get-UMService -Identity MyUMServer | Format-List
```

Detailed Description

The **Get-UMService** cmdlet retrieves the properties for a Mailbox server that's running the Microsoft Exchange Unified Messaging service or returns a list of available Mailbox servers from Active Directory. When the cmdlet is used for a single Mailbox server, it returns the Mailbox server properties including **MaxCalls**, **MaxFaxCalls**, and **UMDialPlans**. The properties and their values for

the Mailbox server are stored in the Unified Messaging section of the Exchange Server configuration object in Active Directory.

The **ExchangeVersion** attribute that's returned is the minimum version of Microsoft Exchange you can use to manage the returned object. This attribute isn't the same as the version of Microsoft Exchange that's displayed in the Exchange Administration Center when you select **Server Configuration**.

After this task is completed, you can view the parameters and values for a single Mailbox server or a list of all of the Mailbox servers in Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox Server (UM service)" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.UMServerIdParameter	The *Identity* parameter specifies the name of the Mailbox server that's running the Microsoft Exchange Unified Messaging service. If this parameter isn't supplied, a list of all Mailbox servers is returned.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-UMService

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Set-UMService** cmdlet to set the properties on a Microsoft Exchange Server 2013 Mailbox server or Exchange Server 2007 or Exchange Server 2010 Unified Messaging server that's running the Microsoft Exchange Unified Messaging (UM) service.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-UMService -Identity <UMServerIdParameter> [-Confirm
[<SwitchParameter>]] [-DialPlans <MultiValuedProperty>] [-DomainController
<Fqdn>] [-ExternalHostFqdn <UMSmartHost>] [-ExternalServiceFqdn
<UMSmartHost>] [-GrammarGenerationSchedule <ScheduleInterval[]>] [-
IPAddressFamily <IPv4Only | IPv6Only | Any>] [-IPAddressFamilyConfigurable
<$true | $false>] [-IrmLogEnabled <$true | $false>] [-IrmLogMaxAge
<EnhancedTimeSpan>] [-IrmLogMaxDirectorySize <Unlimited>] [-
IrmLogMaxFileSize <ByteQuantifiedSize>] [-IrmLogPath <LocalLongFullPath>]
[-MaxCallsAllowed <Int32>] [-SIPAccessService
<ProtocolConnectionSettings>] [-SipTcpListeningPort <Int32>] [-
SipTlsListeningPort <Int32>] [-Status <Enabled | Disabled | NoNewCalls>]
[-UMForwardingAddressTemplate <String>] [-UMPodRedirectTemplate <String>]
[-UMStartupMode <TCP | TLS | Dual>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example performs the following actions:

- Enables the Microsoft Exchange Unified Messaging service on the Mailbox server MyUMServer to accept both IPv4 and IPv6 data packets.
- Enables the Microsoft Exchange Unified Messaging service on the Mailbox server MyUMServer to

start up using both TCP and TLS mode.

```
Set-UMService -Identity MyUMServer -IPAddressFamily Any -
UMStartupMode Dual
```

EXAMPLE 2

This example removes the Mailbox server MyUMServer from all UM dial plans.

```
Set-UMService -Identity MyUMServer -DialPlans $null
```

EXAMPLE 3

This example performs the following actions:

- Adds the Mailbox server MyUMServer to the UM dial plan MySIPDialPlan.
- Sets the maximum number of incoming calls to 50.
- Sets northamerica.lyncpoolna.contoso.com as the FQDN for the SIP access service that is used by Microsoft Lync Server for inbound and outbound calling from remote Lync clients.
- Enables the Microsoft Exchange Unified Messaging service on the Mailbox server MyUMServer to start up in TLS mode.

```
Set-UMService -Identity MyUMServer -DialPlans MySIPDialPlan
-MaxCallsAllowed 50 -SipAccessService
northamerica.lyncpoolna.contoso.com -UMStartupMode TLS
```

Detailed Description

The **Set-UMService** cmdlet sets specific properties on an Exchange 2013 Mailbox server or Exchange 2007 or Exchange 2010 Unified Messaging server that's running the Microsoft Exchange Unified Messaging service. This cmdlet can be used to set individual Unified Messaging parameters for a specified Mailbox or Unified Messaging server. After this task is completed, the cmdlet sets the parameters and the values specified.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Mailbox server (UM service)" entry in the Unified Messaging permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UMS	The *Identity* parameter specifies the ID for the

		erverIdParameter	Exchange 2013 Mailbox or Exchange 2007 or Exchange 2010 Unified Messaging server object to be configured that's running the Microsoft Exchange Unified Messaging service. This parameter specifies the directory object ID for the Mailbox server.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DialPlans	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *DialPlans* parameter specifies all dial plans for which an Exchange 2013 Client Access or Mailbox server or Exchange 2007 or Exchange 2010 Unified Messaging server handles incoming calls. Exchange 2013 Client Access and Mailbox servers can't be associated with a TelExt or

			E.164 UM dial plan but can be associated or added to SIP dial plans. If you're integrating Unified Messaging with Microsoft Office Communications Server 2007 R2, Lync Server 2010, or Lync Server 2013, you must associate or add all Client Access and Mailbox servers to SIP dial plans. This parameter can also be used to associate Exchange 2007 or Exchange 2010 Unified Messaging servers to a UM dial plan. To enter multiple values and overwrite any existing entries, use the following syntax: `<value1>,<value2>...`. If the values contain spaces or otherwise require quotation marks, you need to use the following syntax: `"<value1>","<value2>"...` To add or remove one or more values without affecting any existing

			entries, use the following syntax: `@{Add="<value1>","<value2>"...; Remove="<value1>","<value2>"...}`.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExternalHostFqdn	Optional	Microsoft.Exchange.Data.UMSmartHost	This parameter is reserved for internal Microsoft use.
ExternalServiceFqdn	Optional	Microsoft.Exchange.Data.UMSmartHost	This parameter is reserved for internal Microsoft use.
GrammarGenerationSchedule	Optional	Microsoft.Exchange.Common.ScheduleInterval[]	The *GrammarGenerationSchedule* parameter specifies the scheduled times to start speech grammar generation. This parameter allows only one start time per day. The default scheduled time for grammar generation is 02:00–02:30 local time each day.
IPAddressFamily	Optional	Microsoft.Exchange.Data.Directory.IPAddress	The *IPAddressFamily* parameter specifies

		Family	whether the UM IP gateway will use Internet Protocol version 4 (IPv4), IPv6, or both to communicate. If set to `IPv4Only`, the UM IP gateway only uses IPv4 to communicate. If set to `IPv6Only`, the UM IP gateway only uses IPv6. If set to `Any`, IPv6 is used first, and then if necessary, it falls back to IPv4. The default is `IPv4Only`.
IPAddressFamilyConfigurable	Optional	System.Boolean	The *IPAddressFamilyConfigurable* parameter specifies whether you're able to set the *IPAddressFamily* parameter to `IPv6Only` or `Any`. The default is `$true`.
IrmLogEnabled	Optional	System.Boolean	The *IrmLogEnabled* parameter specifies whether to enable logging of Information Rights Management (IRM) transactions. IRM logging is enabled by default. Values include: • `$true` Enable IRM logging • `$false` Disable IRM logging

IrmLogMaxAge	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *IrmLogMaxAge* parameter specifies the maximum age for the IRM log file. Log files that are older than the specified value are deleted. The default value is 30 days. To specify a value, enter it as a time span: dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. The valid input range for this parameter is from 00:00:00 through 24855.03:14:07. Setting the value of the *IrmLogMaxAge* parameter to 00:00:00 prevents the automatic removal of IRM log files because of their age.
IrmLogMaxDirectorySize	Optional	Microsoft.Exchange.Data.Unlimited	The *IrmLogMaxDirectorySize* parameter specifies the maximum size of all IRM logs in the connectivity log directory. When a directory reaches its maximum file size, the server deletes the oldest log files first. The default

			value is 250 megabytes (MB). When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value of the *IrmLogMaxFileSize* parameter must be less than or equal to the value of the *IrmLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the connectivity log directory.
IrmLogMaxFileSize	Optional	Microsoft.Exchange.Data.ByteQuantifiedSize	The *IrmLogMaxFileSize* parameter specifies the maximum size of each IRM log file. When a log file reaches its maximum file size, a new log file is

			created. The default value is 10 MB. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. The value of the *IrmLogMaxFileSize* parameter must be less than or equal to the value of the *IrmLogMaxDirectorySize* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes. If you enter a value of `unlimited`, no size limit is imposed on the IRM log files.
IrmLogPath	Optional	Microsoft.Exchange.Data.LocalLongFullPath	The *IrmLogPath* parameter specifies the default IRM log directory location. The default value is `C:\Program Files\Microsoft\Exchange`

			`Server\V15`. If you set the value of the *IrmLogPath* parameter to `$null`, you effectively disable IRM logging. However, if you set the value of the *IrmLogPath* parameter to `$null` when the value of the *IrmLogEnabled* attribute is `$true`, Exchange will log errors in the Application event log. The preferred way for disabling IRM logging is to set the *IrmLogEnabled* parameter to `$false`.
MaxCallsAllowed	Optional	System.Int32	The *MaxCallsAllowed* parameter specifies the maximum number of concurrent voice calls that an Exchange 2013 Mailbox server or Exchange 2013 Mailbox or Exchange 2007 or Exchange 2010 Unified Messaging server that's running the Microsoft Exchange Unified Messaging service allows.
SIPAccessService	Optional	Microsoft.Exchange.Data.ProtocolConnectionSettings	The *SIPAccessService* parameter specifies the FQDN and Transmission

			Control Protocol (TCP) port of the nearest Lync Server pool location for inbound and outbound calls from remote Lync users located outside of the network. When this parameter isn't set, the Mailbox server running the Microsoft Exchange Unified Messaging service may select a Lync pool for Real-Time Transport Protocol (RTP) media traffic that isn't the closest geographically to the remote user. This parameter is optional when you're configuring Unified Messaging with single Lync Server pool deployments. However, for Lync Server deployments that span multiple geographic regions, it's recommended that you specify this parameter. This parameter is set on a per-Mailbox server running the Microsoft Exchange Unified Messaging service basis

			and must point to the Lync Server pool that is located the closest geographically to the Mailbox server.
SipTcpListeningPort	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
SipTlsListeningPort	Optional	System.Int32	This parameter is reserved for internal Microsoft use.
Status	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.ServerStatus	The *Status* parameter will be removed in future versions of the product.
UMForwardingAddressTemplate	Optional	System.String	This parameter is reserved for internal Microsoft use.
UMPodRedirectTemplate	Optional	System.String	This parameter is reserved for internal Microsoft use.
UMStartupMode	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UMStartupMode	The *UMStartupMode* parameter specifies whether the Microsoft Exchange Unified Messaging service on an Exchange 2013 Mailbox server or Exchange 2007 or Exchange 2010 Unified Messaging server will start up in TCP, TLS, or Dual mode. If the Mailbox or Unified Messaging server is being added to

			UM dial plans that have different security settings, you should choose Dual mode. In Dual mode, the Mailbox or Unified Messaging server can listen on ports 5060 and 5061 at the same time. If the startup mode is changed, the Microsoft Exchange Unified Messaging service must be restarted.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Test-ExchangeUMCallFlow

Exchange Management Shell > Exchange 2013 cmdlets > Unified Messaging cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-04-04*

Note:

The **Test-ExchangeUMCallFlow** cmdlet (the UM Troubleshooting Tool) isn't included in Microsoft Exchange Server 2013 or Microsoft Exchange Server 2010. You need to download the UM Troubleshooting Tool from the Microsoft Download Center. For more information, see Unified Messaging Troubleshooting Tool.

Use the **Test-ExchangeUMCallFlow** cmdlet (the UM Troubleshooting Tool) to test call flow between Client Access servers running the Microsoft Exchange Unified Messaging Call Router service, Mailbox servers running the Microsoft Exchange Unified Messaging service, VoIP gateways, IP PBXs, Session Initiation Protocol (SIP) servers and Microsoft Lync Server. The **Test-ExchangeUMCallFlow** cmdlet can be used to diagnose configuration errors found in telephony components, Exchange 2010 SP1 or later or Exchange 2013 Unified Messaging settings, and connectivity issues between on-premises and hybrid Unified Messaging deployments.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Test-ExchangeUMCallFlow [-Mode <Gateway | SIPClient>] [-VoIPSecurity
<Unsecured | SIPSecured | Secured>] [-CertificateThumbprint <string>] [-
NextHop <string>] [-Diversion <string>] [-HuntGroup <string>]


Test-ExchangeUMCallFlow [-Mode <Gateway | SIPClient>] [-NextHop <string>]
[-CalledParty <string>] [-CallingParty <string>] [-Credential
<PSCredential>]
```

Examples

EXAMPLE 1

This example uses the `Gateway` mode and tests the call flow in a non-Lync Server environment. This example sets the VoIP security mode to `Unsecured`, uses the IP address 10.1.1.1 as the next hop, and includes an extension number in the diversion information.

```
Test-ExchangeUMCallFlow -Mode Gateway -VoIPSecurity
Unsecured -NextHop 10.1.1.1 -Diversion 12345
```

EXAMPLE 2

This example uses the `SIPClient` mode and tests the call flow with a Secured UM dial plan in an environment that contains servers running Lync Server. By default, when you run the cmdlet, the cmdlet uses the credentials of the user currently logged onto the computer.

```
Test-ExchangeUMCallFlow -Mode SIPClient -CallingParty
tony@contoso.com -CalledParty david@contoso.com -Credential
$get
```

EXAMPLE 3

This example uses the `Gateway` mode and tests the call flow in a non-Lync Server environment. This example sets the VoIP security mode to `Secured`, uses the IP address 10.176.10.194 as the next hop, and includes diversion information.

```
Test-ExchangeUMCallFlow -Mode Gateway -VoIPSecurity Secured
-CertificateThumbprint
a909502dd82ae41433e6f83886b00d4277a32a7b -NextHop
gateway.contoso.com -HuntGroup 10000 -Diversion "History-
Info: <sip:10001@10.176.10.194;user=phone?Reason=SIP%
3Bcause%3D487%3Btext%
3DTimeout>;index=1,<sip:10000@10.176.10.194;user=phone?
Reason=SIP>;index=1.1"
```

Detailed Description

The UM Troubleshooting Tool is an Exchange Management Shell cmdlet named **Test-ExchangeUMCallFlow**. You can use this cmdlet to diagnose configuration errors specific to call answering scenarios to test whether voice mail is functioning correctly in both on-premises and hybrid UM deployments. The **Test-ExchangeUMCallFlow** cmdlet only supports testing of call answering scenarios; however, it can't currently be used to test the following incoming call scenarios:

- Incoming calls to a UM auto attendant.
- Incoming calls to an Outlook Voice Access number as an unauthenticated user.
- Incoming calls to an Outlook Voice Access number as an authenticated Outlook Voice Access user.

You can use this cmdlet in deployments with Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server or in UM deployments with VoIP gateways or IP PBXs.

This cmdlet emulates calls and runs a series of diagnostic tests that help on-premises administrators to identify configuration errors in telephony equipment, Exchange 2010 SP1 and Exchange 2013 Unified Messaging settings, and connectivity issues between on-premises and hybrid deployments of Exchange 2010 SP1 and Exchange 2013 Unified Messaging.

When you run the cmdlet, it states the reason and possible solutions for issues that have been detected. It also outputs general audio quality metrics for diagnosing audio quality issues related to network connectivity such as jitter and average packet loss. The **Test-ExchangeUMCallFlow** cmdlet supports testing UM components and services in `Secured`, `SIP Secured`, and `Unsecured` calls, and it can be run either in `Gateway` or `SIPClient` modes.

Important:
The **Test-ExchangeUMCallFlow** cmdlet must be used to test only the voice mail functionality of a server running the Exchange 2010 SP1 UM server role and a Mailbox server running the Microsoft Exchange Unified Messaging service with Exchange 2013 installed.

The **Test-ExchangeUMCallFlow** cmdlet can be installed on a computer running the Exchange 2010 SP1 or later UM server role or a Mailbox server running Exchange 2013 or on another 64-bit computer running:

- Either the Windows 7 or Windows Vista operating systems.
- Either the Windows Server 2008 or Windows Server 2008 R2 operating systems.

Prior to installing the UM Troubleshooting Tool, the following components must be installed on a 64-bit version of Windows 7, Windows Vista, or the 64-bit edition of Windows Server 2008:

- Microsoft .NET Framework 3.5 SP1. For information, see Microsoft .NET Framework 3.5 Service Pack 1.
- Microsoft .NET Framework 3.5 Family Update for Windows Vista x64 and Windows Server 2008 x64 updates if the tool will be run on a computer running Windows Vista or Windows Server 2008. For more information, see Microsoft .NET Framework 3.5 Family Update for Windows Vista x64, and Windows Server 2008 x64.
- Windows Remote Management (WinRM) 2.0 and Windows PowerShell V2 (Windows6.0-KB968930.msu). For details, see Microsoft Knowledge Base article 968930, Windows Management Framework Core package (Windows PowerShell 2.0 and WinRM 2.0).
- Unified Communications Managed API 2.0, Core Runtime (64-bit) (UcmaRuntimeWebDownloadX64.msi). For information, see Unified Communications Managed API 2.0, Core Runtime (64-bit).

Parameters

Parameter	Required	Type	Description
CalledParty	Required	System.String	The *CalledParty* parameter specifies the SIP URI of the Microsoft Office Communications Server 2007 R2 or Microsoft Lync Server user that has been

			enabled for the Enterprise Voice user that the **Test-ExchangeUMCallFlow** cmdlet will make the voice call to, for example: `-CalledParty tonysmith@contoso.com`. Use this parameter if you're running the tool in `SIPClient` mode.
CallingParty	Required	System.String	The *CallingParty* parameter specifies the SIP URI of the Office Communications Server 2007 R2 or Lync Server user that has been enabled for the Enterprise Voice user who's making the incoming call, for example: `-CallingParty tonysmith@contoso.com`. Use this parameter if you're running the tool in `SIPClient` mode.
Diversion	Required	System.String	The *Diversion* parameter specifies the string that should be sent as diversion information for the incoming call. This can be in the form of a Diversion or History-Info header. The diversion information can be either

			an extension number or also include additional diversion information. When you provide diversion information as a History-Info header, verify the following: • There are at least two different entries with different user parts. • The last entry should contain the user's associated UM dial plan pilot number. • The second to last entry should include the UM-enabled user's extension number. This entry must also include the appropriate Reason text. This text must be properly escaped in accordance with standard URL parameter escaping rules.
Mode	Required	Microsoft.Exchange.UM.TroubleshootingTool.TestMode	The *Mode* parameter specifies whether the deployment being tested includes VoIP gateways, IP PBX, or Office Communications Server R2 or Lync servers. You

			can specify either `Gateway` mode when your UM deployment includes VoIP gateways or IP PBXs or `SIPClient` mode if your UM deployment includes Office Communications Server 2007 R2 or Lync Server.
NextHop	Required	System.String	The *NextHop* parameter specifies the IP address or fully qualified domain name (FQDN) and can also include the TCP port of the next hop that the **Test-ExchangeUMCallFlow** cmdlet must connect to while emulating the VoIP gateway or IP PBX. When you include the TCP port, you must include either port 5060 for `Unsecured` mode or port 5061 for `Secured` or `SIPSecured` mode, for example: `gateway.contoso.com:5061` If you're using the cmdlet in a hybrid environment, you must enter the FQDN of the Session Border Controller (SBC). If you're using the

			cmdlet in an on-premises environment, you must use the FQDN of an Exchange 2013 Mailbox server or Exchange 2010 SP1 UM server running the Microsoft Exchange Unified Messaging service.
CertificateThumbprint	Optional	System.String	The *CertificateThumbprint* parameter specifies the thumbprint of the certificate used for Transport Layer Security (TLS). This is required if either the `SIPSecured` or `Secured` mode is configured on the UM dial plan. This certificate thumbprint is the certificate that was exported from the VoIP gateway, IP Private Branch eXchange (PBX), or Session Border Controller (SBC). Also, the computer that has the UM Troubleshooting Tool installed and is being used to test for call flow must trust the next hop's certificate of authority.

Credential	Optional	System.String	The *Credential* parameter specifies the credentials that will be used to run the cmdlet.
HuntGroup	Optional	System.String	The *HuntGroup* parameter specifies the UM hunt group associated with the VoIP gateway being emulated. This is typically an extension number. Use this parameter if you're running the tool in `Gateway` mode.
VoIPSecurity	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UMVoIPSecurityType	The *VoIPSecurity* parameter specifies the security mode when using the cmdlet in `Gateway` mode. You can use one of the following Voice over IP (VoIP) security modes: • `Secured` (TLS/SRTP) • `Unsecured` (TCP/RTP) (default) • `SIPSecured` (TLS/RTP)

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Users and Groups Cmdlets

Topic Last Modified: *2014-04-28*

User cmdlets

Get-Recipient

Get-SecurityPrincipal

Get-User

Set-User

Mail user cmdlets

Disable-MailUser

Enable-MailUser

Get-MailUser

New-MailUser

Remove-MailUser

Set-MailUser

Mail contact cmdlets

Get-Contact

Set-Contact

Disable-MailContact

Enable-MailContact

Get-MailContact

New-MailContact

Remove-MailContact

Set-MailContact

Group cmdlets

Disable-DistributionGroup

Enable-DistributionGroup

Get-DistributionGroup

New-DistributionGroup

Remove-DistributionGroup

Set-DistributionGroup

Add-DistributionGroupMember

Get-DistributionGroupMember

Remove-DistributionGroupMember

Update-DistributionGroupMember

Get-DynamicDistributionGroup

New-DynamicDistributionGroup

Remove-DynamicDistributionGroup

Set-DynamicDistributionGroup

Get-Group

Set-Group

Get-Contact

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-Contact** cmdlet to retrieve information on a specified contact or contacts.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-Contact [-Anr <String>] <COMMON PARAMETERS>

Get-Contact [-Identity <ContactIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-Filter <String>]
[-IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-ReadFromDomainController
```

```
<SwitchParameter>] [-RecipientTypeDetails <RecipientTypeDetails[]>] [-
ResultSize <Unlimited>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example retrieves the contact Markus Breyer using the *Identity* parameter and pipelines the object to the **Format-List** command to display the information about the contact.

```
Get-Contact -Identity MarkusBreyer | Format-List
```

EXAMPLE 2

This example uses the *Anr* parameter to retrieve all mail-enabled contacts whose names start with Markus.

```
Get-Contact -Anr Markus* -RecipientTypeDetails MailContact
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an

			attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the user name and password to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that

			retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ContactIdParameter	The *Identity* parameter specifies the contact. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-

			4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController*

			parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter returns objects only from the specified organizational unit (OU).
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain

			controller in the user's domain. If you set the recipient scope to include all recipients in the forest, and if you don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your Exchange servers.
RecipientTypeDetails	Optional	Microsoft.Exchange.Data.Directory.Recipient.RecipientTypeDetails[]	The *RecipientTypeDetails* parameter specifies the type of recipients returned. Recipient types are divided into recipient types and subtypes. Each recipient type contains all common properties for all subtypes. For example, the type UserMailbox represents a user account in Active Directory that has an associated mailbox. Because there are several mailbox types, each mailbox type is identified

			by the *RecipientTypeDetails* parameter. For example, a conference room mailbox has *RecipientTypeDetails* set to ConferenceRoomMailbox, whereas a user mailbox has *RecipientTypeDetails* set to UserMailbox. You can select from the following values: • `ArbitrationMailbox` • `ConferenceRoomMailbox` • `Contact` • `DiscoveryMailbox` • `DynamicDistributionGroup` • `EquipmentMailbox` • `ExternalManagedContact` • `ExternalManagedDistributionGroup` • `LegacyMailbox` • `LinkedMailbox` • `MailboxPlan` • `MailContact` • `MailForestContact` • `MailNonUniversalGroup` • `MailUniversalDistributionGroup` • `MailUniversalSecurityGroup` • `MailUser` • `PublicFolder` • `RoleGroup` • `RoomList` • `RoomMailbox` • `SharedMailbox` • `SystemAttendantMailbox` • `SystemMailbox` • `User` • `UserMailbox`
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to

			return.
SortBy	Optional	System.String	The *SortBy* parameter specifies to sort by a single attribute in ascending order.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-Contact

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-Contact** cmdlet to modify the settings of an existing contact.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-Contact -Identity <ContactIdParameter> [-AllowUMCallsFromNonUsers
<None | SearchEnabled>] [-AssistantName <String>] [-City <String>] [-
Company <String>] [-Confirm [<SwitchParameter>]] [-CountryOrRegion
<CountryInfo>] [-CreateDTMFMap <$true | $false>] [-Department <String>] [-
DisplayName <String>] [-DomainController <Fqdn>] [-Fax <String>] [-
FirstName <String>] [-GeoCoordinates <GeoCoordinates>] [-HomePhone
<String>] [-IgnoreDefaultScope <SwitchParameter>] [-Initials <String>] [-
LastName <String>] [-Manager <UserContactIdParameter>] [-MobilePhone
<String>] [-Name <String>] [-Notes <String>] [-Office <String>] [-OtherFax
<MultiValuedProperty>] [-OtherHomePhone <MultiValuedProperty>] [-
OtherTelephone <MultiValuedProperty>] [-Pager <String>] [-Phone <String>]
[-PhoneticDisplayName <String>] [-PostalCode <String>] [-PostOfficeBox
<MultiValuedProperty>] [-SeniorityIndex <Int32>] [-SimpleDisplayName
<String>] [-StateOrProvince <String>] [-StreetAddress <String>] [-
TelephoneAssistant <String>] [-Title <String>] [-UMCallingLineIds
<MultiValuedProperty>] [-UMDtmfMap <MultiValuedProperty>] [-WebPage
<String>] [-WhatIf [<SwitchParameter>]] [-WindowsEmailAddress
```

<SmtpAddress>]

Examples

EXAMPLE 1

This example makes the following changes to the existing contact Arlene Huff in the Users container in the Active Directory domain contoso.com:

- Change the *City* parameter value to Seattle.
- Change the *Company* parameter value to Contoso.

```
Set-Contact -Identity "contoso.com/Users/Arlene Huff" -City "Seattle" -Company "Contoso"
```

Detailed Description

You can use the **Set-Contact** cmdlet to modify the settings of contact objects that are visible in Active Directory Users and Computers. If the contact is mail-enabled, use the **Set-MailContact** cmdlet to modify the contact's email settings that aren't available by using the **Set-Contact** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ContactIdParameter	The *Identity* parameter specifies the object that you want to modify. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips

			• Distinguished Name (DN) Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • Domain\Account Example: Atlanta\JPhillips • GUID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • Immutable ID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • Legacy Exchange DN Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • SMTP Address Example: Jeff.Phillips@contoso.com • User Principal Name Example: JPhillips@contoso.com
AllowUMCallsFromNonUsers	Optional	Microsoft.Exchange.Data.Directory.Recipient.AllowUMCallsFromNonUsersFlags	The *AllowUMCallsFromNonUsers* parameter specifies whether to exclude the contact from directory searches.
AssistantName	Optional	System.String	The *AssistantName* parameter specifies the name of the contact's assistant.

City	Optional	System.String	The *City* parameter specifies the contact's city.
Company	Optional	System.String	The *Company* parameter specifies the contact's company.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
CountryOrRegion	Optional	Microsoft.Exchange.Data.Directory.CountryInfo	The *CountryOrRegion* parameter specifies the contact's country or region.
CreateDTMFMap	Optional	System.Boolean	The *CreateDTMFMap* parameter specifies that a dual-tone multiple-frequency (DTMF) map be created for the contact.
Department	Optional	System.String	The *Department* parameter specifies the contact's department.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the contact's name in the

			Exchange Administration Center and in the Exchange global address list (GAL). The *DisplayName* parameter has meaning only for mail-enabled objects.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Fax	Optional	System.String	The *Fax* parameter specifies the contact's fax number.
FirstName	Optional	System.String	The *FirstName* parameter specifies the contact's first name.
GeoCoordinates	Optional	Microsoft.Exchange.Data.GeoCoordinates	The *GeoCoordinates* parameter specifies the contact's physical location in latitude, longitude, and altitude coordinates. Use this parameter to specify the global position of the

			contact's location. You have to specify one of the following sets of coordinates; use semicolons to separate the values: • Latitude and longitude; for example, `"47.644125;-122.122411"` • Latitude, longitude, and altitude; for example, `"47.644125;-122.122411;161.432"`
HomePhone	Optional	System.String	The *HomePhone* parameter specifies the contact's home telephone number.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the

			IgnoreDefaultScope parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as GUID, aren't accepted.
Initials	Optional	System.String	The *Initials* parameter specifies the initials for the contact's name.
LastName	Optional	System.String	The *LastName* parameter specifies the contact's surname.
Manager	Optional	Microsoft.Exchange.Configuration.Tasks.UserContactIdParameter	The *Manager* parameter specifies the contact's manager.
MobilePhone	Optional	System.String	The *MobilePhone* parameter specifies the contact's primary mobile phone number.
Name	Optional	System.String	The *Name* parameter specifies the contact's name in Active Directory

			Users and Computers, and also in the **Group Name** field in the Exchange Administration Center if the group is mail-enabled. The *Name* value can't exceed 64 characters.
Notes	Optional	System.String	The *Notes* parameter specifies additional information about the contact.
Office	Optional	System.String	The *Office* parameter specifies the contact's physical office name or number.
OtherFax	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *OtherFax* parameter specifies the contact's alternative fax number.
OtherHomePhone	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *OtherHomePhone* parameter specifies the contact's alternative home telephone number.
OtherTelephone	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *OtherTelephone* parameter specifies the contact's alternative office telephone number.
Pager	Optional	System.String	The *Pager* parameter specifies the contact's pager number.

Phone	Optional	System.String	The *Phone* parameter specifies the contact's office telephone number.
PhoneticDisplayName	Optional	System.String	The *PhoneticDisplayName* parameter specifies a phonetic pronunciation of the *DisplayName* parameter. The maximum length of this parameter value is 255 characters.
PostalCode	Optional	System.String	The *PostalCode* parameter specifies the contact's postal code.
PostOfficeBox	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *PostOfficeBox* parameter specifies the contact's post office box number.
SeniorityIndex	Optional	System.Int32	The *SeniorityIndex* parameter specifies the order in which this contact will display in a hierarchical address book. A contact with a value of 2 will display higher in an address book than a contact with a value of 1.
SimpleDisplayName	Optional	System.String	The *SimpleDisplayName* parameter is used to display an alternative

			description of the object when only a limited set of characters is permitted. This limited set of characters consists of ASCII characters 26 through 126, inclusively. The *SimpleDisplayName* parameter has meaning only for mail-enabled objects.
StateOrProvince	Optional	System.String	The *StateOrProvince* parameter specifies the contact's state or province.
StreetAddress	Optional	System.String	The *StreetAddress* parameter specifies the contact's physical address.
TelephoneAssistant	Optional	System.String	The *TelephoneAssistant* parameter specifies the telephone number of the contact's assistant.
Title	Optional	System.String	The *Title* parameter specifies the contact's title.
UMCallingLineIds	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UMCallingLineIds* parameter specifies telephone numbers or extensions that can be mapped to a Unified Messaging (UM)-enabled

			user. You can specify more than one telephone number for each user, separated by a comma. Values for this parameter must be less than 128 characters in length and may include an optional plus sign (+) that precedes the numbers. Each UM-enabled user must have a unique *UMCallingLineIds* parameter value.
UMDtmfMap	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UMDtmfMap* parameter specifies whether you want to create a user-defined DTMF map for the UM-enabled contact.
WebPage	Optional	System.String	The *WebPage* parameter specifies the contact's web page.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply

			any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsEmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *WindowsEmailAddress* parameter specifies the contact's email address stored in Active Directory.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-DistributionGroup

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-DistributionGroup** cmdlet to remove mail capabilities from a mail-enabled distribution group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-DistributionGroup -Identity <DistributionGroupIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
IgnoreDefaultScope <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example disables Distribution Group1.

```
Disable-DistributionGroup -Identity "Distribution Group1"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DistributionGroupIdParameter	The *Identity* parameter specifies the identity of the distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-

			5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-DistributionGroup

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-DistributionGroup** cmdlet to mail-enable an existing universal group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-DistributionGroup -Identity <GroupIdParameter> [-Alias <String>] [-
Confirm [<SwitchParameter>]] [-DisplayName <String>] [-DomainController
<Fqdn>] [-OverrideRecipientQuotas <SwitchParameter>] [-PrimarySmtpAddress
<SmtpAddress>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example enables the distribution group Distribution Group1.

```
Enable-DistributionGroup -Identity "Distribution Group1"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.GroupIdParameter	The *Identity* parameter specifies the identity of the distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID`

			Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Alias	Optional	System.String	The *Alias* parameter specifies the email alias of the group.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the display name of the distribution group. A display name is typically the same as the *Domain*

			\Account Name.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the primary SMTP address for the distribution group. By default, the primary SMTP address is generated based on the default email address policy. If you specify a primary SMTP address by using this parameter, the cmdlet sets the *EmailAddressPolicyEnabled* attribute of the distribution group to `$false`, and the email addresses of this distribution group won't be automatically updated

			based on email address policies.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DistributionGroup

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-DistributionGroup** cmdlet to query for existing distribution groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-DistributionGroup [-ManagedBy <GeneralRecipientIdParameter>] <COMMON
PARAMETERS>


Get-DistributionGroup [-Anr <String>] <COMMON PARAMETERS>


Get-DistributionGroup [-Identity <DistributionGroupIdParameter>] <COMMON
PARAMETERS>


COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-Filter <String>]
[-IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-ReadFromDomainController
<SwitchParameter>] [-RecipientTypeDetails <RecipientTypeDetails[]>] [-
ResultSize <Unlimited>] [-SortBy <String>] [-UsnForReconciliationSearch
<Int64>]
```

Examples

EXAMPLE 1

This example retrieves the group Marketing Reports and pipelines the object to the **Format-List** command to display the information about the distribution group.

```
Get-DistributionGroup -Identity "Marketing Reports" |
Format-List
```

EXAMPLE 2

This example retrieves all distribution groups whose names contain the string "marketing" and pipelines the object to the **Format-Table** command to display the distribution group names and who they're managed by.

```
Get-DistributionGroup -Anr marketing | Format-Table Name,
ManagedBy
```

Detailed Description

You can use distribution groups to create email distribution lists and security groups to assign permissions to shared resources. Distribution groups can be used only with email applications to send email messages to collections of users. You can use the **Get-DistributionGroup** cmdlet to query for existing distribution groups.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password for accessing Active Directory. The default is the current user's credentials. This parameter requires the creation and passing of a credential object. This credential object is

			created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.DistributionGroupIdParameter	The *Identity* parameter specifies the identity of the distribution group object. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example:

			Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope

			setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
ManagedBy	Optional	Microsoft.Exchange.Configuration.Tasks.Gene	The *ManagedBy* parameter indicates the

		ralRecipientIdParameter	DN of the user or contact that manages the group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com

Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter limits results to a specific organizational unit (OU) container.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest, and if you don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your Exchange servers.

RecipientTypeDetails	Optional	Microsoft.Exchange.Data.Directory.Recipient.RecipientTypeDetails[]	The *RecipientTypeDetails* parameter specifies the type of recipients returned. Recipient types are divided into recipient types and subtypes. Each recipient type contains all common properties for all subtypes. For example, the type UserMailbox represents a user account in Active Directory that has an associated mailbox. Because there are several mailbox types, each mailbox type is identified by the *RecipientTypeDetails* parameter. For example, a conference room mailbox has *RecipientTypeDetails* set to ConferenceRoomMailbox, whereas a user mailbox has *RecipientTypeDetails* set to UserMailbox. You can select from the following values: • `ArbitrationMailbox` • `ConferenceRoomMailbox` • `Contact` • `DiscoveryMailbox` • `DynamicDistributionGroup` • `EquipmentMailbox` • `ExternalManagedContact` • `ExternalManagedDistri`

			butionGroup • LegacyMailbox • LinkedMailbox • MailboxPlan • MailContact • MailForestContact • MailNonUniversalGroup • MailUniversalDistribution Group • MailUniversalSecurity Group • MailUser • PublicFolder • RoleGroup • RoomList • RoomMailbox • SharedMailbox • SystemAttendantMailbox • SystemMailbox • User • UserMailbox
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter indicates the maximum number of recipient objects returned.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute used to sort the results.
UsnForReconciliationSearch	Optional	System.Int64	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-DistributionGroup

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-DistributionGroup** cmdlet to create a distribution group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-DistributionGroup -Name <String> [-Alias <String>] [-
ArbitrationMailbox <MailboxIdParameter>] [-BypassNestedModerationEnabled
<$true | $false>] [-Confirm [<SwitchParameter>]] [-CopyOwnerToMember
<SwitchParameter>] [-DisplayName <String>] [-DomainController <Fqdn>] [-
ExternalDirectoryObjectId <String>] [-IgnoreNamingPolicy
<SwitchParameter>] [-ManagedBy <MultiValuedProperty>] [-
MemberDepartRestriction <Closed | Open | ApprovalRequired>] [-
MemberJoinRestriction <Closed | Open | ApprovalRequired>] [-Members
<MultiValuedProperty>] [-ModeratedBy <MultiValuedProperty>] [-
ModerationEnabled <$true | $false>] [-Notes <String>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-OverrideRecipientQuotas
<SwitchParameter>] [-PrimarySmtpAddress <SmtpAddress>] [-RoomList
<SwitchParameter>] [-SamAccountName <String>] [-
SendModerationNotifications <Never | Internal | Always>] [-Type
<Distribution | Security>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a mail-enabled universal security group that has the following parameters:

- The group name is Managers.
- The group is created in the Users container in the domain contoso.com.
- The group *SamAccountName* parameter value is Managers.
- The group type is Security.

```
New-DistributionGroup -Name "Managers" -OrganizationalUnit
"contoso.com/Users" -SamAccountName "Managers" -Type
"Security"
```

EXAMPLE 2

This example creates the distribution group ITDepartment and ignores the naming policy.

```
New-DistributionGroup -Name "ITDepartment" -
```

`IgnoreNamingPolicy`

Detailed Description

You can use the **New-DistributionGroup** cmdlet to create Active Directory group objects of the following types:

- Mail-enabled universal security group (USG)
- Universal distribution group

Distribution groups are used to consolidate groups of recipients into a single point of contact for email messages. Distribution groups can't be used to assign permissions to network resources in Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Name	Required	System.String	The *Name* parameter specifies the name for the new distribution group. The value specified in the *Name* parameter is also used for the *DisplayName* parameter if the *DisplayName* parameter isn't specified. The *Name* parameter value can't exceed 64 characters. **Note:** If a group naming policy is enforced, you need to follow the naming constraints specified in the

			DistributionGroupNameBlockedWordList and the *DistributionGroupNamingPolicy* parameters of the **Set-OrganizationConfig** cmdlet. If the values of the *Name* and *DisplayName* parameters are different: • The *Name* parameter specifies the distribution group name in Active Directory Users and Computers. • The *DisplayName* parameter specifies the distribution group name in the Exchange Administration Center (EAC) and in the Exchange global address list (GAL).
Alias	Optional	System.String	The *Alias* parameter specifies the alias of the distribution group. The *Alias* parameter is then used to generate the primary SMTP email address of the object. The value of the *Alias* parameter can't contain spaces. If the *Alias* parameter isn't specified, the value of the

			SamAccountName parameter is used to generate the primary SMTP email address, with any spaces converted to underscores.
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArbitrationMailbox* parameter specifies the mailbox used to manage the moderation process. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-

			4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
BypassNestedModerationEnabled	Optional	System.Boolean	The *BypassNestedModerationEnabled* parameter specifies whether to allow the parent group moderators to provide approval for any nested groups that are also moderated. If you set this parameter to `$true`, after a moderator approves a message sent to this distribution group, the message is automatically approved for any other moderated recipients that are members of this distribution group. The default value is `$false`. **Note:** This parameter can be set only by top-level

			organization and tenant administrators.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-confirm:$False`. You must include a colon (`:`) in the syntax.
CopyOwnerToMember	Optional	System.Management.Automation.SwitchParameter	The *CopyOwnerToMember* parameter specifies that a recipient specified in the *ManagedBy* parameter is also a member of the distribution group.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the name of the distribution group in the EAC and in the Exchange GAL. If the *DisplayName* parameter isn't specified, the value of the *Name* parameter is used for the *DisplayName* parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExternalDirectoryObjectId	Optional	System.String	This parameter is reserved for internal Microsoft use.
IgnoreNamingPolicy	Optional	System.Management.Automation.SwitchParameter	The *IgnoreNamingPolicy* parameter specifies that the naming policy defined in the *DistributionGroupNamingPolicy* parameter of the **Set-OrganizationConfig** cmdlet can be ignored. You don't need to specify a value with this parameter.
ManagedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ManagedBy* parameter specifies the name of the mailbox recipient or Active Directory user that appears in the **Managed by** tab of the Active Directory object. If this parameter isn't specified, the creator of the group is the owner.

			This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com To specify an Active Directory user, use one of

			the following values: • GUID • DN • UPN • *Domain\Account Name* The recipients specified with the *ManagedBy* parameter aren't members of the distribution group. If you want recipients specified in this parameter to be added as members of the distribution group, use the *CopyOwnerToMember* parameter.
MemberDepartRestriction	Optional	Microsoft.Exchange.Data.Directory.Recipient.MemberUpdateType	The *MemberDepartRestriction* parameter specifies the restrictions that you can put on recipients who want to depart the group membership. This parameter can take one of the following values: • `Open` • `Closed` • `ApprovalRequired` **Note:** Universal security groups can't use the `Open` value. By default, they're set to `Closed`.

MemberJoinRestriction	Optional	Microsoft.Exchange.Data.Directory.Recipient.MemberUpdateType	The *MemberJoinRestriction* parameter specifies the restrictions that you can put on recipients who want to join the group membership. This parameter can take one of the following values: • `Open` • `Closed` • `ApprovalRequired` **Note:** Universal security groups can't use the `Open` value. By default, they're set to `Closed`.
Members	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Members* parameter specifies the initial list of recipients or Active Directory users who are a part of this distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=c

			ontoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com To specify an Active Directory user, use one of the following values: • GUID • DN • UPN • *Domain\Account Name*
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users who are responsible for moderating the messages sent to this distribution group. To designate more than one

			user, separate the users by commas. The *ModeratedBy* parameter is required if you set the *ModerationEnabled* parameter to `$true`. If you leave this parameter blank and there is a user already specified as the manager of this distribution group, the *ModeratedBy* field is automatically set to the *ManagedBy* parameter of the distribution group. Otherwise, an error is returned.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable moderation for the distribution group. To ensure moderation, set this parameters to `$true`. To disable moderation, set this parameter to `$false`. The default value is `$false`.
Notes	Optional	System.String	The *Notes* parameter specifies additional information about the

			distribution group.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies the container where the distribution group is created.
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the primary return SMTP email address for the distribution group. This parameter has meaning only if the distribution group has multiple SMTP email addresses.
RoomList	Optional	System.Management.Automation.SwitchParameter	The *RoomList* parameter specifies that all members of the distribution group are room resource mailboxes. You can create a distribution list for an office building in your organization and add all the rooms in that building to the distribution group.

			Room list distribution groups are then used to generate a list of building locations in Microsoft Outlook 2010 so the user can select a building and get information about when rooms are available in that building, without having to add the rooms in that building individually.
SamAccountName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *SamAccountName* parameter specifies the name for clients of the object running older operating systems. The *SamAccountName* parameter is displayed in Active Directory. If the *Alias* parameter isn't specified, the following conditions are true: • The value of the *SamAccountName* parameter is used for the value of the *Alias* parameter, with any spaces converted to underscores.

			• The *SamAccountName* parameter, with any spaces converted to underscores, is used to generate the primary SMTP email address of the object.
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when they send a message to the moderated distribution group. You can specify one of the following values: • `Always` • `Internal` • `Never` Set this parameter to `Always` if you want notifications to be sent to all senders. Set this parameter to `Internal` if you want notifications to be sent only to the senders who are internal to your organization. Set this parameter to `Never` to disable all status notifications.

			Note: The senders are always notified if their message is rejected by the moderators, regardless of the value of the parameter. The default value is `Never`.
Type	Optional	Microsoft.Exchange.Management.RecipientTasks.GroupType	The *Type* parameter specifies the group type created in Active Directory. The group's scope is always Universal. Valid values are `Distribution` or `Security`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-DistributionGroup

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-DistributionGroup** cmdlet to delete an existing distribution group from Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-DistributionGroup -Identity <DistributionGroupIdParameter> [-
BypassSecurityGroupManagerCheck <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ForReconciliation
<SwitchParameter>] [-IgnoreDefaultScope <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes an existing distribution group Temporary Staff.

```
Remove-DistributionGroup -Identity "Temporary Staff"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DistributionGroupIdParameter	The *Identity* parameter specifies the distribution group that you want to remove. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com

			• User Principal Name Example: JPhillips@contoso.com
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* parameter specifies whether to bypass security checks and moderation for the member being added, if the specified distribution group is a moderated distribution group.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration

			change to Active Directory.
ForReconciliation	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN

			for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-DistributionGroup

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-DistributionGroup** cmdlet to modify the settings of an existing distribution group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-DistributionGroup -Identity <DistributionGroupIdParameter> [-
AcceptMessagesOnlyFrom <MultiValuedProperty>] [-
AcceptMessagesOnlyFromDLMembers <MultiValuedProperty>] [-
AcceptMessagesOnlyFromSendersOrMembers <MultiValuedProperty>] [-Alias
<String>] [-ArbitrationMailbox <MailboxIdParameter>] [-
BypassModerationFromSendersOrMembers <MultiValuedProperty>] [-
BypassNestedModerationEnabled <$true | $false>] [-
BypassSecurityGroupManagerCheck <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-CreateDTMFMap <$true | $false>] [-CustomAttribute1
<String>] [-CustomAttribute10 <String>] [-CustomAttribute11 <String>] [-
CustomAttribute12 <String>] [-CustomAttribute13 <String>] [-
CustomAttribute14 <String>] [-CustomAttribute15 <String>] [-
CustomAttribute2 <String>] [-CustomAttribute3 <String>] [-CustomAttribute4
<String>] [-CustomAttribute5 <String>] [-CustomAttribute6 <String>] [-
CustomAttribute7 <String>] [-CustomAttribute8 <String>] [-CustomAttribute9
<String>] [-DisplayName <String>] [-DomainController <Fqdn>] [-
EmailAddresses <ProxyAddressCollection>] [-EmailAddressPolicyEnabled
<$true | $false>] [-ExpansionServer <String>] [-ExtensionCustomAttribute1
<MultiValuedProperty>] [-ExtensionCustomAttribute2 <MultiValuedProperty>]
[-ExtensionCustomAttribute3 <MultiValuedProperty>] [-
ExtensionCustomAttribute4 <MultiValuedProperty>] [-
ExtensionCustomAttribute5 <MultiValuedProperty>] [-ForceUpgrade
<SwitchParameter>] [-GenerateExternalDirectoryObjectId <SwitchParameter>]
[-GrantSendOnBehalfTo <MultiValuedProperty>] [-
HiddenFromAddressListsEnabled <$true | $false>] [-IgnoreDefaultScope
<SwitchParameter>] [-IgnoreNamingPolicy <SwitchParameter>] [-MailTip
<String>] [-MailTipTranslations <MultiValuedProperty>] [-ManagedBy
<MultiValuedProperty>] [-MaxReceiveSize <Unlimited>] [-MaxSendSize
<Unlimited>] [-MemberDepartRestriction <Closed | Open | ApprovalRequired>]
[-MemberJoinRestriction <Closed | Open | ApprovalRequired>] [-ModeratedBy
<MultiValuedProperty>] [-ModerationEnabled <$true | $false>] [-Name
<String>] [-PrimarySmtpAddress <SmtpAddress>] [-RejectMessagesFrom
<MultiValuedProperty>] [-RejectMessagesFromDLMembers
<MultiValuedProperty>] [-RejectMessagesFromSendersOrMembers
<MultiValuedProperty>] [-ReportToManagerEnabled <$true | $false>] [-
ReportToOriginatorEnabled <$true | $false>] [-
RequireSenderAuthenticationEnabled <$true | $false>] [-RoomList
<SwitchParameter>] [-SamAccountName <String>] [-
SendModerationNotifications <Never | Internal | Always>] [-
SendOofMessageToOriginatorEnabled <$true | $false>] [-SimpleDisplayName
<String>] [-UMDtmfMap <MultiValuedProperty>] [-WhatIf [<SwitchParameter>]]
[-WindowsEmailAddress <SmtpAddress>]
```

Examples

EXAMPLE 1

This example changes the display name of an existing distribution group from Accounting to Accounting Group.

```
Set-DistributionGroup -Identity "Accounting" -DisplayName
"Accounting Group"
```

EXAMPLE 2

This example converts the Bldg34 Conf Rooms distribution group to a room list.

```
Set-DistributionGroup -Identity "Bldg34 Conf Rooms" -
RoomList
```

EXAMPLE 3

This example changes the name of an existing distribution group from Ed_DirectReports to Ayla_DirectReports and ignores the group naming policy.

```
Set-DistributionGroup -Identity Ed_DirectReports -Name
Ayla_DirectReports -IgnoreNamingPolicy
```

Detailed Description

Distribution groups are used to consolidate groups of recipients into a single point of contact for email messages. Distribution groups can't be used to assign permissions to network resources in Active Directory. You can use the **Set-DistributionGroup** cmdlet to overwrite existing settings or to add new settings.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DistributionGroupIdParameter	The *Identity* parameter specifies the object that you want to modify. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)`

			Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
AcceptMessagesOnlyFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFrom* parameter specifies the mailbox users, mail users, and mail contacts who can send email messages to this distribution group. You can also specify Microsoft Exchange as a valid recipient for this parameter. If you configure a distribution

			group to accept messages only from the Microsoft Exchange recipient, the distribution group only receives system-generated messages. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address`

			Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com By default, the value of this parameter is blank, which enables the distribution group to accept messages from all senders.
AcceptMessagesOnlyFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromDLMembers* parameter specifies the distribution groups whose members are allowed to send email messages to this distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips

			• `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com By default, the value of this parameter is blank, which enables the distribution group to accept messages from all senders.
AcceptMessagesOnlyFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromSendersOrMembers* parameter specifies the recipients who can send email messages to this distribution group. You can specify users, contacts, or distribution groups. If you specify a

			distribution group, messages are accepted from all recipients that are members of that distribution group. You can also specify Exchange as a valid recipient for this parameter. If you configure a distribution group to accept messages only from the Exchange recipient, the distribution group only receives system-generated messages. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2

			• `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com By default, the value of this parameter is blank, which enables the distribution group to accept messages from all senders.
Alias	Optional	System.String	The *Alias* parameter specifies the alias of the distribution group. The *Alias* parameter is used to generate the primary SMTP email address of the object. The value of the *Alias* parameter can't contain spaces. If the *Alias* parameter isn't specified, the value of the *SamAccountName* parameter is used to

			generate the primary SMTP email address, with any spaces converted to underscores.
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArbitrationMailbox* parameter specifies the mailbox that's used to manage the moderation process. This parameter is automatically set when moderation is enabled.
BypassModerationFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassModerationFromSendersOrMembers* parameter specifies senders for whom moderation is to be bypassed. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example:

			CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
BypassNestedModerationEnabled	Optional	System.Boolean	The *ByPassNestedModerationEnabled* parameter specifies whether to allow the parent group moderators to provide approval for any nested groups that are also moderated. If you set this parameter to `$true`, all moderation in any nested distribution groups is

			bypassed on approved email messages. The default value is `$false`.
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* parameter specifies whether to bypass security checks and moderation for the member that's being added. Use this parameter when the specified distribution group is a moderated distribution group, and you want to bypass security checks and moderation for the member that's being added.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CreateDTMFMap	Optional	System.Boolean	The *CreateDTMFMap*

			parameter specifies that a dual-tone multiple-frequency (DTMF) map be created for the distribution group.
CustomAttribute1	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute10	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute11	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute12	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to

			store additional information.
CustomAttribute13	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute14	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute15	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute2	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.

CustomAttribute3	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute4	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute5	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute6	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute7	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify

			custom attributes. You can use these attributes to store additional information.
CustomAttribute8	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute9	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the distribution group's name in the Exchange Administration Center and in the Exchange global address list (GAL).
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain

			name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EmailAddresses	Optional	Microsoft.Exchange.Data.ProxyAddressCollection	The *EmailAddresses* parameter specifies the email alias of the distribution group. All valid Microsoft Exchange Server 2013 email address types can be used. You can specify multiple values for the *EmailAddresses* parameter as a comma-delimited list. **Important:** Exchange 2013 doesn't validate custom addresses for correct formatting. You must ensure that the custom address you specify complies with the format requirements for that address type. Because X.400 addresses are considered custom addresses in Exchange 2013, they also aren't validated, and you must provide the correct syntax when specifying an X.400 address.
EmailAddressPolicyEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013.

			The *EmailAddressPolicyEnabled* parameter specifies the application of email address policies to this distribution group. Valid values are `$true` or `$false`. The default value is `$true`. By default, all applicable email address policies are applied to this distribution group.
ExpansionServer	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *ExpansionServer* parameter specifies the Exchange server used to expand the distribution group. Enter the expansion server as a legacy Exchange DN value. The default behavior is to use the closest Exchange Server 2003 computer, or the closest Exchange Server 2007 or Exchange server that has the Hub Transport server role installed.
ExtensionCustomAttri	Optional	Microsoft.Exchange.Da	The

bute1		ta.MultiValuedProperty	*ExtensionCustomAttribute 1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute 1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values.

			For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that

			store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes.

			For more information about using multivalued properties, see Modifying multivalued properties.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* parameter suppresses the following confirmation: "To save changes on object <*object name*>, the object must be upgraded to the current Exchange version. After the upgrade, this object can't be managed by an earlier version of the Exchange Management Tools. Do you want to continue to upgrade and save the object?" This confirmation occurs when you upgrade a distribution group that was created in Microsoft Exchange Server 2003. You can't manage an Exchange 2003 distribution group by using the Exchange Administration Center until you update the object's version.
GenerateExternalDirectoryObjectId	Optional	System.Management.Automation.SwitchPar	This parameter is reserved for internal Microsoft use.

		ameter	
GrantSendOnBehalfTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *GrantSendOnBehalfTo* parameter specifies a mailbox user who can send on behalf of this distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/

			cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com By default, the value of the *GrantSendOnBehalfTo* parameter is blank, which means that no other mailbox user has permission to send on behalf of this distribution group.
HiddenFromAddressListsEnabled	Optional	System.Boolean	The *HiddenFromAddressListsEnabled* parameter specifies whether to hide the distribution group from any Exchange address list. Values are `$true` or `$false`. The default value is `$false`.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell

			session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
IgnoreNamingPolicy	Optional	System.Management.Automation.SwitchParameter	The *IgnoreNamingPolicy* parameter specifies that the naming policy defined in the *DistributionGroupNamingPolicy* parameter of the **Set-OrganizationConfig** cmdlet can be ignored. You don't need to specify a value with this

			parameter.
MailTip	Optional	System.String	The *MailTip* parameter specifies the message that's displayed to senders when they start drafting an email message to this recipient. The *MailTip* parameter message must be less than or equal to 250 characters.
MailTipTranslations	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *MailTipTranslations* parameter specifies additional languages when you want to provide the *MailTip* parameter information for this recipient in multiple languages. For each language, you must provide the locale, followed by a colon and the specific *MailTip* parameter message in that language. Each *MailTip* parameter message must be less than or equal to 250 characters. Multiple languages can be separated by commas.

ManagedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ManagedBy* parameter specifies the name of the mailbox recipient that appears on the **Managed by** tab of the Active Directory object. If this parameter isn't specified, the creator of the group is the owner. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/

			ou=AdministrativeGroup/ cn=Recipients/ cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com The recipients specified with the *ManagedBy* parameter aren't automatically members of the distribution group. If you want recipients specified in this parameter to be added as members of the distribution group, you need to add them as members.
MaxReceiveSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxReceiveSize* parameter specifies the maximum allowed email message size that can be sent to this distribution group. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes)

			• MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. Valid values are from 0 through 2147482624 bytes. By default, the *MaxReceiveSize* parameter is set to unlimited.
MaxSendSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxSendSize* parameter specifies the maximum allowed email message size that can be sent from this distribution group. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. Valid values are from 0 through 2147482624 bytes.

			By default, the *MaxSendSize* parameter is set to unlimited.
MemberDepartRestriction	Optional	Microsoft.Exchange.Data.Directory.Recipient.MemberUpdateType	The *MemberDepartRestriction* parameter specifies the restrictions that you can put on recipients who want to depart the group membership. This parameter takes the following values: • `Open` • `Closed` • `ApprovalRequired` **Note:** Universal security groups can't use the `Open` value. By default, they're set to `Closed`.
MemberJoinRestriction	Optional	Microsoft.Exchange.Data.Directory.Recipient.MemberUpdateType	The *MemberJoinRestriction* parameter specifies the restrictions that you can put on recipients who want to join the group membership. You can use the following values: • `Open` • `Closed` • `ApprovalRequired` **Note:** Universal security groups can't use the `Open` value. By default, they're set to `Closed`.

ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the list of users who are responsible for moderating the messages sent to this distribution group. To designate more than one user, separate the users by commas. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/

			ou=AdministrativeGroup/ cn=Recipients/ cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com This parameter is required if you set the *ModerationEnabled* parameter to `$true`. If you leave this parameter blank and there's a user already specified as the manager of this distribution group, the *ModeratedBy* field is automatically set to the *ManagedBy* parameter of the distribution group. Otherwise, an error is returned.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable moderation of email sent to this distribution group. To ensure moderation, set this parameter to `$true`. To disable moderation, set this parameter to `$false`. The default value is `$false`.

Name	Optional	System.String	The *Name* parameter specifies the name of the distribution group. The *Name* parameter specifies the distribution group name in Active Directory Users and Computers. **Note:** If group naming policy is enforced, you need to follow the naming constraints specified in the Set-OrganizationConfig cmdlet.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the primary return SMTP email address for the distribution group. This parameter only has meaning if the distribution group has multiple SMTP email addresses.
RejectMessagesFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFrom* parameter specifies mailbox users and mail-enabled contacts who aren't allowed to send email messages to this distribution group. This parameter accepts the following values:

			• Alias Example: JPhillips • Canonical DN Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • Display Name Example: Jeff Phillips • Distinguished Name (DN) Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • Domain\Account Example: Atlanta\JPhillips • GUID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • Immutable ID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • Legacy Exchange DN Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • SMTP Address Example: Jeff.Phillips@contoso.com • User Principal Name Example: JPhillips@contoso.com By default, the value of this parameter is blank, which enables the distribution group to

			accept messages from all mailbox users and all mail-enabled contacts.
RejectMessagesFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromDLMembers* parameter specifies the distribution groups who aren't allowed to send email messages to the distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN`

			Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • SMTP Address Example: Jeff.Phillips@contoso.com • User Principal Name Example: JPhillips@contoso.com By default, the value of this parameter is blank, which enables the distribution group to accept messages from all distribution groups.
RejectMessagesFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromSendersOrMembers* parameter specifies the identity of recipients from whom messages are rejected. This parameter accepts the following values: • Alias Example: JPhillips • Canonical DN Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • Display Name Example: Jeff Phillips • Distinguished Name (DN) Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=c

			ontoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
ReportToManagerEnabled	Optional	System.Boolean	The *ReportToManagerEnabled* parameter specifies whether to allow delivery reports to be sent to the distribution group manager. Valid values are `$true` or `$false`. The default value is `$false`.
ReportToOriginatorEnabled	Optional	System.Boolean	The *ReportToOriginatorEnabled* parameter specifies whether to allow delivery

			reports to be sent to the senders of email messages that are sent to this distribution group. Valid values are `$true` or `$false`. The default value is `$true`.
RequireSenderAuthenticationEnabled	Optional	System.Boolean	The *RequireSenderAuthenticationEnabled* parameter specifies whether to require email message delivery from authenticated senders. If the value is `$true`, messages are accepted only from authenticated senders. Valid values are `$true` or `$false`. The default value is `$true`.
RoomList	Optional	System.Management.Automation.SwitchParameter	The *RoomList* parameter specifies that all members of the distribution group are room resource mailboxes. You can create a distribution list for an office building in your organization and add all the rooms in that building to the distribution group. Room list distribution groups are used to

			generate a list of building locations in Microsoft Outlook so the user can select a building and get suggestions about when rooms are available in that building, without having to add the rooms in that building individually. You can't use the *RoomList* parameter with the *ExternalManaged* parameter.
SamAccountName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *SamAccountName* parameter specifies the name for clients of the object running older operating systems. The *SamAccountName* parameter is displayed in Active Directory.
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when they send a message to the

			moderated distribution group. You can specify one of the following values: • `Always` • `Internal` • `Never` If you want notifications to be sent to all external and internal senders, set this parameter to `Always`. If you want notifications to be sent only to the senders who are internal to your organization, set this parameter to `Internal`. To disable all status notifications, set this parameter to `Never`. **Note:** Senders are always notified if their message is rejected by the moderators, regardless of the value of the parameter. The default value is `Never`.
SendOofMessageToOriginatorEnabled	Optional	System.Boolean	The *SendOofMessageToOriginatorEnabled* parameter specifies whether to allow out-of-office messages to be delivered to the senders of email

			messages sent to this distribution group. Valid values are `$true` or `$false`. The default value is `$false`.
SimpleDisplayName	Optional	System.String	The *SimpleDisplayName* parameter is used to display an alternative description of the object when only a limited set of characters is permitted. This limited set of characters consists of ASCII characters 26 through 126, inclusively.
UMDtmfMap	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UMDtmfMap* parameter specifies whether you want to create a user-defined DTMF map for the UM-enabled distribution group.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.
WindowsEmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *WindowsEmailAddress* parameter specifies the value of the **E-mail** field of the Active Directory object.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Add-DistributionGroupMember

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Add-DistributionGroupMember** cmdlet to add a recipient to a distribution group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Add-DistributionGroupMember -Identity <DistributionGroupIdParameter> [-
BypassSecurityGroupManagerCheck <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Member
<RecipientWithAdUserGroupIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example adds John Evans to the distribution group Staff.

```
Add-DistributionGroupMember -Identity "Staff" -Member
"JohnEvans@contoso.com"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DistributionGroupIdParameter	The *Identity* parameter specifies the identity of the distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips

			• GUID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • Immutable ID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • Legacy Exchange DN Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • SMTP Address Example: Jeff.Phillips@contoso.com • User Principal Name Example: JPhillips@contoso.com
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* parameter specifies whether to bypass security checks and moderation for the member being added, if the specified distribution group is a moderated distribution group.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before

			processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Member	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientWithAdUserGroupIdParameter	The *Member* parameter specifies the recipient or Active Directory user to add to the distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,D

			C=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com To specify an Active Directory user, use one of the following values: • GUID • DN • UPN • *Domain\Account Name*
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what

			changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DistributionGroupMember

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-DistributionGroupMember** cmdlet to find existing distribution group members.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-DistributionGroupMember -Identity <DistributionGroupMemberIdParameter>
[-Credential <PSCredential>] [-DomainController <Fqdn>] [-
IgnoreDefaultScope <SwitchParameter>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>]
```

Examples

EXAMPLE 1

This example returns the existing distribution group members for the distribution group Marketing

USA.

```
Get-DistributionGroupMember -Identity "Marketing USA"
```

EXAMPLE 2

This example sets the scope of the search to the entire forest by running the **Set-ADServerSettings** cmdlet, and then the **Get-DistributionGroupMember** cmdlet searches the entire forest for the distribution group members in the Marketing Worldwide distribution group.

```
Set-ADServerSettings -ViewEntireForest $true
Get-DistributionGroupMember -Identity "Marketing Worldwide"
```

Detailed Description

If your organization has multiple Active Directory domains, you may need to run the **Set-ADServerSettings** cmdlet with the *ViewEntireForest* parameter set to `$true` before running the **Get-DistributionGroupMember** cmdlet to view the entire forest. For more information, see EXAMPLE 2.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DistributionGroupMemberIdParameter	The *Identity* parameter specifies the identity of the distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)`

			Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see

			Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController*

			parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *Credential* parameter.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest, and if you don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary

			to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your Exchange servers.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of members returned.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-DistributionGroupMember

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-DistributionGroupMember** cmdlet to remove an existing recipient from a distribution group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-DistributionGroupMember -Identity <DistributionGroupIdParameter> [-
BypassSecurityGroupManagerCheck <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Member
<GeneralRecipientIdParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes Jan Dryml from the distribution group Technical Support.

```
Remove-DistributionGroupMember -Identity "Technical
Support" -Member "Jan Dryml"
```

Detailed Description

The **Remove-DistributionGroupMember** cmdlet removes an existing recipient from a distribution group or a mail-enabled security group. You can't use the **Remove-DistributionGroupMember** cmdlet to remove a recipient from a dynamic distribution group. A dynamic distribution group queries Active Directory mail-enabled objects and builds the group's membership based on the results. The group's membership is recalculated whenever an email message is sent to the group.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DistributionGroupIdParameter	The *Identity* parameter specifies the distribution group that you want to modify. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name`

			(DN) Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* parameter specifies whether to bypass security checks and moderation for the member being added, if the specified distribution group is a moderated distribution group.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Member	Optional	Microsoft.Exchange.Configuration.Tasks.GeneralRecipientIdParameter	The *Member* parameter specifies the recipient that you want to remove from the distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips

			• `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You

			don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Update-DistributionGroupMember

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Update-DistributionGroupMember** cmdlet to update a member of a specified distribution group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Update-DistributionGroupMember -Identity <DistributionGroupIdParameter> [-
BypassSecurityGroupManagerCheck <SwitchParameter>] [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-Members
<MultiValuedProperty>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example updates John's membership in the distribution group Research Reports.

```
Update-DistributionGroupMember -Identity "Research Reports"
-Members john@contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DistributionGroupIdParameter	The *Identity* parameter specifies the identity of the distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.c

			om • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* parameter specifies whether to bypass security checks and moderation for the member being added, if the specified distribution group is a moderated distribution group.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-

			premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
Members	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Members* parameter specifies the recipient or Active Directory user to update as a member of the distribution group. To specify a recipient, you can use any of the following values: This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID`

			Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com To specify an Active Directory user, you can choose from this subset of the values: • `Distinquished Name (DN)` • `Domain\Account` • `GUID` • `User Principal Name (UPN)` Separate multiple users with a comma, for example, `"contoso\ayla","contoso\tony"`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf*

			switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-DynamicDistributionGroup

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-DynamicDistributionGroup** cmdlet to retrieve the settings on an existing dynamic distribution group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-DynamicDistributionGroup [-ManagedBy <GeneralRecipientIdParameter>]
<COMMON PARAMETERS>

Get-DynamicDistributionGroup [-Anr <String>] <COMMON PARAMETERS>

Get-DynamicDistributionGroup [-Identity <DynamicGroupIdParameter>] <COMMON
PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
```

```
Credential <PSCredential>] [-DomainController <Fqdn>] [-Filter <String>]
[-IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example retrieves the dynamic distribution group Marketing and pipes the object to the **Format-List** command to display the information about the dynamic distribution group.

```
Get-DynamicDistributionGroup -Identity "Marketing" |
Format-List
```

EXAMPLE 2

This example retrieves all dynamic distribution groups that contain the string "research" and pipes the object to the **Format-Table** command to display the dynamic distribution group names and who they're managed by.

```
Get-DynamicDistributionGroup -Anr *research* | Format-Table
Name, ManagedBy
```

Detailed Description

You can use distribution groups to create email distribution lists and security groups to assign permissions to shared resources. Distribution groups can be used only with email applications (such as Microsoft Exchange) to send email messages to collections of users. You can use the **Get-DynamicDistributionGroup** cmdlet to retrieve the settings on an existing dynamic distribution group in Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Dynamic distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParamet	This parameter is reserved for internal Microsoft use.

		er	
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Credential	Optional	System.Management.Automation.PSCredential	The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.DynamicGroupIdParameter	The *Identity* parameter specifies the dynamic distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account`

			Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default

			scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
ManagedBy	Optional	Microsoft.Exchange.Configuration.Tasks.GeneralRecipientIdParameter	The *ManagedBy* parameter specifies the name of the mailbox user, mail-enabled group, or mail-enabled contact that appears in the **Managed by** tab of the Active Directory object. This parameter accepts the following values:

			• Alias Example: JPhillips • Canonical DN Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • Display Name Example: Jeff Phillips • Distinguished Name (DN) Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • Domain\Account Example: Atlanta\JPhillips • GUID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • Immutable ID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • Legacy Exchange DN Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • SMTP Address Example: Jeff.Phillips@contoso.com • User Principal Name Example: JPhillips@contoso.com
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.

OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter limits results to a specific organizational unit (OU) container.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest and don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Exchange.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of recipient

			objects returned.
SortBy	Optional	System.String	The *SortBy* parameter is used to sort the results.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-DynamicDistributionGroup

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-DynamicDistributionGroup** cmdlet to create a dynamic distribution group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-DynamicDistributionGroup -RecipientFilter <String> <COMMON PARAMETERS>

New-DynamicDistributionGroup -IncludedRecipients <None | MailboxUsers |
Resources | MailContacts | MailGroups | MailUsers | AllRecipients> [-
ConditionalCompany <MultiValuedProperty>] [-ConditionalCustomAttribute1
<MultiValuedProperty>] [-ConditionalCustomAttribute10
<MultiValuedProperty>] [-ConditionalCustomAttribute11
<MultiValuedProperty>] [-ConditionalCustomAttribute12
<MultiValuedProperty>] [-ConditionalCustomAttribute13
<MultiValuedProperty>] [-ConditionalCustomAttribute14
<MultiValuedProperty>] [-ConditionalCustomAttribute15
<MultiValuedProperty>] [-ConditionalCustomAttribute2
<MultiValuedProperty>] [-ConditionalCustomAttribute3
<MultiValuedProperty>] [-ConditionalCustomAttribute4
<MultiValuedProperty>] [-ConditionalCustomAttribute5
<MultiValuedProperty>] [-ConditionalCustomAttribute6
<MultiValuedProperty>] [-ConditionalCustomAttribute7
<MultiValuedProperty>] [-ConditionalCustomAttribute8
<MultiValuedProperty>] [-ConditionalCustomAttribute9
<MultiValuedProperty>] [-ConditionalDepartment <MultiValuedProperty>] [-
ConditionalStateOrProvince <MultiValuedProperty>] <COMMON PARAMETERS>
```

```
COMMON PARAMETERS: -Name <String> [-Alias <String>] [-ArbitrationMailbox
<MailboxIdParameter>] [-Confirm [<SwitchParameter>]] [-DisplayName
<String>] [-DomainController <Fqdn>] [-ExternalDirectoryObjectId <String>]
[-ModeratedBy <MultiValuedProperty>] [-ModerationEnabled <$true | $false>]
[-Organization <OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-OverrideRecipientQuotas
<SwitchParameter>] [-PrimarySmtpAddress <SmtpAddress>] [-
RecipientContainer <OrganizationalUnitIdParameter>] [-
SendModerationNotifications <Never | Internal | Always>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates the dynamic distribution group Marketing Group in the Users container in the contoso.com domain. The Marketing Group dynamic distribution group contains all mailbox users or mail-enabled contacts found anywhere in the contoso.com domain who have a **Department** field that equals the strings "Marketing" or "Sales".

```
New-DynamicDistributionGroup -Name "Marketing Group" -Alias
"Marketing_Group" -IncludedRecipients
"MailboxUsers,MailContacts" -OrganizationalUnit
"contoso.com/Users" -ConditionalDepartment
"Marketing","Sales" -RecipientContainer "contoso.com"
```

EXAMPLE 2

This example uses the *RecipientFilter* parameter to create the dynamic distribution group Pacific Northwest in the Users container in the contoso.com domain. The Pacific Northwest dynamic distribution group contains all mailbox users found anywhere in the contoso.com domain who have a **State/Province** field that equals "Washington" or "Oregon".

```
New-DynamicDistributionGroup -Name "Pacific Northwest" -
Alias "Pacific_Northwest" -OrganizationalUnit "contoso.com/
Users" -RecipientFilter {((RecipientType -eq 'UserMailbox')
-and ((StateOrProvince -eq 'Washington' -or StateOrProvince
-eq 'Oregon')))} -RecipientContainer "contoso.com"
```

Detailed Description

A dynamic distribution group queries Active Directory mail-enabled objects and builds the group membership based on the results. The group membership is recalculated whenever an email message is sent to the group. The query filters provided with Microsoft Exchange are limited to any combination of the following parameters:

- *ConditionalCompany*

- *ConditionalCustomAttribute N* (where *N* is a value from 1 through 15)
- *ConditionalDepartment*
- *ConditionalStateOrProvince*
- *IncludedRecipients*

You can also create any custom query using the *RecipientFilter* parameter.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Dynamic distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
IncludedRecipients	Required	Microsoft.Exchange.Data.Directory.Recipient.WellKnownRecipientType	The *IncludedRecipients* parameter filters the recipient types used to build the dynamic distribution group. The *IncludedRecipients* parameter can't be used if the *RecipientFilter* parameter is specified. You can use the following values: • `AllRecipients` • `MailboxUsers` • `Resources` • `MailContacts` • `MailGroups` • `MailUsers` • `None` `AllRecipients` can be used only by itself. When multiple values in the *IncludedRecipients* parameter are separated by commas, the **OR** Boolean operator is applied.

Name	Required	System.String	The *Name* parameter specifies the name of the dynamic distribution group. If the *DisplayName* parameter isn't specified, the value of the *Name* parameter is also used for the *DisplayName* parameter. The *Name* parameter value can't exceed 64 characters. If the values of the *Name* and *DisplayName* parameters are different: • The *Name* parameter specifies the dynamic distribution group name in Active Directory Users and Computers. • The *DisplayName* parameter specifies the dynamic distribution group name in the Exchange Administration Center and in the Exchange global address list (GAL).
RecipientFilter	Required	System.String	The *RecipientFilter* parameter filters the mail-enabled recipients used to

			build the dynamic distribution group. The *RecipientFilter* parameter can't be used if any of the following parameters are specified: • *IncludedRecipients* • *ConditionalCompany* • *ConditionalCustomAttribute N* (where *N* is a value from 1 through 15) • *ConditionalDepartment* • *ConditionalStateOrProvince* The *RecipientFilter* parameter uses `OPath` syntax to query Active Directory and filter recipients.
Alias	Optional	System.String	The *Alias* parameter specifies the alias of the dynamic distribution group. The value of the *Alias* parameter is used to generate the primary SMTP email address for the dynamic distribution group. The value of the *Alias* parameter can't contain spaces. If the *Alias* parameter isn't specified,

			the value of the *Name* parameter is used for the value of the *Alias* parameter, with any spaces in the *Name* parameter converted to underscore characters (_).
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArbitrationMailbox* parameter specifies the mailbox used to manage the moderation process.
ConditionalCompany	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCompany* parameter filters the mailbox users or mail-enabled contacts used to build the dynamic distribution group. When multiple values of the *ConditionalCompany* parameter are separated by commas, the **OR** Boolean operator is applied.
ConditionalCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify

			filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute10	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter.

			You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute11	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute12	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribu*

			te1 value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute13	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAtt	Optional	Microsoft.Exchange.Da	The

ribute14		ta.MultiValuedProperty	*ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute15	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in

			this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify

			filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter.

			You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute6	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribu*

			te1 value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute7	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAtt	Optional	Microsoft.Exchange.Da	The

ribute8		ta.MultiValuedProperty	*ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute9	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in

			this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalDepartment	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalDepartment* parameter filters the mailbox users or mail-enabled contacts used to build the dynamic distribution group. The *ConditionalDepartment* parameter can't be used if the *RecipientFilter* parameter is specified. When multiple values of the *ConditionalDepartment* parameter are separated by commas, the **OR** Boolean operator is applied.
ConditionalStateOrProvince	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalStateOrProvince* parameter filters mailbox users or mail-enabled contacts used to build the dynamic distribution group. The

			ConditionalStateOrProvince parameter can't be used if the *RecipientFilter* parameter is specified. When multiple values of the *ConditionalStateOrProvince* parameter are separated by commas, the **OR** Boolean operator is applied.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the name of the dynamic distribution group in the Exchange Administration Center and in the Exchange GAL. If the *DisplayName* parameter isn't specified, the value of the *Name* parameter is used for the *DisplayName*

			parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExternalDirectoryObjectId	Optional	System.String	This parameter is reserved for internal Microsoft use.
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users who are responsible for moderating the messages sent to this dynamic distribution group. To designate more than one user, separate the users with commas. This parameter is required if you set the *ModerationEnabled* parameter to `$true`.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable moderation for the

			dynamic distribution group. To enable moderation, set this parameter to `$true`. To disable moderation, set this parameter to `$false`. The default value is `$false`.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies where to create the dynamic distribution group in Active Directory by using canonical name syntax.
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the primary return SMTP email address for the dynamic distribution group. This parameter has meaning only if the dynamic distribution group has multiple SMTP email addresses.

RecipientContainer	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientContainer* parameter filters the recipients used to build the dynamic distribution group based on their location in Active Directory. The value of the *RecipientContainer* parameter can be the canonical name of an organizational unit (OU) or a domain. If you don't specify a value for the *RecipientContainer* parameter, the cmdlet will default to use the local container. This location is specified by using the *OrganizationalUnit* parameter.
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when they send a message to the moderated dynamic distribution group. You can specify one of the following values: • `Always` • `Internal` • `Never`

			If you want notifications to be sent to all senders, set this parameter to `Always`. If you want notifications to be sent only to the senders internal to your organization, set this parameter to `Internal`. To disable all status notifications, set this parameter to `Never`. **Note:** Senders are always notified if their message is rejected by the moderators, regardless of the value of this parameter. The default value is `Never`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-DynamicDistributionGroup

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-DynamicDistributionGroup** cmdlet to delete an existing dynamic distribution group. This cmdlet removes the dynamic distribution group from Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-DynamicDistributionGroup -Identity <DynamicGroupIdParameter> [-
Confirm [<SwitchParameter>]] [-DomainController <Fqdn>] [-
IgnoreDefaultScope <SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example deletes the dynamic distribution group Test Users.

```
Remove-DynamicDistributionGroup -Identity "Test Users"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Dynamic distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DynamicGroupIdParameter	The *Identity* parameter specifies the dynamic distribution group that you want to remove. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips

			• `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the

			default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-DynamicDistributionGroup

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-12*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-DynamicDistributionGroup** cmdlet to modify the settings of an existing dynamic distribution group.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-DynamicDistributionGroup -Identity <DynamicGroupIdParameter> [-
AcceptMessagesOnlyFrom <MultiValuedProperty>] [-
AcceptMessagesOnlyFromDLMembers <MultiValuedProperty>] [-
AcceptMessagesOnlyFromSendersOrMembers <MultiValuedProperty>] [-Alias
<String>] [-ArbitrationMailbox <MailboxIdParameter>] [-
BypassModerationFromSendersOrMembers <MultiValuedProperty>] [-
ConditionalCompany <MultiValuedProperty>] [-ConditionalCustomAttribute1
<MultiValuedProperty>] [-ConditionalCustomAttribute10
<MultiValuedProperty>] [-ConditionalCustomAttribute11
<MultiValuedProperty>] [-ConditionalCustomAttribute12
<MultiValuedProperty>] [-ConditionalCustomAttribute13
<MultiValuedProperty>] [-ConditionalCustomAttribute14
<MultiValuedProperty>] [-ConditionalCustomAttribute15
<MultiValuedProperty>] [-ConditionalCustomAttribute2
<MultiValuedProperty>] [-ConditionalCustomAttribute3
<MultiValuedProperty>] [-ConditionalCustomAttribute4
```

```
<MultiValuedProperty>] [-ConditionalCustomAttribute5
<MultiValuedProperty>] [-ConditionalCustomAttribute6
<MultiValuedProperty>] [-ConditionalCustomAttribute7
<MultiValuedProperty>] [-ConditionalCustomAttribute8
<MultiValuedProperty>] [-ConditionalCustomAttribute9
<MultiValuedProperty>] [-ConditionalDepartment <MultiValuedProperty>] [-
ConditionalStateOrProvince <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-CreateDTMFMap <$true | $false>] [-CustomAttribute1
<String>] [-CustomAttribute10 <String>] [-CustomAttribute11 <String>] [-
CustomAttribute12 <String>] [-CustomAttribute13 <String>] [-
CustomAttribute14 <String>] [-CustomAttribute15 <String>] [-
CustomAttribute2 <String>] [-CustomAttribute3 <String>] [-CustomAttribute4
<String>] [-CustomAttribute5 <String>] [-CustomAttribute6 <String>] [-
CustomAttribute7 <String>] [-CustomAttribute8 <String>] [-CustomAttribute9
<String>] [-DisplayName <String>] [-DomainController <Fqdn>] [-
EmailAddresses <ProxyAddressCollection>] [-EmailAddressPolicyEnabled
<$true | $false>] [-ExpansionServer <String>] [-ExtensionCustomAttribute1
<MultiValuedProperty>] [-ExtensionCustomAttribute2 <MultiValuedProperty>]
[-ExtensionCustomAttribute3 <MultiValuedProperty>] [-
ExtensionCustomAttribute4 <MultiValuedProperty>] [-
ExtensionCustomAttribute5 <MultiValuedProperty>] [-ForceUpgrade
<SwitchParameter>] [-GrantSendOnBehalfTo <MultiValuedProperty>] [-
HiddenFromAddressListsEnabled <$true | $false>] [-IgnoreDefaultScope
<SwitchParameter>] [-IncludedRecipients <None | MailboxUsers | Resources |
MailContacts | MailGroups | MailUsers | AllRecipients>] [-MailTip
<String>] [-MailTipTranslations <MultiValuedProperty>] [-ManagedBy
<GeneralRecipientIdParameter>] [-MaxReceiveSize <Unlimited>] [-MaxSendSize
<Unlimited>] [-ModeratedBy <MultiValuedProperty>] [-ModerationEnabled
<$true | $false>] [-Name <String>] [-Notes <String>] [-PhoneticDisplayName
<String>] [-PrimarySmtpAddress <SmtpAddress>] [-RecipientContainer
<OrganizationalUnitIdParameter>] [-RecipientFilter <String>] [-
RejectMessagesFrom <MultiValuedProperty>] [-RejectMessagesFromDLMembers
<MultiValuedProperty>] [-RejectMessagesFromSendersOrMembers
<MultiValuedProperty>] [-ReportToManagerEnabled <$true | $false>] [-
ReportToOriginatorEnabled <$true | $false>] [-
RequireSenderAuthenticationEnabled <$true | $false>] [-
SendModerationNotifications <Never | Internal | Always>] [-
SendOofMessageToOriginatorEnabled <$true | $false>] [-SimpleDisplayName
<String>] [-UMDtmfMap <MultiValuedProperty>] [-WhatIf [<SwitchParameter>]]
[-WindowsEmailAddress <SmtpAddress>]
```

Examples

EXAMPLE 1

This example applies the following changes to the existing dynamic distribution group Developers:

- Change the `ConditionalCompany` query filter to `Contoso`.
- Change the `IncludedRecipients` query filter to `MailboxUsers`.
- Add the value `Internal` to the `ConditionalCustomAttribute1` attribute.

```
Set-DynamicDistributionGroup -Identity "Developers" -
ConditionalCompany "Contoso" -IncludedRecipients
MailboxUsers -ConditionalCustomAttribute1 "Internal"
```

Detailed Description

A dynamic distribution group queries Active Directory mail-enabled objects and builds the group membership based on the results. The group membership is recalculated whenever an email message is sent to a group. You can use the **Set-DynamicDistributionGroup** cmdlet to overwrite existing settings or to add new settings.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Dynamic distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.DynamicGroupIdParameter	The *Identity* parameter specifies the object that you want to modify. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/

			ou=AdministrativeGroup/ cn=Recipients/ cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
AcceptMessagesOnlyFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFrom* parameter specifies the mailbox users, mail users, and mail contacts that can send email messages to this dynamic distribution group. You can also specify Microsoft Exchange as a valid recipient for this parameter. If you configure a dynamic distribution group to accept messages only from the Microsoft Exchange recipient, the dynamic distribution group only receives system-generated messages. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN`

			Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com By default, this parameter is blank, which enables the dynamic distribution group to accept messages from all senders.

AcceptMessagesOnlyFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromDLMembers* parameter specifies the distribution groups whose members are allowed to send email messages to this dynamic distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/

			cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com By default, this parameter is blank, which enables the dynamic distribution group to accept messages from all senders.
AcceptMessagesOnlyFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromSendersOrMembers* parameter specifies the recipients who can send email messages to this dynamic distribution group. You can specify users, contacts, or distribution groups. If you specify a distribution group, messages are accepted from all recipients that are members of that distribution group. You can also specify Microsoft Exchange as a valid recipient for this parameter. If you configure a distribution group to accept messages

			only from the Microsoft Exchange recipient, the dynamic distribution group only receives system-generated messages. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address`

			Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com By default, the value of this parameter is blank, which enables the distribution group to accept messages from all senders. This parameter can't be used in conjunction with the *AcceptMessagesOnlyFrom* or *AcceptMessagesOnlyFromDLMembers* parameters.
Alias	Optional	System.String	The *Alias* parameter specifies the alias of the object. The value of the *Alias* parameter can't contain spaces.
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArbitrationMailbox* parameter specifies the mailbox used to manage the moderation process. This parameter accepts the following values: • `Alias`

			Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
BypassModerationFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassModerationFromSendersOrMembers* parameter specifies the list of senders for whom

			moderation is bypassed. You can specify users or distribution groups. If you specify a distribution group, the moderation is bypassed for all recipients that are members of that distribution group. Separate multiple values with a comma. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN`

			Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
ConditionalCompany	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCompany* parameter filters the mailbox users or mail-enabled contacts used to build the dynamic distribution group. When multiple values of the *ConditionalCompany* parameter are separated by commas, the **OR** Boolean operator is applied. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients'

			custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute10	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this

			parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute11	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute12	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all

			included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute13	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute14	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *ConditionalCustomAttribu*

		y	*te1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute15	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use

			the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients'

			custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this

			parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute6	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all

			included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute7	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute8	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *ConditionalCustomAttribu*

		y	*te1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalCustomAttribute9	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalCustomAttribute1* to *ConditionalCustomAttribute15* parameters specify filters for recipients' custom attributes. For example, if you set the *ConditionalCustomAttribute1* value to `Marketing`, all included recipients whose *CustomAttribute1* value is `Marketing` are included in this filter. You must use

			the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalDepartment	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalDepartment* parameter filters the mailbox users or mail-enabled contacts used to build the dynamic distribution group. When multiple values of the *ConditionalDepartment* parameter are separated by commas, the **OR** Boolean operator is applied. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
ConditionalStateOrProvince	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ConditionalStateOrProvince* parameter filters the mailbox users or mail-enabled contacts used to build the dynamic

			distribution group. When multiple values of the *ConditionalStateOrProvince* parameter are separated by commas, the **OR** Boolean operator is applied. You must use the *IncludedRecipients* parameter if you use a *Conditional* parameter. You can't use this parameter if you use the *RecipientFilter* parameter.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CreateDTMFMap	Optional	System.Boolean	The *CreateDTMFMap* parameter specifies that a dual-tone multiple-frequency (DTMF) map be created for the dynamic distribution group.
CustomAttribute1	Optional	System.String	The *CustomAttribute1* to

			CustomAttribute15 parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute10	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute11	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute12	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute13	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can

			use these attributes to store additional information.
CustomAttribute14	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute15	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute2	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute3	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.

CustomAttribute4	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute5	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute6	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute7	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute8	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify

			custom attributes. You can use these attributes to store additional information.
CustomAttribute9	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the name of the distribution group in the Exchange Administration Center and in the Exchange global address list (GAL).
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EmailAddresses	Optional	Microsoft.Exchange.Data.ProxyAddressCollec	The *EmailAddresses* parameter specifies the

		tion	email addresses of the distribution group. All Microsoft Exchange email address types are valid. You can specify multiple values for the *EmailAddresses* parameter. Separate multiple values by using commas. **Important:** Exchange doesn't validate custom addresses for proper formatting. You must ensure that the custom address you specify complies with the format requirements for that address type. Because X.400 addresses are considered custom addresses in Exchange, they also aren't validated and you must provide the correct syntax when specifying an X.400 address.
EmailAddressPolicyEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *EmailAddressPolicyEnabled* parameter specifies the application of email address policies to this dynamic distribution group. Valid values are

			`$true` or `$false`. The default value is `$true`. By default, all applicable email address policies are applied to this distribution group.
ExpansionServer	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *ExpansionServer* parameter specifies the Exchange server used to expand the distribution group. Enter the expansion server as a `Legacy Exchange DN` value. The default behavior is to use the closest Exchange server that has the Mailbox server role installed.
ExtensionCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute*

			parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *ExtensionCustomAttribute*

		y	*1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute 1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information

			about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* parameter specifies whether to suppress the following confirmation: "To save changes on

			object, the object must be upgraded to the current Exchange version. After upgrade, this object cannot be managed by a previous version of Exchange System Manager. Do you want to continue to upgrade and save the object?" This confirmation occurs when you upgrade a dynamic distribution group that was created in Microsoft Exchange Server 2003. You can't manage an Exchange 2003 dynamic distribution group by using the Exchange Administration Center until you update the object's version and change the recipient filter by using either the *RecipientFilter* or *IncludedRecipients* parameters.
GrantSendOnBehalfTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *GrantSendOnBehalfTo* parameter specifies a mailbox user who can send on behalf of this dynamic distribution

			group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com

			By default, the *GrantSendOnBehalfTo* parameter is blank, which means that no other mailbox user has permission to send on behalf of this distribution group.
HiddenFromAddressListsEnabled	Optional	System.Boolean	The *HiddenFromAddressListsEnabled* parameter specifies whether to hide the distribution group from any Exchange address lists. Valid values are `$true` or `$false`. The default value is `$false`.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default

			scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
IncludedRecipients	Optional	Microsoft.Exchange.Data.Directory.Recipient.WellKnownRecipientType	The *IncludedRecipients* parameter filters the recipient types used to build the dynamic distribution group. The available values for the *IncludedRecipients* parameter are: • `None` • `AllRecipients` • `MailboxUsers` • `Resources` • `MailContacts` • `MailUsers` • `MailGroups` The `AllRecipients` value can be used only by itself. When multiple values of

			the *IncludedRecipients* parameter are separated by commas, the **OR** Boolean operator is applied.
MailTip	Optional	System.String	The *MailTip* parameter specifies the message displayed to senders when they start drafting an email message to this dynamic distribution group. The *MailTip* parameter message must be less than or equal to 250 characters.
MailTipTranslations	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *MailTipTranslations* parameter specifies additional languages when you want to provide the *MailTip* parameter information for this recipient in multiple languages. For each language, you must provide the locale, followed by a colon and the specific *MailTip* parameter message in that language. Each *MailTip* parameter message must be less than or equal to 250

			characters. Multiple languages can be separated by commas.
ManagedBy	Optional	Microsoft.Exchange.Configuration.Tasks.GeneralRecipientIdParameter	The *ManagedBy* parameter specifies the name of the mailbox user, mail-enabled group, or mail-enabled contact that appears on the **Managed by** tab of the Active Directory object. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.c

			om • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com **Note:** Don't set this parameter to be managed by a recipient that's not mail-enabled because, if you try to edit this dynamic distribution group at a later time, you'll receive an error message stating that the object couldn't be found.
MaxReceiveSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxReceiveSize* parameter specifies the maximum allowed email message size that can be sent to this distribution group. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes)

			• `TB` (terabytes) Unqualified values are treated as bytes. Valid values are from 0 through 2147482624 bytes. By default, the *MaxReceiveSize* parameter is set to `unlimited`.
MaxSendSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxSendSize* parameter specifies the maximum allowed email message size that can be sent from this distribution group. When you enter a value, qualify the value with one of the following units: • `B` (bytes) • `KB` (kilobytes) • `MB` (megabytes) • `GB` (gigabytes) • `TB` (terabytes) Unqualified values are treated as bytes. Valid values are from 0 through 2147482624 bytes. By default, the *MaxSendSize* parameter is set to `unlimited`.

ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users who are responsible for moderating the messages sent to this dynamic distribution group. To designate more than one user, separate the users with commas. This parameter is required if you set the *ModerationEnabled* parameter to `$true`. If you leave this parameter blank and there's a user who's already specified as the manager of this mailbox, the *ModeratedBy* parameter is automatically set by the *ManagedBy* parameter of the mailbox. Otherwise, an error is returned.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable moderation for the dynamic distribution group. To enable moderation, set this parameter to `$true`. To disable moderation, set

			this parameter to `$false`. The default value is `$false`.
Name	Optional	System.String	The *Name* parameter specifies the name of the dynamic distribution group. The *Name* parameter specifies the distribution group name in Active Directory Users and Computers.
Notes	Optional	System.String	The *Notes* parameter specifies comments about the distribution group.
PhoneticDisplayName	Optional	System.String	The *PhoneticDisplayName* parameter specifies a phonetic pronunciation of the *DisplayName* parameter. The maximum length of this parameter value is 255 characters.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the primary return SMTP email address for the distribution group. This parameter has meaning only if the distribution group has multiple SMTP

			email addresses.
RecipientContainer	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *RecipientContainer* parameter filters the recipients used to build the dynamic distribution group based on their location in Active Directory. The value of the *RecipientContainer* parameter can be the canonical name of an organizational unit (OU) or a domain. If a value for the *RecipientContainer* parameter isn't specified, the cmdlet will default to use the local container.
RecipientFilter	Optional	System.String	The *RecipientFilter* parameter filters the mail-enabled recipients used to build the dynamic distribution group. The *RecipientFilter* parameter can't be used if any of the following parameters are specified: • *IncludedRecipients* • *ConditionalCompany* • *ConditionalDepartment* • *ConditionalStateOrProvince* The *RecipientFilter*

			parameter uses `OPath` syntax to query Active Directory and filter recipients. The *RecipientFilter* parameter can use any combination of the following object types as filters: • `RecipientType` • `ConditionalCompany` • `ConditionalDepartment` • `ConditionalStateOrProvince`
RejectMessagesFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFrom* parameter specifies mailbox users and mail-enabled contacts that aren't allowed to send email messages to this distribution group. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account`

			Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com By default, this parameter is blank, which enables the distribution group to accept messages from all mailbox users and all mail-enabled contacts.
RejectMessagesFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromDLMembers* parameter specifies the distribution groups who aren't allowed to send email messages to the distribution group. This parameter accepts the following values:

			• Alias Example: JPhillips • Canonical DN Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • Display Name Example: Jeff Phillips • Distinguished Name (DN) Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • Domain\Account Example: Atlanta\JPhillips • GUID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • Immutable ID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • Legacy Exchange DN Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • SMTP Address Example: Jeff.Phillips@contoso.com • User Principal Name Example: JPhillips@contoso.com By default, this parameter is blank, which enables the distribution group to accept messages from all

			distribution groups.
RejectMessagesFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromSendersOrMembers* parameter specifies the recipients who aren't allowed to send email messages to this distribution group. You can specify users, contacts, or distribution groups. If you specify a distribution group, messages from any recipient that's a member of that distribution group is rejected. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-

			4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com By default, the value of this parameter is blank, which enables the distribution group to accept messages from all senders.
ReportToManagerEnabled	Optional	System.Boolean	The *ReportToManagerEnabled* parameter specifies whether to enable delivery reports to be sent to the distribution group manager. Valid values are `$true` or `$false`. The default value is `$false`.
ReportToOriginatorEnabled	Optional	System.Boolean	The *ReportToOriginatorEnable*

			d parameter specifies whether to enable delivery reports to be sent to the sender of email messages that are sent to this distribution group. Valid values are `$true` or `$false`. The default value is `$true`.
RequireSenderAuthenticationEnabled	Optional	System.Boolean	The *RequireSenderAuthenticationEnabled* parameter specifies whether to require email message delivery from authenticated senders. Valid values are `$true` or `$false`. The default value is `$true`. If the value is `$true`, messages are accepted only from authenticated senders.
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when a message they sent to the moderated distribution group is rejected by one of the moderators. You

			can specify one of the following values: • `Always` • `Internal` • `Never` If you want notifications to be sent to all senders, set this parameter to `Always`. If you want notifications to be sent to only the senders that are internal to your organization, set this parameter to `Internal`. To disable all status notifications, set this parameter to `Never`. The default value is `Never`.
SendOofMessageToOriginatorEnabled	Optional	System.Boolean	The *SendOofMessageToOriginatorEnabled* parameter specifies whether to enable out-of-office messages to be delivered to the sender of email messages that are sent to this distribution group. Valid values are `$true` or `$false`. The default value is `$false`.
SimpleDisplayName	Optional	System.String	The *SimpleDisplayName*

			parameter is used to display an alternative description of the object when only a limited set of characters is permitted. This limited set of characters consists of ASCII characters 26 through 126, inclusively.
UMDtmfMap	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UMDtmfMap* parameter specifies whether you want to create a user-defined DTMF map for the Unified Messaging (UM)-enabled dynamic distribution group.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsEmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *WindowsEmailAddress*

			parameter specifies the value of the **E-mail** field of the Active Directory object.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-Group

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-Group** cmdlet to query for existing groups.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-Group [-Anr <String>] <COMMON PARAMETERS>

Get-Group [-Identity <GroupIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-Filter <String>]
[-IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-ReadFromDomainController
<SwitchParameter>] [-RecipientTypeDetails <RecipientTypeDetails[]>] [-
ResultSize <Unlimited>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example uses the **Get-Group** cmdlet without parameters to return all of the groups in Active Directory.

```
Get-Group
```

EXAMPLE 2

This example uses the *Identity* parameter to return the group Marketing Reports.

```
Get-Group -Identity "Marketing Reports"
```

EXAMPLE 3

This example uses the *Anr* parameter to return all groups that begin with "Mar".

```
Get-Group -Anr Mar*
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an

			attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the account used to read Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.

			The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.GroupIdParameter	The *Identity* parameter specifies the identity of the distribution group object. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)`

			Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This

			allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.Orga	The *OrganizationalUnit* parameter specifies an

		nizationalUnitIdParameter	organizational unit (OU) and is used to limit the results.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest, and if you don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Microsoft Exchange.
RecipientTypeDetails	Optional	Microsoft.Exchange.Data.Directory.Recipient.RecipientTypeDetails[]	The *RecipientTypeDetails* parameter specifies the type of recipients

			returned. Recipient types are divided into recipient types and subtypes. Each recipient type contains all common properties for all subtypes. For example, the type UserMailbox represents a user account in Active Directory that has an associated mailbox. Because there are several mailbox types, each mailbox type is identified by the *RecipientTypeDetails* parameter. For example, a conference room mailbox has *RecipientTypeDetails* set to ConferenceRoomMailbox, whereas a user mailbox has *RecipientTypeDetails* set to UserMailbox. You can select from the following values: • `ArbitrationMailbox` • `ConferenceRoomMailbox` • `Contact` • `DiscoveryMailbox` • `DynamicDistributionGroup` • `EquipmentMailbox` • `ExternalManagedContact` • `ExternalManagedDistributionGroup` • `LegacyMailbox` • `LinkedMailbox` • `MailboxPlan` • `MailContact`

			• MailForestContact • MailNonUniversalGroup • MailUniversalDistributionGroup • MailUniversalSecurityGroup • MailUser • PublicFolder • RoleGroup • RoomList • RoomMailbox • SharedMailbox • SystemAttendantMailbox • SystemMailbox • User • UserMailbox
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of members returned.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute to sort by.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-Group

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-04-11*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-Group** cmdlet to modify group object settings visible in Active Directory Users and Computers. If the group is a mail-enabled security group or a distribution group, you can use the **Set-DistributionGroup** cmdlet to modify other Microsoft Exchange settings that aren't available by using the **Set-Group** cmdlet.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-Group -Identity <GroupIdParameter> <COMMON PARAMETERS>

Set-Group -Identity <GroupIdParameter> [-Universal <SwitchParameter>]
<COMMON PARAMETERS>

COMMON PARAMETERS: [-BypassSecurityGroupManagerCheck <SwitchParameter>] [-
Confirm [<SwitchParameter>]] [-DisplayName <String>] [-DomainController
<Fqdn>] [-IgnoreDefaultScope <SwitchParameter>] [-IsHierarchicalGroup
<$true | $false>] [-ManagedBy <GeneralRecipientIdParameter[]>] [-Name
<String>] [-Notes <String>] [-PhoneticDisplayName <String>] [-
SeniorityIndex <Int32>] [-SimpleDisplayName <String>] [-WhatIf
[<SwitchParameter>]] [-WindowsEmailAddress <SmtpAddress>]
```

Examples

EXAMPLE 1

This example applies the following changes to the existing global security group Legal Department:

- Change the group's scope to universal.
- Add a *Notes* parameter value of verified.

```
Set-Group -Identity "Legal Department" -Universal -Notes
"verified"
```

EXAMPLE 2

This example specifies that the group Human Resources is a hierarchical group and will display last within its hierarchy because its index number is 1.

```
Set-Group -Identity "Human Resources" -IsHierarchicalGroup
$true -SeniorityIndex 1
```

Detailed Description

You can't use the **Set-Group** cmdlet to modify dynamic distribution groups. To modify dynamic distribution groups, use the **Set-DynamicDistributionGroup** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for

this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Distribution groups" entry in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.GroupIdParameter	The *Identity* parameter specifies the object that you want to modify. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/

			cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
BypassSecurityGroupManagerCheck	Optional	System.Management.Automation.SwitchParameter	The *BypassSecurityGroupManagerCheck* parameter specifies whether to bypass security checks and moderation if the specified group is a moderated distribution group.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the name of the group in the Exchange Administration Center and in the Exchange global address

			list (GAL). This parameter has significance only if the group is mail-enabled.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory. The *DomainController* parameter isn't supported on Edge Transport servers. An Edge Transport server uses the local instance of Active Directory Lightweight Directory Services (AD LDS) to read and write data.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope

			setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as GUID, aren't accepted.
IsHierarchicalGroup	Optional	System.Boolean	The *IsHierarchicalGroup* parameter specifies whether the group is part of a hierarchical address book. This parameter accepts `$true` or `$false`. The default value is `$false`.

ManagedBy	Optional	Microsoft.Exchange.Configuration.Tasks.GeneralRecipientIdParameter[]	The *ManagedBy* parameter specifies the name of the user, group, or contact that appears in the **Managed by** tab of the Active Directory object. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips

			• `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Name	Optional	System.String	The *Name* parameter specifies the name of the group. The *Name* parameter specifies the group name in Active Directory Users and Computers.
Notes	Optional	System.String	The *Notes* parameter specifies the notes that describe the purpose of the group.
PhoneticDisplayName	Optional	System.String	The *PhoneticDisplayName* parameter specifies a phonetic pronunciation of the *DisplayName* parameter. The maximum length of this parameter value is 255 characters.
SeniorityIndex	Optional	System.Int32	The *SeniorityIndex* parameter specifies the order in which this group will display in a hierarchical address book. A group with a value of 2 will display higher in an

			address book than a group with a value of 1.
SimpleDisplayName	Optional	System.String	The *SimpleDisplayName* parameter is used to display an alternative description of the object when only a limited set of characters is permitted. This limited set of characters consists of ASCII characters 26 through 126, inclusively. The *SimpleDisplayName* parameter has meaning only for mail-enabled objects.
Universal	Optional	System.Management.Automation.SwitchParameter	The *Universal* parameter changes the scope of the group from `global` or `domain local` to `universal`.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.
WindowsEmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *WindowsEmailAddress* parameter specifies the value of the **E-mail** field of the Active Directory object.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-MailContact

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-MailContact** cmdlet to remove the Exchange attributes of an existing mail-enabled contact in Active Directory. The contact object remains in Active Directory but is no longer mail-enabled.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-MailContact -Identity <MailContactIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreDefaultScope
<SwitchParameter>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example mail-disables a contact.

```
Disable-MailContact -Identity EdMeadows
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailContactIdParameter	The *Identity* parameter specifies the mail contact. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID`

			Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax `-Confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.	The *IgnoreDefaultScope* parameter instructs the

		Automation.SwitchParameter	command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what

			changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-MailContact

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-03-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-MailContact** cmdlet to mail-enable an existing contact in Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-MailContact -Identity <ContactIdParameter> -ExternalEmailAddress
<ProxyAddress> [-Alias <String>] [-Confirm [<SwitchParameter>]] [-
DisplayName <String>] [-DomainController <Fqdn>] [-MacAttachmentFormat
<BinHex | UuEncode | AppleSingle | AppleDouble>] [-MessageBodyFormat <Text
| Html | TextAndHtml>] [-MessageFormat <Text | Mime>] [-
OverridePrimarySmtpAddress <SmtpAddress>] [-UsePreferMessageFormat <$true | $false>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example mail-enables an existing contact in Active Directory and provides the contact with the external email address roland@tailspintoys.com.

```
Enable-MailContact -Identity Roland -ExternalEmailAddress
"roland@tailspintoys.com"
```

Detailed Description

The **Enable-MailContact** cmdlet mail-enables an existing contact in Active Directory by adding the attributes required by Microsoft Exchange. The contact's identity, associated alias, and target email address are required to mail-enable a contact.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
ExternalEmailAddress	Required	Microsoft.Exchange.Data.ProxyAddress	The *ExternalEmailAddress* parameter specifies the target email address.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.ContactIdParameter	The *Identity* parameter specifies the identity of the contact. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example:

			CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Alias	Optional	System.String	The *Alias* parameter specifies the alias of the mail-enabled contact.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the contact's display name.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
MacAttachmentFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MacAttachmentFormat	The *MacAttachmentFormat* parameter specifies the Apple Macintosh operating system attachment format for messages sent to the mail contact. The valid values for this parameter are: • `BinHex` • `UuEncode` • `AppleSingle` • `AppleDouble` By default, this parameter is set to `BinHex`. The acceptable values for the *MacAttachmentFormat* parameter are dependent on the *MessageFormat* parameter. If the

			MessageFormat parameter is set to `Text`, you can only use `BinHex` or `UuEncode` values for this parameter. If the *MessageFormat* parameter is set to `Mime`, you can only use `BinHex`, `AppleSingle`, or `AppleDouble` values for this parameter.
MessageBodyFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageBodyFormat	The *MessageBodyFormat* parameter specifies the message body format for messages sent to the mail contact. The valid values for this parameter are: • `Text` • `Html` • `TextAndHtml` By default, this parameter is set to `TextAndHtml`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`,

			the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to set this parameter to `Html` or `TextAndHtml`, you must also set the *MessageFormat* parameter to `Mime`.
MessageFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageFormat	The *MessageFormat* parameter specifies the message format for messages sent to the mail contact. The valid values for this parameter are: • `Text` • `Mime` By default, this parameter is set to `Mime`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you

			want to change the *MessageFormat* parameter from `Mime` to `Text`, you must also change the *MessageBodyFormat* parameter to `Text`.
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the *Reply to* address for the mail user. By default, the primary SMTP address is the same as the *ExternalEmailAddress* parameter value. We recommend that you don't set this parameter unless you're in a cross-forest scenario. If you want to set the primary SMTP address to be a different address from the external email address, you need to set the *EmailAddressPolicyEnabled* parameter to `$false` by using the **Set-MailContact** cmdlet, otherwise the mail user's

			primary SMTP address will use the *ExternalEmailAddress* parameter value, regardless of the value in the *PrimarySMTPAddress* parameter.
UsePreferMessageFormat	Optional	System.Boolean	The *UsePreferMessageFormat* parameter specifies whether the recipient-preferred message format settings override the global settings for mail sent to this user. When set to `$true`, this parameter specifies that the recipient-preferred message format settings override the global settings for mail sent to this user.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a

			value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailContact

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailContact** cmdlet to retrieve all of the specified contact attributes from Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailContact [-Anr <String>] <COMMON PARAMETERS>

Get-MailContact [-Identity <MailContactIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-Filter <String>]
[-IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-ReadFromDomainController
<SwitchParameter>] [-RecipientTypeDetails <RecipientTypeDetails[]>] [-
ResultSize <Unlimited>] [-SortBy <String>] [-UsnForReconciliationSearch
<Int64>]
```

Examples

EXAMPLE 1

This example retrieves the mail-enabled contact Arlene.

```
Get-MailContact -Identity Arlene | Format-List
```

Detailed Description

The **Get-MailContact** cmdlet retrieves all attributes of the specified contact. No parameters are required. If the cmdlet is run without a parameter, a complete list of contacts for the Exchange organization is returned.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**

Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the account to use to gain access to Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable

			properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailContactIdParameter	The *Identity* parameter identifies the contact. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com

			• `User Principal Name` Example: JPhillips@contoso.com
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't

			accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies a container in which to limit the results. Either an organizational unit (OU) or a domain can be specified. Also, the canonical name should be specified, for example: • OU: westcoast.contoso.com/users • Domain: westcoast.contoso.com
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain

			controller in the user's domain. If you set the recipient scope to include all recipients in the forest and you don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers running Microsoft Exchange.
RecipientTypeDetails	Optional	Microsoft.Exchange.Data.Directory.Recipient.RecipientTypeDetails[]	The *RecipientTypeDetails* parameter specifies the type of recipients returned. Recipient types are divided into recipient types and subtypes. Each recipient type contains all common properties for all subtypes. For example, the type UserMailbox represents a user account in Active Directory that has an associated mailbox. Because there are several mailbox types, each

			mailbox type is identified by the *RecipientTypeDetails* parameter. For example, a conference room mailbox has *RecipientTypeDetails* set to ConferenceRoomMailbox, whereas a user mailbox has *RecipientTypeDetails* set to UserMailbox. You can select from the following values: • `ArbitrationMailbox` • `ConferenceRoomMailbox` • `Contact` • `DiscoveryMailbox` • `DynamicDistributionGroup` • `EquipmentMailbox` • `ExternalManagedContact` • `ExternalManagedDistributionGroup` • `LegacyMailbox` • `LinkedMailbox` • `MailboxPlan` • `MailContact` • `MailForestContact` • `MailNonUniversalGroup` • `MailUniversalDistributionGroup` • `MailUniversalSecurityGroup` • `MailUser` • `PublicFolder` • `RoleGroup` • `RoomList` • `RoomMailbox` • `SharedMailbox` • `SystemAttendantMailbox` • `SystemMailbox` • `User` • `UserMailbox`
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the total number

			of recipient objects to return. If not specified, the parameter returns all results that match the filter.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. Sorting is performed one attribute at a time and is always performed in ascending order.
UsnForReconciliationSearch	Optional	System.Int64	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailContact

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MailContact** cmdlet to create a mail-enabled contact.

```
New-MailContact -ExternalEmailAddress <ProxyAddress> -Name <String> [-
Alias <String>] [-ArbitrationMailbox <MailboxIdParameter>] [-Confirm
[<SwitchParameter>]] [-DisplayName <String>] [-DomainController <Fqdn>] [-
ExternalDirectoryObjectId <String>] [-FirstName <String>] [-Initials
<String>] [-LastName <String>] [-MacAttachmentFormat <BinHex | UuEncode |
AppleSingle | AppleDouble>] [-MessageBodyFormat <Text | Html |
TextAndHtml>] [-MessageFormat <Text | Mime>] [-ModeratedBy
<MultiValuedProperty>] [-ModerationEnabled <$true | $false>] [-
Organization <OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-OverrideRecipientQuotas
<SwitchParameter>] [-PrimarySmtpAddress <SmtpAddress>] [-
SendModerationNotifications <Never | Internal | Always>] [-
UsePreferMessageFormat <$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example creates a mail-enabled contact using the required parameters and the *OrganizationalUnit* parameter.

```
New-MailContact -Name "Chris Ashton" -ExternalEmailAddress
"Chris@tailspintoys.com" -OrganizationalUnit "Marketing"
```

Detailed Description

The **New-MailContact** cmdlet creates a new mail contact object in Active Directory, and then mail-enables the mail contact.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
ExternalEmailAddress	Required	Microsoft.Exchange.Data.ProxyAddress	The *ExternalEmailAddress* parameter specifies the target email address.
Name	Required	System.String	The *Name* parameter specifies the common name of the mail contact.

Alias	Optional	System.String	The *Alias* parameter specifies the alias of the mail contact.
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArbitrationMailbox* parameter specifies the mailbox used to manage the moderation process.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the name displayed in Microsoft Outlook for the mail contact.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the

			fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ExternalDirectoryObjectId	Optional	System.String	This parameter is reserved for internal Microsoft use.
FirstName	Optional	System.String	The *FirstName* parameter specifies the first name of the mail contact.
Initials	Optional	System.String	The *Initials* parameter specifies the initials of the mail contact.
LastName	Optional	System.String	The *LastName* parameter specifies the last name of the mail contact.
MacAttachmentFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MacAttachmentFormat	The *MacAttachmentFormat* parameter specifies the Apple Macintosh operating system attachment format for messages sent to the mail contact. The valid values for this parameter are: • `BinHex` • `UuEncode` • `AppleSingle` • `AppleDouble` By default, this parameter is set to `BinHex`.

			The acceptable values for the *MacAttachmentFormat* parameter are dependent on the *MessageFormat* parameter. If the *MessageFormat* parameter is set to `Text`, you can only use `BinHex` or `UuEncode` values for this parameter. If the *MessageFormat* parameter is set to `Mime`, you can only use `BinHex`, `AppleSingle`, or `AppleDouble` values for this parameter.
MessageBodyFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageBodyFormat	The *MessageBodyFormat* parameter specifies the message body format for messages sent to the mail contact. The valid values for this parameter are: • `Text` • `Html` • `TextAndHtml` By default, this parameter is set to `TextAndHtml`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat*

			parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to set this parameter to `Html` or `TextAndHtml`, you must also set the *MessageFormat* parameter to `Mime`.
MessageFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageFormat	The *MessageFormat* parameter specifies the message format for messages sent to the mail contact. The valid values for this parameter are: • `Text` • `Mime` By default, this parameter is set to `Mime`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to

			any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to change the *MessageFormat* parameter from `Mime` to `Text`, you must also change the *MessageBodyFormat* parameter to `Text`.
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users who are responsible for moderating the messages sent to this mailbox. To designate more than one user, separate the users with commas. This parameter is required if you set the *ModerationEnabled* parameter to `$true`. If you leave this parameter blank and there's a user already specified as the manager of this mailbox, the *ModeratedBy* parameter is automatically set by the

			ManagedBy parameter of the mailbox. Otherwise, an error is returned.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter enables or disables moderation for the mailbox. To enable moderation, set this parameter to `$true`. To disable moderation, set this parameter to `$false`. The default value is `$false`.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies the organizational unit (OU) to which the new contact is added, for example, redmond.contoso.com/contacts.
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in on-premises Exchange 2013. The *PrimarySmtpAddress*

			parameter specifies the primary SMTP address for the mail contact. By default, the primary SMTP address is generated based on the default email address policy. If you specify a primary SMTP address by using this parameter, the command sets the **EmailAddressPolicyEnabled** attribute of the mail contact to `$false`, and the email addresses of this mail contact aren't automatically updated based on email address policies.
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when a message they sent to the moderated distribution group is rejected by one of the moderators. You can specify one of the following values: • `Always` • `Internal` • `Never`

			Set this parameter to `Always` if you want notifications to be sent to all senders. Set this parameter to `Internal` if you want notifications to be sent only to the senders internal to your organization. Set this parameter to `Never` to disable all status notifications. The default value is `Never`.
UsePreferMessageFormat	Optional	System.Boolean	The *UsePreferMessageFormat* parameter specifies whether recipient preferred message format settings are used. When set to `$true`, this parameter specifies that the recipient preferred message format settings override the global settings for mail sent to this user.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object.

			By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailContact

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MailContact** cmdlet to delete an existing mail-enabled contact from Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MailContact -Identity <MailContactIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ForReconciliation
<SwitchParameter>] [-IgnoreDefaultScope <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the mail-enabled contact John from Active Directory.

```
Remove-MailContact -Identity contoso.com/john
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailContactIdParameter	The *Identity* parameter specifies the identity of the mail contact. You can use the following values: • ADObjectID • Distinguished name (DN) • GUID • Alias
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.

DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
ForReconciliation	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the

			following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailContact

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-09*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailContact** cmdlet to modify an existing mail-enabled contact in Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailContact -Identity <MailContactIdParameter> [-
AcceptMessagesOnlyFrom <MultiValuedProperty>] [-
AcceptMessagesOnlyFromDLMembers <MultiValuedProperty>] [-
AcceptMessagesOnlyFromSendersOrMembers <MultiValuedProperty>] [-Alias
<String>] [-ArbitrationMailbox <MailboxIdParameter>] [-
BypassModerationFromSendersOrMembers <MultiValuedProperty>] [-Confirm
[<SwitchParameter>]] [-CreateDTMFMap <$true | $false>] [-CustomAttribute1
<String>] [-CustomAttribute10 <String>] [-CustomAttribute11 <String>] [-
CustomAttribute12 <String>] [-CustomAttribute13 <String>] [-
CustomAttribute14 <String>] [-CustomAttribute15 <String>] [-
CustomAttribute2 <String>] [-CustomAttribute3 <String>] [-CustomAttribute4
<String>] [-CustomAttribute5 <String>] [-CustomAttribute6 <String>] [-
CustomAttribute7 <String>] [-CustomAttribute8 <String>] [-CustomAttribute9
<String>] [-DisplayName <String>] [-DomainController <Fqdn>] [-
EmailAddresses <ProxyAddressCollection>] [-EmailAddressPolicyEnabled
<$true | $false>] [-ExtensionCustomAttribute1 <MultiValuedProperty>] [-
ExtensionCustomAttribute2 <MultiValuedProperty>] [-
ExtensionCustomAttribute3 <MultiValuedProperty>] [-
ExtensionCustomAttribute4 <MultiValuedProperty>] [-
ExtensionCustomAttribute5 <MultiValuedProperty>] [-ExternalEmailAddress
<ProxyAddress>] [-ForceUpgrade <SwitchParameter>] [-
GenerateExternalDirectoryObjectId <SwitchParameter>] [-GrantSendOnBehalfTo
<MultiValuedProperty>] [-HiddenFromAddressListsEnabled <$true | $false>]
[-IgnoreDefaultScope <SwitchParameter>] [-MacAttachmentFormat <BinHex |
UuEncode | AppleSingle | AppleDouble>] [-MailTip <String>] [-
MailTipTranslations <MultiValuedProperty>] [-MaxReceiveSize <Unlimited>]
[-MaxRecipientPerMessage <Unlimited>] [-MaxSendSize <Unlimited>] [-
MessageBodyFormat <Text | Html | TextAndHtml>] [-MessageFormat <Text |
Mime>] [-ModeratedBy <MultiValuedProperty>] [-ModerationEnabled <$true |
$false>] [-Name <String>] [-PrimarySmtpAddress <SmtpAddress>] [-
RejectMessagesFrom <MultiValuedProperty>] [-RejectMessagesFromDLMembers
<MultiValuedProperty>] [-RejectMessagesFromSendersOrMembers
<MultiValuedProperty>] [-RemovePicture <SwitchParameter>] [-
RemoveSpokenName <SwitchParameter>] [-RequireSenderAuthenticationEnabled
<$true | $false>] [-SecondaryAddress <String>] [-SecondaryDialPlan
<UMDialPlanIdParameter>] [-SendModerationNotifications <Never | Internal |
Always>] [-SimpleDisplayName <String>] [-UMDtmfMap <MultiValuedProperty>]
[-UseMapiRichTextFormat <Never | Always | UseDefaultSettings>] [-
UsePreferMessageFormat <$true | $false>] [-UserCertificate
<MultiValuedProperty>] [-UserSMimeCertificate <MultiValuedProperty>] [-
WhatIf [<SwitchParameter>]] [-WindowsEmailAddress <SmtpAddress>]
```

Examples

EXAMPLE 1

This example sets John Rodman's external email address to john@contoso.com.

```
Set-MailContact -Identity "John Rodman" -
ExternalEmailAddress "john@contoso.com"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailContactIdParameter	The *Identity* parameter specifies the mail contact. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-

			5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
AcceptMessagesOnlyFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFrom* parameter specifies the mailbox users, mail users, and mail contacts that can send email messages to this mail contact. You can also specify Microsoft Exchange as a valid recipient for this parameter. If you configure a mail contact to accept messages only from the Microsoft Exchange recipient, it only receives system-generated messages. You can use any of the following values for the

			valid senders: • DN • Canonical name • GUID • Name • Display name • Alias • Exchange DN • Primary SMTP email address By default, this parameter is blank. This default value enables the mail contact to accept messages from all senders.
AcceptMessagesOnlyFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromDLMembers* parameter specifies the distribution groups whose members are allowed to send email messages to this mail contact. You can use any of the following values for the allowed distribution groups: • DN • Canonical name • GUID • Name • Display name • Legacy Exchange DN

			• Primary SMTP email address By default, this parameter is blank. This default value enables the mail contact to accept messages from all senders.
AcceptMessagesOnlyFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromSendersOrMembers* parameter specifies the recipients who are allowed to send email messages to this mail contact. You can use any of the following values for the allowed distribution groups: • DN • Canonical name • GUID • Name • Display name • Legacy Exchange DN • Primary SMTP email address By default, this parameter is blank. This default value enables the mail contact to accept messages from all senders.
Alias	Optional	System.String	The *Alias* parameter

			specifies the alias of the mail-enabled contact. An alias can contain letters, numbers, and the following punctuation marks and symbols: • ! • # • $ • % • ^ • & • * • + • - • . • / • = • ? • _ • { • } • | • ~
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArbitrationMailbox* parameter specifies the mailbox used to manage the moderation process.
BypassModerationFro	Optional	Microsoft.Exchange.Da	The

mSendersOrMembers		ta.MultiValuedProperty	*BypassModerationFromSendersOrMembers* parameter specifies the recipients whose messages bypass moderation when sending to this mail contact. You can use any of the following values: • DN • Canonical name • GUID • Name • Display name • Legacy Exchange DN • Primary SMTP email address By default, this value is blank. This default value ensures that all messages are moderated when this mail contact is configured for moderation. Senders that are designated as moderators for this mail contact are allowed to send messages to this mail contact.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to

			acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CreateDTMFMap	Optional	System.Boolean	The *CreateDTMFMap* parameter specifies that a dual tone multi-frequency (DTMF) map be created for the contact.
CustomAttribute1	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute10	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute11	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional

			information.
CustomAttribute12	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute13	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute14	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute15	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute2	Optional	System.String	The *CustomAttribute1* to

			CustomAttribute15 parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute3	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute4	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute5	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute6	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can

			use these attributes to store additional information.
CustomAttribute7	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute8	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute9	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the display name of the user.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EmailAddresses	Optional	Microsoft.Exchange.Data.ProxyAddressCollection	The *EmailAddresses* parameter specifies the email address of the mail contact. All valid Microsoft Exchange email address types may be used. You can specify multiple values for the *EmailAddresses* parameter as a comma-delimited list. If you include multiple values, the first email address becomes the mail contact's primary SMTP email address. **Important:** Exchange doesn't validate custom addresses for proper formatting. You must ensure that the custom address you specify complies with the format requirements for that address type. Because X.400 addresses are considered custom addresses in Exchange, they aren't validated, and

			you must provide the correct syntax when specifying an X.400 address.
EmailAddressPolicyEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *EmailAddressPolicyEnabled* parameter specifies whether the email addresses for the mailbox are automatically updated based on the email address policies defined.
ExtensionCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information

			about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list.

			Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute4	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.

ExtensionCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExternalEmailAddress	Optional	Microsoft.Exchange.Data.ProxyAddress	The *ExternalEmailAddress* parameter specifies the external email address of the recipient.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* parameter suppresses the following confirmation: "To save changes on object <*object name*>, the object must be upgraded to the current Exchange

			version. After the upgrade, this object can't be managed by an earlier version of the Exchange management tools. Do you want to continue to upgrade and save the object?" This confirmation appears when you upgrade a mail contact that was created in Exchange Server 2003. You can't manage an Exchange 2003 mail contact by using the Exchange Administration Center until you update the object's version.
GenerateExternalDirectoryObjectId	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
GrantSendOnBehalfTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *GrantSendOnBehalfTo* parameter specifies the DNs of recipients that can send messages on behalf of this contact.
HiddenFromAddressListsEnabled	Optional	System.Boolean	The *HiddenFromAddressListsEnabled* parameter specifies whether the contact appears in

			address lists. The possible values for this parameter are `$true` or `$false`. If the value is `$true`, the contact doesn't appear in the address list. The default value is `$false`.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* switch tells the command to ignore the default recipient scope setting for the Exchange Management Shell session, and to use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently available in the default scope. Using the *IgnoreDefaultScope* switch introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an

			appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
MacAttachmentFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MacAttachmentFormat	The *MacAttachmentFormat* parameter specifies the Apple Macintosh operating system attachment format for messages sent to the mail contact. The valid values for this parameter are: • `BinHex` • `UuEncode` • `AppleSingle` • `AppleDouble` By default, this parameter is set to `BinHex`. The acceptable values for the *MacAttachmentFormat* parameter are dependent on the *MessageFormat* parameter. If the *MessageFormat* parameter is set to `Text`, you can only use `BinHex` or `UuEncode` values for

			this parameter. If the *MessageFormat* parameter is set to `Mime`, you can only use `BinHex`, `AppleSingle`, or `AppleDouble` values for this parameter.
MailTip	Optional	System.String	The *MailTip* parameter specifies the message displayed to senders when they start drafting an email message to this recipient. The *MailTip* parameter message must be less than or equal to 250 characters.
MailTipTranslations	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *MailTipTranslations* parameter specifies additional languages when you want to provide the *MailTip* parameter information for this recipient in multiple languages. For each language, you must provide the locale, followed by a colon and the specific *MailTip* parameter value in that language. Each *MailTip* parameter value must be less than or equal to 250

			characters. Multiple languages can be separated by commas.
MaxReceiveSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxReceiveSize* parameter specifies the maximum size of email messages that can be received, from 1 kilobyte (KB) to 2,097,151 KB. If a value isn't specified, the limit is set to the maximum value.
MaxRecipientPerMessage	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxRecipientPerMessage* parameter specifies the maximum number of recipients for messages from this mail contact.
MaxSendSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxSendSize* parameter specifies the maximum size of email messages that can be

			sent, from 1 KB to 2,097,151 KB. If a value isn't specified, the limit is set to the maximum value.
MessageBodyFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageBodyFormat	The *MessageBodyFormat* parameter specifies the message body format for messages sent to the mail contact. The valid values for this parameter are: • `Text` • `Html` • `TextAndHtml` By default, this parameter is set to `TextAndHtml`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to set this parameter to `Html` or `TextAndHtml`, you must also set the

			MessageFormat parameter to `Mime`.
MessageFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageFormat	The *MessageFormat* parameter specifies the message format for messages sent to the mail contact. The valid values for this parameter are: • `Text` • `Mime` By default, this parameter is set to `Mime`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to change the *MessageFormat* parameter from `Mime` to `Text`, you must also change the *MessageBodyFormat*

			parameter to `Text`.
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users who are responsible for moderating the messages sent to the mail-enabled contact. To designate more than one user, separate the users with commas. This parameter is required if you set the *ModerationEnabled* parameter to `$true`. If you leave this parameter blank and there is a user who is already specified as the manager of this distribution group, the *ModeratedBy* parameter is automatically set by the *ManagedBy* parameter of the mail-enabled contact. Otherwise, an error is returned.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable moderation for the mail-enabled contact. The possible values for this parameter are `$true` or

			`$false`. To enable moderation, set this parameter to `$true`. To disable moderation, set this parameter to `$false`. The default value is `$false`.
Name	Optional	System.String	The *Name* parameter specifies the name of the mail-enabled contact.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in on-premises Exchange 2013. The *PrimarySmtpAddress* parameter specifies the primary SMTP email address.
RejectMessagesFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFrom* parameter specifies the recipients from which to reject messages. You can use any of the following values to specify the recipients: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN

			• Primary SMTP email address By default, this parameter is blank. This default value enables the mail contact to accept messages from all senders.
RejectMessagesFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromDLMembers* parameter specifies the distribution list members from which to reject messages. You can use any of the following values to specify the recipients: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank. This default value enables the mail contact to accept messages from all senders.
RejectMessagesFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedPropert	The *RejectMessagesFromSend*

		y	*ersOrMembers* parameter specifies the recipients who aren't allowed to send email messages to this mail contact. You can use any of the following values to specify the recipients: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank. This default value enables the mail contact to accept messages from all senders.
RemovePicture	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemovePicture* parameter specifies whether to remove the picture that a user has added to a mail contact. A picture file can be added to the mail contact by

			using the **Import-RecipientDataProperty** cmdlet.
RemoveSpokenName	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemoveSpokenName* parameter specifies whether to remove the spoken name that a user has added to a mail contact. A sound file can be added to the mail contact by using the **Import-RecipientDataProperty** cmdlet.
RequireSenderAuthenticationEnabled	Optional	System.Boolean	The *RequireSenderAuthenticationEnabled* parameter specifies whether to accept messages only from authenticated recipients. The default value is `$false`.
SecondaryAddress	Optional	System.String	The *SecondaryAddress* parameter specifies the secondary address that's used by the Unified Messaging (UM)-enabled contact.

SecondaryDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *SecondaryDialPlan* parameter specifies a secondary UM dial plan to use. This parameter is provided to create a secondary proxy address.
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when they send a message to the moderated distribution group. You can specify one of the following values: • `Always` • `Internal` • `Never` Set this parameter to `Always` if you want notifications to be sent to all senders. Set this parameter to `Internal` if you want notifications to be sent only to the senders internal to your organization. Set this parameter to `Never` to disable all status notifications.

			Note: The sender is always notified if the message is rejected by the moderators, regardless of the value of this parameter. The default value is `Never`.
SimpleDisplayName	Optional	System.String	The *SimpleDisplayName* parameter is used to display an alternative description of the object when only a limited set of characters is permitted. This limited set of characters consists of ASCII characters 26 through 126, inclusively.
UMDtmfMap	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UMDtmfMap* parameter specifies whether you want to create a user-defined DTMF map for the UM-enabled contact.
UseMapiRichTextFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.UseMapiRichTextFormat	The *UseMapiRichTextFormat* parameter specifies how messages in MAPI rich text format (RTF) are handled for this mail contact. Set this parameter to `Never` to convert all messages sent

			to this contact to plain text. Set this parameter to `Always` to always send messages to this contact in MAPI RTF. Set this parameter to `UseDefaultSettings` to have the format decided based on the setting configured in the MAPI client that sent the message.
UsePreferMessageFormat	Optional	System.Boolean	The *UsePreferMessageFormat* parameter specifies whether the message format settings configured for the mail contact override the global settings configured for the remote domain. The possible values for this parameter are `$true` or `$false`. Set this parameter to `$true` to have the message format settings configured for the mail contact to override any global settings. The default value is `$false`.
UserCertificate	Optional	Microsoft.Exchange.Data.MultiValuedPropert	This parameter is reserved for internal Microsoft use.

		y	
UserSMimeCertificate	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsEmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *WindowsEmailAddress* parameter specifies the Windows email address for this mailbox. This address isn't used by Exchange.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Disable-MailUser

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-05-05*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Disable-MailUser** cmdlet to mail-disable an existing user in Active Directory by removing the attributes used by Exchange.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Disable-MailUser <COMMON PARAMETERS>

Disable-MailUser [-Archive <SwitchParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: -Identity <MailUserIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-IgnoreDefaultScope
<SwitchParameter>] [-IgnoreLegalHold <SwitchParameter>] [-
IncludeSoftDeletedObjects <SwitchParameter>] [-
PreventRecordingPreviousDatabase <SwitchParameter>] [-WhatIf
[<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example mail-disables an existing user.

```
Disable-MailUser markus@contoso.com
```

Detailed Description

The **Disable-MailUser** cmdlet mail-disables an existing user by removing the Active Directory Exchange Server attributes of the user. The user isn't deleted from Active Directory.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description

Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailUserIdParameter	The *Identity* parameter specifies the identity of the mail user object. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example:

			JPhillips@contoso.com
Archive	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch can be used to suppress the confirmation prompt that appears by default when this cmdlet is run. To suppress the confirmation prompt, use the syntax -`confirm:$False`. You must include a colon (`:`) in the syntax.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory

			objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
IgnoreLegalHold	Optional	System.Management.Automation.SwitchParameter	The *IgnoreLegalHold* switch ignores the legal hold status of the mail user and allows you to disable the cloud-based mailbox on legal hold. **⚠Warning:** When you disable a mailbox, the mailbox is disconnected from the user account. After you disable a mailbox, you can't include it in a discovery search. Disconnected mailboxes are permanently deleted from the mailbox

			database after the deleted mailbox retention period expires. Check with your organization's legal or Human Resources department before disabling a mailbox on legal hold.
IncludeSoftDeletedObjects	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PreventRecordingPreviousDatabase	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet

Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Enable-MailUser

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-07-03*

This cmdlet is available only in on-premises Exchange Server 2013.

Use the **Enable-MailUser** cmdlet to mail-enable an existing user in Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Enable-MailUser -ExternalEmailAddress <ProxyAddress> [-MacAttachmentFormat
<BinHex | UuEncode | AppleSingle | AppleDouble>] [-MessageBodyFormat <Text
| Html | TextAndHtml>] [-MessageFormat <Text | Mime>] [-UsageLocation
<CountryInfo>] [-UsePreferMessageFormat <$true | $false>] <COMMON
PARAMETERS>


Enable-MailUser -ArchiveGuid <Guid> [-Archive <SwitchParameter>] [-
ArchiveName <MultiValuedProperty>] <COMMON PARAMETERS>


COMMON PARAMETERS: -Identity <UserIdParameter> [-AddOnSKUCapability
<MultiValuedProperty>] [-Alias <String>] [-BypassModerationCheck
<SwitchParameter>] [-Confirm [<SwitchParameter>]] [-DisplayName <String>]
[-DomainController <Fqdn>] [-IncludeSoftDeletedObjects <SwitchParameter>]
[-JournalArchiveAddress <SmtpAddress>] [-OverrideRecipientQuotas
<SwitchParameter>] [-PreserveEmailAddresses <SwitchParameter>] [-
PrimarySmtpAddress <SmtpAddress>] [-SKUAssigned <$true | $false>] [-
SKUCapability <None | BPOS_S_Deskless | BPOS_S_Standard |
BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard | BPOS_B_Standard |
BPOS_B_CustomDomain | BPOS_S_MidSize | BPOS_S_ArchiveAddOn |
BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn | BPOS_Unmanaged |
TOU_Signed | FederatedUser | Partner_Managed | MasteredOnPremise |
ResourceMailbox | ExcludedFromBackSync | UMFeatureRestricted |
RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example mail-enables user John with the external email address john@contoso.com.

```
Enable-MailUser -Identity John -ExternalEmailAddress
```

```
john@contoso.com
```

Detailed Description

The **Enable-MailUser** cmdlet mail-enables an existing user in Active Directory by adding the attributes required by Microsoft Exchange. The user's identity and an external email address are required. Mail-enabled users have an email address at the Exchange organization (for example, john@contoso.com), but they don't have an Exchange mailbox. Email messages addressed to the mail-enabled user are sent instead to the specified external email address.

Note:
To create a user who is mail-enabled at the time of creation, use the **New-MailUser** cmdlet.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
ArchiveGuid	Required	System.Guid	This parameter is reserved for internal Microsoft use.
ExternalEmailAddress	Required	Microsoft.Exchange.Data.ProxyAddress	The *ExternalEmailAddress* parameter specifies an email address outside the organization. Email messages sent to the mail-enabled user are sent to this external address.
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	The *Identity* parameter specifies the identity of the mail user object. This parameter accepts the following values: • `Alias`

			Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
AddOnSKUCapability	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
Alias	Optional	System.String	The *Alias* parameter

			specifies the alias of the user. An alias can contain letters, numbers, and the following punctuation marks and symbols: • ! • # • $ • % • ^ • & • * • + • - • . • / • = • ? • _ • { • } • | • ~
Archive	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
ArchiveName	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
BypassModerationCheck	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the display name for the mail user. The *DisplayName* parameter is the name that appears in the Exchange Administration Center.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
IncludeSoftDeletedObjects	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
JournalArchiveAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is reserved for internal Microsoft use.

MacAttachmentFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MacAttachmentFormat	The *MacAttachmentFormat* parameter specifies the Apple Macintosh operating system attachment format for messages sent to the mail user. The valid values for this parameter are: • `BinHex` • `UuEncode` • `AppleSingle` • `AppleDouble` By default, this parameter is set to `BinHex`. The acceptable values for the *MacAttachmentFormat* parameter are dependent on the *MessageFormat* parameter. If the *MessageFormat* parameter is set to `Text`, you can only use `BinHex` or `UuEncode` values for this parameter. If the *MessageFormat* parameter is set to `Mime`, you can only use `BinHex`, `AppleSingle`, or `AppleDouble` values for this parameter.
MessageBodyFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.	The *MessageBodyFormat* parameter specifies the

		MessageBodyFormat	message body format for messages sent to the mail user. The valid values for this parameter are: • `Text` • `Html` • `TextAndHtml` By default, this parameter is set to `TextAndHtml`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to set this parameter to `Html` or `TextAndHtml`, you must also set the *MessageFormat* parameter to `Mime`.
MessageFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageFormat	The *MessageFormat* parameter specifies the message format for messages sent to the mail user.

			The valid values for this parameter are: • `Text` • `Mime` By default, this parameter is set to `Mime`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to change the *MessageFormat* parameter from `Mime` to `Text`, you must also change the *MessageBodyFormat* parameter to `Text`.
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PreserveEmailAddresses	Optional	System.Management.Automation.SwitchPar	This parameter is reserved for internal Microsoft use.

		ameter	
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *PrimarySmtpAddress* parameter specifies the *Reply to* address for the mail user. By default, the primary SMTP address is the same as the *ExternalEmailAddress* parameter value. We recommend that you don't set this parameter unless you're in a cross-forest scenario. If you want to set the primary SMTP address to be a different address from the external email address, you need to set the *EmailAddressPolicyEnabled* parameter to `$false` by using the **Set-MailUser** cmdlet, otherwise the mail user's primary SMTP address will use the *ExternalEmailAddress* parameter value regardless of the value in the *PrimarySmtpAddress* parameter.
SKUAssigned	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.

SKUCapability	Optional	Microsoft.Exchange.Data.Directory.Capability	This parameter is reserved for internal Microsoft use.
UsageLocation	Optional	Microsoft.Exchange.Data.Directory.CountryInfo	This parameter is reserved for internal Microsoft use.
UsePreferMessageFormat	Optional	System.Boolean	The *UsePreferMessageFormat* parameter specifies whether the message format settings configured for the mailbox override the global settings configured for the domain. Set this parameter to `$true` to have the message format settings configured for the mailbox to override any global settings.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-MailUser

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-MailUser** cmdlet to retrieve the specified user's mail-related attributes from Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-MailUser [-ArchiveDatabase <DatabaseIdParameter>] <COMMON PARAMETERS>

Get-MailUser [-Anr <String>] <COMMON PARAMETERS>

Get-MailUser [-Identity <MailUserIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
Credential <PSCredential>] [-DomainController <Fqdn>] [-Filter <String>]
[-IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-ReadFromDomainController
<SwitchParameter>] [-ResultSize <Unlimited>] [-SoftDeletedMailUser
<SwitchParameter>] [-SortBy <String>] [-UsnForReconciliationSearch
<Int64>]
```

Examples

EXAMPLE 1

This example retrieves a complete list of mail-enabled users for the entire Exchange organization.

```
Get-MailUser
```

EXAMPLE 2

This example retrieves the settings for the mail user named Ed, and then the command is piped to the **Format-List** cmdlet for display.

```
Get-MailUser -Identity Ed | Format-List
```

Detailed Description

The **Get-MailUser** cmdlet retrieves all mail-related attributes of the specified user.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName**

			• **LastName** • **Alias**
ArchiveDatabase	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is reserved for internal Microsoft use.
Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the account to use to gain access to Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter

			indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.MailUserIdParameter	The *Identity* parameter identifies the user. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/

			ou=AdministrativeGroup/ cn=Recipients/ cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
IgnoreDefaultScope	Optional	System.Management. Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server

			automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies a container in which to limit the results. You can specify either an organizational unit (OU) or a domain. The canonical name should be specified, for example: • OU: westcoast.contoso.com/users • Domain: westcoast.contoso.com
ReadFromDomainController	Optional	System.Management.Automation.SwitchPar	This parameter is available only in on-

		ameter	premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest and don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Exchange.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of recipient objects to return. If not specified, the parameter returns all results that match the filter.
SoftDeletedMailUser	Optional	System.Management.Automation.SwitchPar	This parameter is reserved for internal Microsoft use.

		ameter	
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. Sorting is performed one attribute at a time and is always performed in ascending order.
UsnForReconciliationSearch	Optional	System.Int64	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

New-MailUser

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-08*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **New-MailUser** cmdlet to create a mail-enabled user.

For information about the parameter sets in the Syntax section below, see Syntax.

```
New-MailUser -ExternalEmailAddress <ProxyAddress> [-MacAttachmentFormat
<BinHex | UuEncode | AppleSingle | AppleDouble>] [-MessageBodyFormat <Text
| Html | TextAndHtml>] [-MessageFormat <Text | Mime>] [-UsageLocation
<CountryInfo>] [-UsePreferMessageFormat <$true | $false>] <COMMON
PARAMETERS>
```

```
New-MailUser -MicrosoftOnlineServicesID <WindowsLiveId> -Password
<SecureString> [-ExternalEmailAddress <ProxyAddress>] [-UsageLocation
<CountryInfo>] <COMMON PARAMETERS>


New-MailUser -ExternalEmailAddress <ProxyAddress> -Password <SecureString>
-UserPrincipalName <String> [-MacAttachmentFormat <BinHex | UuEncode |
AppleSingle | AppleDouble>] [-MessageBodyFormat <Text | Html |
TextAndHtml>] [-MessageFormat <Text | Mime>] [-UsageLocation
<CountryInfo>] [-UsePreferMessageFormat <$true | $false>] <COMMON
PARAMETERS>


New-MailUser -Password <SecureString> -WindowsLiveID <WindowsLiveId> [-
EvictLiveId <SwitchParameter>] [-ExternalEmailAddress <ProxyAddress>] [-
UsageLocation <CountryInfo>] <COMMON PARAMETERS>


New-MailUser -ImportLiveId <SwitchParameter> -WindowsLiveID
<WindowsLiveId> [-ExternalEmailAddress <ProxyAddress>] [-UsageLocation
<CountryInfo>] <COMMON PARAMETERS>


New-MailUser -UseExistingLiveId <SwitchParameter> -WindowsLiveID
<WindowsLiveId> [-BypassLiveId <SwitchParameter>] [-ExternalEmailAddress
<ProxyAddress>] [-NetID <NetID>] [-UsageLocation <CountryInfo>] <COMMON
PARAMETERS>


New-MailUser -FederatedIdentity <String> -WindowsLiveID <WindowsLiveId> [-
EvictLiveId <SwitchParameter>] [-ExternalEmailAddress <ProxyAddress>] [-
NetID <NetID>] <COMMON PARAMETERS>


New-MailUser -FederatedIdentity <String> -MicrosoftOnlineServicesID
<WindowsLiveId> [-NetID <NetID>] <COMMON PARAMETERS>


New-MailUser [-MicrosoftOnlineServicesID <WindowsLiveId>] <COMMON
PARAMETERS>


COMMON PARAMETERS: -Name <String> [-AddOnSKUCapability
<MultiValuedProperty>] [-Alias <String>] [-ArbitrationMailbox
<MailboxIdParameter>] [-Confirm [<SwitchParameter>]] [-DisplayName
<String>] [-DomainController <Fqdn>] [-ExternalDirectoryObjectId <String>]
[-FirstName <String>] [-ImmutableId <String>] [-Initials <String>] [-
Languages <MultiValuedProperty>] [-LastName <String>] [-
MailboxProvisioningConstraint <MailboxProvisioningConstraint>] [-
MailboxProvisioningPreferences <MultiValuedProperty>] [-ModeratedBy
<MultiValuedProperty>] [-ModerationEnabled <$true | $false>] [-
Organization <OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-OverrideRecipientQuotas
<SwitchParameter>] [-PrimarySmtpAddress <SmtpAddress>] [-
RemotePowerShellEnabled <$true | $false>] [-ResetPasswordOnNextLogon
<$true | $false>] [-SamAccountName <String>] [-SendModerationNotifications
<Never | Internal | Always>] [-SKUAssigned <$true | $false>] [-
SKUCapability <None | BPOS_S_Deskless | BPOS_S_Standard |
BPOS_S_Enterprise | BPOS_S_Archive | BPOS_L_Standard | BPOS_B_Standard |
BPOS_B_CustomDomain | BPOS_S_MidSize | BPOS_S_ArchiveAddOn |
BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn | BPOS_Unmanaged |
TOU_Signed | FederatedUser | Partner_Managed | MasteredOnPremise |
ResourceMailbox | ExcludedFromBackSync | UMFeatureRestricted |
RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example consists of two commands.

In the first command, the password *$password* of the type System.Security.SecureString is created. When the command is executed, the prompt "Enter password" is displayed. The string entered by the user becomes the new password string, *$password*.

The second command creates the mail-enabled user object Ed Meadows and assigns the newly created password to the object by means of the *Password* parameter. Ed is an employee at Tailspin Toys, but because he works closely with employees at Contoso, Ltd, he's being given an email address at contoso.com. Ed doesn't have a mailbox on the server running Exchange at Contoso. Email messages sent to Ed's contoso.com email address are sent to his external email address, Ed@tailspintoys.com.

```
$password = Read-Host "Enter password" -AsSecureString
New-MailUser -Name "Ed Meadows" -Password $password -
ExternalEmailAddress ed@tailspintoys.com -UserPrincipalName
ed@contoso.com -OrganizationalUnit contoso.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
ExternalEmailAddress	Required	Microsoft.Exchange.Data.ProxyAddress	The *ExternalEmailAddress* parameter specifies an email address outside of the organization. Email messages sent to the mail-enabled user are sent to this external address.

FederatedIdentity	Required	System.String	This parameter is available only in the cloud-based service. The *FederatedIdentity* parameter associates an on-premises Active Directory user with a user in the cloud.
ImportLiveId	Required	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *ImportLiveID* parameter imports an unmanaged Microsoft account (formerly known as a Windows Live ID) into the cloud-based domain. An unmanaged Microsoft account was created in the domain before the domain was enrolled in the cloud-based service. Importing a Microsoft account into the domain lets you save any settings associated with the Microsoft account, like instant messaging contacts. However, the Microsoft account is now subject to the security and privacy policies of the

			organization.
MicrosoftOnlineServicesID	Required	Microsoft.Exchange.Data.WindowsLiveId	The *MicrosoftOnlineServicesID* parameter specifies the user ID for the object. This parameter only applies to objects in the cloud-based service. It isn't available for on-premises deployments.
Name	Required	System.String	The *Name* parameter specifies the common name (CN) of the mail-enabled user.
Password	Required	System.Security.SecureString	The *Password* parameter specifies the password used by the mail user to secure his or her account.
UseExistingLiveId	Required	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *UseExistingLiveId* parameter uses the specified Microsoft account (formerly known as a Windows Live ID) that already exists in the cloud-based domain. The specified Microsoft account can't have a mail user associated with it.

UserPrincipalName	Required	System.String	This parameter is available only in on-premises Exchange 2013. The *UserPrincipalName* parameter specifies the name of a system user in an email address format (for example, ed@contoso.com).
WindowsLiveID	Required	Microsoft.Exchange.Data.WindowsLiveId	This parameter is available only in the cloud-based service. The *WindowsLiveID* parameter creates a Microsoft account (formerly known as a Windows Live ID) associated with the mail user.
AddOnSKUCapability	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
Alias	Optional	System.String	The *Alias* parameter specifies the alias of the user. An alias can contain letters, numbers, and the following punctuation marks and symbols: • ! • # • $

			• % • ^ • & • * • + • - • . • / • = • ? • _ • { • } • | • ~
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArbitrationMailbox* parameter specifies the mailbox used to manage the moderation process. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example:

			CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
BypassLiveId	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm*

			switch.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the name displayed in Microsoft Outlook for the mail user.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EvictLiveId	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *EvictLiveId* parameter removes an unmanaged Microsoft account (formerly known as a Windows Live ID) from the cloud-based domain. An unmanaged Microsoft account was created in the domain before the domain was enrolled in the cloud-based service.

			Evicting a Microsoft account from the domain lets you save any settings associated with the Microsoft account, like instant messaging contacts.
ExternalDirectoryObjectId	Optional	System.String	This parameter is reserved for internal Microsoft use.
FirstName	Optional	System.String	The *FirstName* parameter specifies the first name of the mail user.
ImmutableId	Optional	System.String	The *ImmutableId* parameter is used by GAL synchronization (GALSync) and specifies a unique and immutable identifier in the form of an SMTP address for an Exchange mailbox used for federated delegation when requesting Security Assertion Markup Language (SAML) tokens. If federation is configured for this mailbox and you don't set this parameter when you create the mailbox, Exchange creates the value for the immutable ID based upon

			the mailbox's **ExchangeGUID** and the federated account namespace, for example, 7a78e7c8-620e-4d85-99d3-c90d90f29699@mail.contoso.com. You must set the *ImmutableId* parameter if Active Directory Federation Services (AD FS) is deployed to allow single sign-on into an off-premises mailbox and AD FS is configured to use a different attribute than **ExchangeGUID** for sign-on token requests. Both, Exchange and AD FS must request the same token for the same user to ensure proper functionality for a cross-premises Exchange deployment scenario.
Initials	Optional	System.String	The *Initials* parameter specifies the initials of the mail user.
Languages	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.

LastName	Optional	System.String	The *LastName* parameter specifies the last name of the mail user.
MacAttachmentFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MacAttachmentFormat	The *MacAttachmentFormat* parameter specifies the Apple Macintosh operating system attachment format for messages sent to the mail user. The valid values for this parameter are: • `BinHex` • `UuEncode` • `AppleSingle` • `AppleDouble` By default, this parameter is set to `BinHex`. The acceptable values for the *MacAttachmentFormat* parameter are dependent on the *MessageFormat* parameter. If the *MessageFormat* parameter is set to `Text`, you can only use `BinHex` or `UuEncode` values for this parameter. If the *MessageFormat* parameter is set to `Mime`, you can only use `BinHex`, `AppleSingle`, or `AppleDouble` values for this parameter.

MailboxProvisioningConstraint	Optional	Microsoft.Exchange.Data.Directory.MailboxProvisioningConstraint	This parameter is reserved for internal Microsoft use.
MailboxProvisioningPreferences	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
MessageBodyFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageBodyFormat	The *MessageBodyFormat* parameter specifies the message body format for messages sent to the mail user. The valid values for this parameter are: • `Text` • `Html` • `TextAndHtml` By default, this parameter is set to `TextAndHtml`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to set this parameter to `Html` or `TextAndHtml`,

			you must also set the *MessageFormat* parameter to `Mime`.
MessageFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageFormat	The *MessageFormat* parameter specifies the message format for messages sent to the mail user. The valid values for this parameter are: • `Text` • `Mime` By default, this parameter is set to `Mime`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to change the *MessageFormat* parameter from `Mime` to `Text`, you must also change the

			MessageBodyFormat parameter to `Text`.
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users responsible for moderating the messages sent to this mail user. To designate more than one user, separate the users with commas. This parameter is required if you set the *ModerationEnabled* parameter to `$true`.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable or disable moderation for the mail user. The two possible values for this parameter are `$true` or `$false`. To enable moderation, set this parameter to `$true`. To disable moderation, set this parameter to `$false`. The default value is `$false`.
NetID	Optional	Microsoft.Exchange.Data.NetID	This parameter is reserved for internal Microsoft use.

Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies the organizational unit (OU) in which the new user is added (for example, redmond.contoso.com/contacts).
OverrideRecipientQuotas	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in on-premises Exchange 2013. The *PrimarySmtpAddress* parameter specifies the primary SMTP address for the mail user. By default, the primary SMTP address is generated based on the default email address policy. If you specify a primary SMTP address by using this parameter, the command sets the **EmailAddressPolicyEnabled** attribute of the mail user to `$false`, and the email addresses of this

			mail user aren't automatically updated based on email address policies.
RemotePowerShellEnabled	Optional	System.Boolean	The *RemotePowerShellEnabled* parameter specifies whether the user can use remote Windows PowerShell. Remote Windows PowerShell is required to open the Exchange Management Shell on Mailbox and Client Access servers. Access to remote Windows PowerShell is required even if you're trying to open the Shell on the local server. The valid values are `$True` and `$False`. The default value is `$True`.
ResetPasswordOnNextLogon	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *ResetPasswordOnNextLogon* parameter specifies whether the user must change the password at

			the next logon. If this parameter is set to `$true`, the user must change the password at the next logon.
SamAccountName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *SamAccountName* parameter defines the logon name used to support clients and servers running older versions of the operating system. This attribute must contain fewer than 20 characters. An account name can contain letters, numbers, and the following punctuation marks and symbols: • ! • # • $ • % • ^ • & • - • . • _ • { • }

			• | • ~
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when they send a message to the moderated distribution group. You can specify one of the following values: • `Always` • `Internal` • `Never` Set this parameter to `Always` if you want notifications to be sent to all senders. Set this parameter to `Internal` if you want notifications to be sent only to the senders who are internal to your organization. Set this parameter to `Never` to disable all status notifications. The default value is `Never`. **Note:** The sender is always notified if the message is rejected by the

			moderators, regardless of the value of this parameter.
SKUAssigned	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
SKUCapability	Optional	Microsoft.Exchange.Data.Directory.Capability	This parameter is reserved for internal Microsoft use.
UsageLocation	Optional	Microsoft.Exchange.Data.Directory.CountryInfo	This parameter is reserved for internal Microsoft use.
UsePreferMessageFormat	Optional	System.Boolean	The *UsePreferMessageFormat* parameter specifies whether the message format settings configured for the mailbox override the global settings configured for the domain. Set this parameter to `$true` to have the message format settings configured for the mailbox to override any global settings.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur

			without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Remove-MailUser

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-13*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Remove-MailUser** cmdlet to remove an existing mail-enabled user from Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Remove-MailUser -Identity <MailUserIdParameter> [-Confirm
[<SwitchParameter>]] [-DomainController <Fqdn>] [-ForReconciliation
<SwitchParameter>] [-IgnoreDefaultScope <SwitchParameter>] [-
IgnoreLegalHold <SwitchParameter>] [-KeepWindowsLiveID <SwitchParameter>]
[-Permanent <$true | $false>] [-WhatIf [<SwitchParameter>]]
```

Examples

EXAMPLE 1

This example removes the mail-enabled user Ed Meadows from Active Directory.

```
Remove-MailUser -Identity "Ed Meadows"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailUserIdParameter	The *Identity* parameter specifies the identity of the mail user. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-

			5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.

ForReconciliation	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the distinguished name (DN) for the *Identity*

			parameter. Other forms of identification, such as alias or GUID, aren't accepted.
IgnoreLegalHold	Optional	System.Management.Automation.SwitchParameter	The *IgnoreLegalHold* switch ignores the legal hold status of the mail user and allows you to remove the user account that's on legal hold and the associated cloud mailbox. **⚠Warning:** After you remove a mailbox, you can't include it in a discovery search. Depending on the command parameters you use, removed mailboxes are either purged immediately or when the deleted mailbox retention period expires. Check with your organization's legal or Human Resources department before disabling a mailbox that's on legal hold.
KeepWindowsLiveID	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in the cloud-based service. The *KeepWindowsLiveID* parameter preserves the Microsoft account (formerly known as a Windows Live ID)

			associated with the deleted mail user.
Permanent	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-MailUser

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-07-03*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-MailUser** cmdlet to modify the mail-related attributes of an existing user in Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-MailUser -Identity <MailUserIdParameter> [-AcceptMessagesOnlyFrom
<MultiValuedProperty>] [-AcceptMessagesOnlyFromDLMembers
<MultiValuedProperty>] [-AcceptMessagesOnlyFromSendersOrMembers
<MultiValuedProperty>] [-AddOnSKUCapability <MultiValuedProperty>] [-
AggregatedMailboxGuids <MultiValuedProperty>] [-Alias <String>] [-
ArbitrationMailbox <MailboxIdParameter>] [-ArchiveGuid <Guid>] [-
ArchiveName <MultiValuedProperty>] [-ArchiveQuota <Unlimited>] [-
ArchiveWarningQuota <Unlimited>] [-BypassLiveId <SwitchParameter>] [-
BypassModerationFromSendersOrMembers <MultiValuedProperty>] [-
CalendarVersionStoreDisabled <$true | $false>] [-Confirm
[<SwitchParameter>]] [-CreateDTMFMap <$true | $false>] [-CustomAttribute1
<String>] [-CustomAttribute10 <String>] [-CustomAttribute11 <String>] [-
CustomAttribute12 <String>] [-CustomAttribute13 <String>] [-
CustomAttribute14 <String>] [-CustomAttribute15 <String>] [-
CustomAttribute2 <String>] [-CustomAttribute3 <String>] [-CustomAttribute4
<String>] [-CustomAttribute5 <String>] [-CustomAttribute6 <String>] [-
CustomAttribute7 <String>] [-CustomAttribute8 <String>] [-CustomAttribute9
<String>] [-DisplayName <String>] [-DomainController <Fqdn>] [-
EmailAddresses <ProxyAddressCollection>] [-EmailAddressPolicyEnabled
<$true | $false>] [-EndDateForRetentionHold <DateTime>] [-ExchangeGuid
<Guid>] [-ExtensionCustomAttribute1 <MultiValuedProperty>] [-
ExtensionCustomAttribute2 <MultiValuedProperty>] [-
ExtensionCustomAttribute3 <MultiValuedProperty>] [-
ExtensionCustomAttribute4 <MultiValuedProperty>] [-
ExtensionCustomAttribute5 <MultiValuedProperty>] [-ExternalEmailAddress
<ProxyAddress>] [-FederatedIdentity <String>] [-ForceUpgrade
<SwitchParameter>] [-GrantSendOnBehalfTo <MultiValuedProperty>] [-
HiddenFromAddressListsEnabled <$true | $false>] [-IgnoreDefaultScope
<SwitchParameter>] [-ImmutableId <String>] [-JournalArchiveAddress
<SmtpAddress>] [-LitigationHoldDate <DateTime>] [-LitigationHoldEnabled
<$true | $false>] [-LitigationHoldOwner <String>] [-MacAttachmentFormat
<BinHex | UuEncode | AppleSingle | AppleDouble>] [-MailboxContainerGuid
<Guid>] [-MailboxProvisioningConstraint <MailboxProvisioningConstraint>]
[-MailboxProvisioningPreferences <MultiValuedProperty>] [-MailTip
<String>] [-MailTipTranslations <MultiValuedProperty>] [-MaxReceiveSize
<Unlimited>] [-MaxSendSize <Unlimited>] [-MessageBodyFormat <Text | Html |
TextAndHtml>] [-MessageFormat <Text | Mime>] [-MicrosoftOnlineServicesID
<SmtpAddress>] [-ModeratedBy <MultiValuedProperty>] [-ModerationEnabled
<$true | $false>] [-Name <String>] [-NetID <NetID>] [-Password
<SecureString>] [-PrimarySmtpAddress <SmtpAddress>] [-RecipientLimits
<Unlimited>] [-RecoverableItemsQuota <Unlimited>] [-
RecoverableItemsWarningQuota <Unlimited>] [-RejectMessagesFrom
<MultiValuedProperty>] [-RejectMessagesFromDLMembers
<MultiValuedProperty>] [-RejectMessagesFromSendersOrMembers
<MultiValuedProperty>] [-RemovePicture <SwitchParameter>] [-
RemoveSpokenName <SwitchParameter>] [-RequireSenderAuthenticationEnabled
<$true | $false>] [-ResetPasswordOnNextLogon <$true | $false>] [-
RetainDeletedItemsFor <EnhancedTimeSpan>] [-RetentionComment <String>] [-
RetentionHoldEnabled <$true | $false>] [-RetentionUrl <String>] [-
SamAccountName <String>] [-SecondaryAddress <String>] [-SecondaryDialPlan
<UMDialPlanIdParameter>] [-SendModerationNotifications <Never | Internal |
Always>] [-SimpleDisplayName <String>] [-SingleItemRecoveryEnabled <$true
| $false>] [-SKUAssigned <$true | $false>] [-SKUCapability <None |
BPOS_S_Deskless | BPOS_S_Standard | BPOS_S_Enterprise | BPOS_S_Archive |
BPOS_L_Standard | BPOS_B_Standard | BPOS_B_CustomDomain | BPOS_S_MidSize |
BPOS_S_ArchiveAddOn | BPOS_S_EopStandardAddOn | BPOS_S_EopPremiumAddOn |
BPOS_Unmanaged | TOU_Signed | FederatedUser | Partner_Managed |
MasteredOnPremise | ResourceMailbox | ExcludedFromBackSync |
UMFeatureRestricted | RichCoexistence | OrganizationCapabilityUMGrammar |
OrganizationCapabilityUMDataStorage | OrganizationCapabilityOABGen |
OrganizationCapabilityGMGen | OrganizationCapabilityClientExtensions |
BEVDirLockdown | OrganizationCapabilityUMGrammarReady |
OrganizationCapabilityMailRouting | OrganizationCapabilityManagement |
OrganizationCapabilityTenantUpgrade | OrganizationCapabilityScaleOut |
OrganizationCapabilityMessageTracking | OrganizationCapabilityPstProvider
```

```
| OrganizationCapabilitySuiteServiceStorage |
OrganizationCapabilityOfficeMessageEncryption |
OrganizationCapabilityMigration>] [-StartDateForRetentionHold <DateTime>]
[-UMDtmfMap <MultiValuedProperty>] [-UsageLocation <CountryInfo>] [-
UseMapiRichTextFormat <Never | Always | UseDefaultSettings>] [-
UsePreferMessageFormat <$true | $false>] [-UserCertificate
<MultiValuedProperty>] [-UserPrincipalName <String>] [-
UserSMimeCertificate <MultiValuedProperty>] [-WhatIf [<SwitchParameter>]]
[-WindowsEmailAddress <SmtpAddress>] [-WindowsLiveID <SmtpAddress>]
```

Examples

EXAMPLE 1

This example sets the email address outside the organization to which mail-enabled user John Woods' (john) email is sent.

```
Set-MailUser john -ExternalEmailAddress
john@tailspintoys.com
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.MailUserIdParameter	The *Identity* parameter specifies the mail user. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example:

			CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
AcceptMessagesOnlyFrom	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFrom* parameter specifies the mailbox users and mail users that can send email messages to this mail user. You can also specify Exchange as a valid recipient for this parameter. If you configure a mail user to accept messages only

			from the Exchange recipient, the mail user only receives system-generated messages. You can use one of the following values for the valid senders: • DN • Canonical name • GUID • Name • Display name • Alias • Exchange DN • Primary SMTP email address By default, this parameter is blank, which enables the mail user to accept messages from all senders.
AcceptMessagesOnlyFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromDLMembers* parameter specifies the distribution groups whose members are allowed to send email messages to this mail user. You can use any of the following values for the allowed distribution groups:

			• DN • Canonical name • GUID • Name • Display name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank, which enables the mail user to accept messages from all senders.
AcceptMessagesOnlyFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *AcceptMessagesOnlyFromSendersOrMembers* parameter specifies the recipients who are allowed to send email messages to this mail user. You can use any of the following values for the allowed distribution groups: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN • Primary SMTP email address

			By default, this parameter is blank, which enables the mail user to accept messages from all senders.
AddOnSKUCapability	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
AggregatedMailboxGuids	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
Alias	Optional	System.String	The *Alias* parameter specifies the alias of the user. An alias can contain letters, numbers, and the following punctuation marks and symbols: • ! • # • $ • % • ^ • & • * • + • - • . • / • = • ? • _ • {

			• } • | • ~
ArbitrationMailbox	Optional	Microsoft.Exchange.Configuration.Tasks.MailboxIdParameter	This parameter is available only in on-premises Exchange 2013. The *ArbitrationMailbox* parameter specifies the mailbox user who is used to manage the moderation process.
ArchiveGuid	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
ArchiveName	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
ArchiveQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *ArchiveQuota* parameter specifies the archive mailbox size at which it no longer accepts messages. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes.

			The value must be greater than the value of the *ArchiveWarningQuota* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes.
ArchiveWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	The *ArchiveWarningQuota* parameter specifies the archive mailbox size at which a warning message is sent to the user. When you enter a value, qualify the value with one of the following units: • B (bytes) • KB (kilobytes) • MB (megabytes) • GB (gigabytes) • TB (terabytes) Unqualified values are treated as bytes. The value must be less than the value of the *ArchiveQuota* parameter. The valid input range for either parameter is from 1 through 9223372036854775807 bytes.

BypassLiveId	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
BypassModerationFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *BypassModerationFromSendersOrMembers* parameter specifies the recipients whose messages bypass moderation when sending to this mail user. You can use any of the following values: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN • Primary SMTP email address This value makes sure that all messages are moderated when this mail user is configured for moderation. **Note:** Senders designated as moderators for this mail user are never moderated.
CalendarVersionStore	Optional	System.Boolean	This parameter is reserved

Disabled			for internal Microsoft use.
Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CreateDTMFMap	Optional	System.Boolean	The *CreateDTMFMap* parameter specifies that a dual tone multi-frequency (DTMF) map be created for the user.
CustomAttribute1	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute10	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.

CustomAttribute11	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute12	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute13	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute14	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute15	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify

			custom attributes. You can use these attributes to store additional information.
CustomAttribute2	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute3	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute4	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute5	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional

			information.
CustomAttribute6	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute7	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute8	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
CustomAttribute9	Optional	System.String	The *CustomAttribute1* to *CustomAttribute15* parameters specify custom attributes. You can use these attributes to store additional information.
DisplayName	Optional	System.String	The *DisplayName*

			parameter specifies the display name of the user.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EmailAddresses	Optional	Microsoft.Exchange.Data.ProxyAddressCollection	The *EmailAddresses* parameter specifies the email alias of the mail user. All valid Exchange email address types may be used. You can specify multiple values for the *EmailAddresses* parameter as a comma-delimited list. **Important:** Exchange doesn't validate custom addresses for proper formatting. You must ensure that the custom address you specify complies with the format requirements for that address type. Because X.400 addresses are considered custom addresses in Exchange, they're also not validated, and you must provide the

			correct syntax when specifying an X.400 address.
EmailAddressPolicyEnabled	Optional	System.Boolean	This parameter is available only in on-premises Exchange 2013. The *EmailAddressPolicyEnabled* parameter specifies whether the email addresses for the mailbox are automatically updated based on the email address policies defined. The two possible values for this parameter are `$true` or `$false`. When this parameter is set to `$true`, you can't change the *PrimarySmtpAddress* or *WindowsEmailAddress* parameters.
EndDateForRetentionHold	Optional	System.DateTime	The *EndDateForRetentionHold* parameter specifies the end date for retention hold for messaging records management (MRM). To use this parameter, the *RetentionHoldEnabled* parameter must be set to `$true`.

ExchangeGuid	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
ExtensionCustomAttribute1	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute2	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each

			ExtensionCustomAttribute parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute3	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttri	Optional	Microsoft.Exchange.Da	The

bute4		ta.MultiValuedProperty	*ExtensionCustomAttribute 1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values. For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExtensionCustomAttribute5	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ExtensionCustomAttribute 1-5* parameters specify custom attributes that store additional information. You can specify multiple values for these parameters as a comma delimited list. Each *ExtensionCustomAttribute* parameter can hold up to 1,300 values.

			For more information about custom attributes, see Custom attributes. For more information about using multivalued properties, see Modifying multivalued properties.
ExternalEmailAddress	Optional	Microsoft.Exchange.Data.ProxyAddress	The *ExternalEmailAddress* parameter specifies an email address outside the organization. Email messages sent to the user are sent to this external address.
FederatedIdentity	Optional	System.String	This parameter is available only in the cloud-based service. The *FederatedIdentity* parameter associates an on-premises Active Directory user with a Microsoft Office user.
ForceUpgrade	Optional	System.Management.Automation.SwitchParameter	The *ForceUpgrade* parameter suppresses the following confirmation: "To save changes on object <*object name*>, the object must be upgraded to the current Exchange version. After the upgrade, this object cannot be

			managed by an earlier version of the Exchange Management Tools. Do you want to continue to upgrade and save the object?" This confirmation occurs when you upgrade a mail user that was created in Microsoft Exchange Server 2003. You can't manage an Exchange 2003 mail user by using the Exchange Administration Center until you update the object's version.
GrantSendOnBehalfTo	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *GrantSendOnBehalfTo* parameter specifies the DN of recipients that can send messages on behalf of this user.
HiddenFromAddressListsEnabled	Optional	System.Boolean	The *HiddenFromAddressListsEnabled* parameter specifies whether the user appears in the address list. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`.

IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.

ImmutableId	Optional	System.String	The *ImmutableId* parameter is used by Outlook Live Directory Sync (OLSync) and specifies a unique and immutable identifier in the form of an SMTP address for an Exchange mailbox that's used for federated delegation when requesting Security Assertion Markup Language (SAML) tokens. If federation is configured for this mailbox and you don't set this parameter when you create the mailbox, Exchange will create the value for the immutable ID based upon the mailbox's **ExchangeGUID** and the federated account namespace, for example, 7a78e7c8-620e-4d85-99d3-c90d90f29699@mail.contoso.com. You must set the *ImmutableId* parameter if Active Directory Federation Services (AD FS) is deployed to allow single

			sign-on into off-premises mailboxes and AD FS is configured to use a different attribute than **ExchangeGUID** for sign-on token requests. Both, Exchange and AD FS must request the same token for the same user to ensure proper functionality for cross-premise Exchange organizations.
JournalArchiveAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is reserved for internal Microsoft use.
LitigationHoldDate	Optional	System.DateTime	The *LitigationHoldDate* parameter specifies the date when the mailbox is placed on litigation hold. The parameter is populated automatically when placing a mailbox on litigation hold. This can be used for informational or reporting purposes. **Note:** When using the Exchange Management Shell to place the mailbox on litigation hold, you can optionally specify any date as the *LitigationHoldDate*, but the mailbox is placed on litigation hold when the

			cmdlet is run.
LitigationHoldEnabled	Optional	System.Boolean	The *LitigationHoldEnabled* parameter specifies that the mailbox is under litigation hold and that messages can't be deleted from the user's account. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`. After a mailbox is placed on litigation hold, deleted items and all versions of changed items are retained in the Recoverable Items folder. Items that are purged from the dumpster are also retained and the items are held indefinitely. If you enable litigation hold, single-item recovery quotas aren't applied.
LitigationHoldOwner	Optional	System.String	The *LitigationHoldOwner* parameter specifies the user who placed the mailbox on litigation hold. The parameter is populated automatically when placing a mailbox on litigation hold. This can

			be used for informational and reporting purposes. **Note:** When using the Shell to place a mailbox on litigation hold, you can optionally specify a string value for this parameter.
MacAttachmentFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MacAttachmentFormat	The *MacAttachmentFormat* parameter specifies the Apple Macintosh operating system attachment format for messages sent to the mail user. The valid values for this parameter are: • `BinHex` • `UuEncode` • `AppleSingle` • `AppleDouble` By default, this parameter is set to `BinHex`. The acceptable values for the *MacAttachmentFormat* parameter are dependent on the *MessageFormat* parameter. If the *MessageFormat* parameter is set to `Text`, you can only use `BinHex` or `UuEncode` values for this parameter. If the *MessageFormat*

			parameter is set to `Mime`, you can only use `BinHex`, `AppleSingle`, or `AppleDouble` values for this parameter.
MailboxContainerGuid	Optional	System.Guid	This parameter is reserved for internal Microsoft use.
MailboxProvisioningConstraint	Optional	Microsoft.Exchange.Data.Directory.MailboxProvisioningConstraint	This parameter is reserved for internal Microsoft use.
MailboxProvisioningPreferences	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
MailTip	Optional	System.String	The *MailTip* parameter specifies the message displayed to senders when they start drafting an email message to this recipient. The *MailTip* parameter message must be less than or equal to 250 characters.
MailTipTranslations	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *MailTipTranslations* parameter specifies additional languages when you want to provide the *MailTip* parameter information for this recipient in multiple languages. For each

			language, you must provide the locale, followed by a colon and the specific *MailTip* parameter message in that language. Each *MailTipTranslations* parameter message must be less than or equal to 250 characters. Multiple languages can be separated by commas.
MaxReceiveSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxReceiveSize* parameter specifies the maximum size of email messages that can be received by the mail user, from 1 kilobyte (KB) through 2,097,151 KB. If not specified, there are no size restrictions.
MaxSendSize	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *MaxSendSize* parameter specifies the maximum size of email messages that can be sent

			by the mail user, from 1 KB through 2,097,151 KB. If not specified, there are no size restrictions.
MessageBodyFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageBodyFormat	The *MessageBodyFormat* parameter specifies the message body format for messages sent to the mail user. The valid values for this parameter are: • `Text` • `Html` • `TextAndHtml` By default, this parameter is set to `TextAndHtml`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to set this parameter to `Html` or `TextAndHtml`, you must also set the *MessageFormat*

			parameter to `Mime`.
MessageFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.MessageFormat	The *MessageFormat* parameter specifies the message format for messages sent to the mail user. The valid values for this parameter are: • `Text` • `Mime` By default, this parameter is set to `Mime`. The *MessageFormat* and *MessageBodyFormat* parameters are interdependent. If the *MessageFormat* parameter is set to `Mime`, the *MessageBodyFormat* parameter can be set to any valid value. However, if the *MessageFormat* parameter is set to `Text`, the *MessageBodyFormat* parameter can only be set to `Text`. Therefore, if you want to change the *MessageFormat* parameter from `Mime` to `Text`, you must also change the *MessageBodyFormat* parameter to `Text`.

MicrosoftOnlineServicesID	Optional	Microsoft.Exchange.Data.SmtpAddress	The *MicrosoftOnlineServicesID* parameter specifies the user ID for the object. This parameter only applies to objects in the cloud-based service. It isn't available for on-premises deployments.
ModeratedBy	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *ModeratedBy* parameter specifies the users responsible for moderating the messages sent to the distribution group. To designate more than one user, separate the users with commas. This parameter is required if you set the *ModerationEnabled* parameter to `$true`. If you leave this parameter blank and there is a user already specified as the manager of this distribution group, the *ModeratedBy* parameter is automatically set with the value in the *ManagedBy* parameter of the **Set-DistributionGroup** cmdlet. Otherwise, an

			error is returned.
ModerationEnabled	Optional	System.Boolean	The *ModerationEnabled* parameter specifies whether to enable moderation for the distribution group. The two possible values for this parameter are `$true` or `$false`. To enable moderation, set this parameter to `$true`. To disable moderation, set this parameter to `$false`. The default value is `$false`.
Name	Optional	System.String	The *Name* parameter specifies the name of the user.
NetID	Optional	Microsoft.Exchange.Data.NetID	This parameter is reserved for internal Microsoft use.
Password	Optional	System.Security.SecureString	This parameter is reserved for internal Microsoft use.
PrimarySmtpAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in on-premises Exchange 2013. The *PrimarySmtpAddress* parameter specifies the primary SMTP address.
RecipientLimits	Optional	Microsoft.Exchange.Da	This parameter is available only in on-

		ta.Unlimited	premises Exchange 2013. The *RecipientLimits* parameter specifies the maximum number of recipients for messages from this user.
RecoverableItemsQuota	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *RecoverableItemsQuota* parameter specifies the size limit for the Recoverable Items folder for a mail-enabled user that has a corresponding remote mailbox or remote archive mailbox in the cloud-based service.
RecoverableItemsWarningQuota	Optional	Microsoft.Exchange.Data.Unlimited	This parameter is available only in on-premises Exchange 2013. The *RecoverableItemsWarningQuota* parameter specifies the size of the Recoverable Items folder before Exchange logs an event to the application event log.
RejectMessagesFrom	Optional	Microsoft.Exchange.Da	The *RejectMessagesFrom*

		ta.MultiValuedProperty	parameter specifies the recipients from which to reject messages. You can use any of the following values to specify the recipients: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank, which enables the mail user to accept messages from all senders.
RejectMessagesFromDLMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromDLMembers* parameter specifies the distribution list members from which to reject messages. You can use any of the following values to specify the recipients: • DN • Alias • Canonical name

			• Display name • GUID • Name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank, which enables the mail user to accept messages from all senders.
RejectMessagesFromSendersOrMembers	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *RejectMessagesFromSendersOrMembers* parameter specifies the recipients who aren't allowed to send email messages to this mail user. You can use any of the following values to specify the recipients: • Alias • Canonical name • Display name • DN • GUID • Name • LegacyExchangeDN • Primary SMTP email address By default, this parameter is blank, which enables

			the mail user to accept messages from all senders.
RemovePicture	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemovePicture* parameter specifies whether to remove the picture that a user has added to a mailbox.
RemoveSpokenName	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *RemoveSpokenName* parameter specifies whether to remove the spoken name that a user has added to a mailbox.
RequireSenderAuthenticationEnabled	Optional	System.Boolean	The *RequireSenderAuthenticationEnabled* parameter specifies whether to accept messages only from authenticated recipients. The two possible values for this parameter are `$true` or `$false`. The default value is `$false`.

ResetPasswordOnNextLogon	Optional	System.Boolean	This parameter is available only in the cloud-based service. The *ResetPasswordOnNextLogon* parameter specifies whether to require mail users to change their Microsoft account (formerly known as Windows Live ID) password the next time they sign in to the cloud-based service. If the *ResetPasswordOnNextLogon* parameter is set to `$true`, it requires mail users to change their Microsoft account password the next time they sign in to the cloud-based service. The two possible values for this parameter are `$true` or `$false`.
RetainDeletedItemsFor	Optional	Microsoft.Exchange.Data.EnhancedTimeSpan	The *RetainDeletedItemsFor* parameter specifies the length of time to keep deleted items. To specify a value, enter it as a time span:

			dd.hh:mm:ss where d = days, h = hours, m = minutes, and s = seconds. For example, to specify a 15-hour interval, enter 15:00:00.
RetentionComment	Optional	System.String	The *RetentionComment* parameter specifies a comment displayed in Outlook regarding the user's retention hold status. This comment can be set only if the *RetentionHoldEnabled* parameter is set to `$true`. This comment should be localized to the user's preferred language.
RetentionHoldEnabled	Optional	System.Boolean	The *RetentionHoldEnabled* parameter specifies whether retention hold is enabled for messaging retention policies. The two possible values for this parameter are `$true` or `$false`. To set the start date for retention hold, use the *StartDateForRetentionHol*

			d parameter.
RetentionUrl	Optional	System.String	The *RetentionUrl* parameter specifies the URL or an external web page with additional details about the organization's messaging retention policies. This URL can be used to expose details regarding retention policies in general, which is usually a customized legal or IT website for the company.
SamAccountName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *SamAccountName* parameter specifies the logon name used to support clients and servers running older versions of the operating system, such as Microsoft Windows NT 4.0, Windows 98, Windows 95, and LAN Manager. This attribute must contain fewer than 20 characters. An account name can contain letters, numbers, and the

			following punctuation marks and symbols: • ! • # • $ • % • ^ • & • - • . • _ • { • } • | • ~
SecondaryAddress	Optional	System.String	The *SecondaryAddress* parameter specifies the secondary address used by the Unified Messaging (UM)-enabled user.
SecondaryDialPlan	Optional	Microsoft.Exchange.Configuration.Tasks.UMDialPlanIdParameter	The *SecondaryDialPlan* parameter specifies a secondary UM dial plan to use. This parameter is provided to create a secondary proxy address.
SendModerationNotifications	Optional	Microsoft.Exchange.Data.Directory.Recipient.TransportModerationNotificationFlags	The *SendModerationNotifications* parameter specifies whether status notifications are sent to users when they send a

			message to the moderated distribution group. You can use one of the following values: • `Always` • `Internal` • `Never` If you want notifications to be sent to all senders, set this parameter to `Always`. If you want notifications to be sent only to the senders who are internal to your organization, set this parameter to `Internal`. To disable all status notifications, set this parameter to `Never`. **Note:** The sender is always notified if the message is rejected by the moderators, regardless of the value of this parameter. The default value is `Never`.
SimpleDisplayName	Optional	System.String	The *SimpleDisplayName* parameter is used to display an alternative description of the object when only a limited set of characters is permitted.

			This limited set of characters consists of ASCII characters 26 through 126, inclusively.
SingleItemRecoveryEnabled	Optional	System.Boolean	The *SingleItemRecoveryEnabled* parameter specifies whether to prevent the Recovery Items folder from being purged. When this parameter is set to `$true`, it prevents the Recovery Items folder from being purged. It also prevents any items from being removed that have been deleted or edited. The possible values for this parameter are `$true` or `$false`. The default value is `$false`.
SKUAssigned	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
SKUCapability	Optional	Microsoft.Exchange.Data.Directory.Capability	This parameter is reserved for internal Microsoft use.
StartDateForRetentionHold	Optional	System.DateTime	The *StartDateForRetentionHold* parameter specifies the start date for retention hold for MRM. To use this parameter, the

			RetentionHoldEnabled parameter must be set to `$true`.
UMDtmfMap	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UMDtmfMap* parameter specifies whether you want to create a user-defined DTMF map for the UM-enabled user.
UsageLocation	Optional	Microsoft.Exchange.Data.Directory.CountryInfo	This parameter is reserved for internal Microsoft use.
UseMapiRichTextFormat	Optional	Microsoft.Exchange.Data.Directory.Recipient.UseMapiRichTextFormat	The *UseMapiRichTextFormat* parameter specifies how messages in MAPI rich text format (RTF) are handled for this mail user. You can use one of the following values: • `Never` • `Always` • `UseDefaultSettings` To convert all messages sent to this user to plain text, set this parameter to `Never`. To always send messages to this user in MAPI RTF, set this parameter to `Always`. To have the format determined based on the

			setting configured in the MAPI client that sent the message, set this parameter to `UseDefaultSettings`.
UsePreferMessageFormat	Optional	System.Boolean	The *UsePreferMessageFormat* parameter specifies whether the message format settings configured for the mail user override the global settings configured for the remote domain. The two possible values for this parameter are `$true` or `$false`. To have the message format settings configured for the mail user to override any global settings, set this parameter to `$true`.
UserCertificate	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
UserPrincipalName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *UserPrincipalName* parameter specifies a UPN for the useront.

UserSMimeCertificate	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsEmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *WindowsEmailAddress* parameter specifies the Windows email address for this mailbox. This address isn't used by Exchange.
WindowsLiveID	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is available only in the cloud-based service. The *WindowsLiveID* parameter renames the Microsoft account associated with the mail user.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-Recipient

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-31*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-Recipient** cmdlet to return a list of recipient objects from Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-Recipient [-RecipientPreviewFilter <String>] <COMMON PARAMETERS>

Get-Recipient [-BookmarkDisplayName <String>] [-Identity
<RecipientIdParameter>] [-IncludeBookmarkObject <$true | $false>] <COMMON
PARAMETERS>

Get-Recipient [-Database <DatabaseIdParameter>] <COMMON PARAMETERS>

Get-Recipient [-Anr <String>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
AuthenticationType <Managed | Federated>] [-Capabilities
<MultiValuedProperty>] [-Credential <PSCredential>] [-DomainController
<Fqdn>] [-Filter <String>] [-IgnoreDefaultScope <SwitchParameter>] [-
Organization <OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-Properties <String[]>] [-PropertySet
<All | ControlPanel | ConsoleSmallSet | ConsoleLargeSet | Minimum>] [-
ReadFromDomainController <SwitchParameter>] [-RecipientType
<RecipientType[]>] [-RecipientTypeDetails <RecipientTypeDetails[]>] [-
ResultSize <Unlimited>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example retrieves information about all the recipients in your organization.

```
Get-Recipient -ResultSize unlimited
```

EXAMPLE 2

This example retrieves information about all the mail contacts in your organization and sorts them by name.

```
Get-Recipient -RecipientType MailContact -SortBy Name
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient provisioning permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)**

			• **DisplayName** • **FirstName** • **LastName** • **Alias**
AuthenticationType	Optional	Microsoft.Exchange.Data.Directory.AuthenticationType	This parameter is available only in the cloud-based service. The *AuthenticationType* parameter specifies the recipient by authentication type. Use one of the following values: • `Federated` • `Managed`
BookmarkDisplayName	Optional	System.String	This parameter is reserved for internal Microsoft use.
Capabilities	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the user name and password to use to access Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-**

			Credential cmdlet. For more information, see Get-Credential.
Database	Optional	Microsoft.Exchange.Configuration.Tasks.DatabaseIdParameter	This parameter is available only in on-premises Exchange 2013. The *Database* parameter specifies a mailbox database. Use this parameter to return all recipients stored on a specific mailbox database. Use the mailbox database *Name* property as the value for this parameter.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable

			properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.RecipientIdParameter	The *Identity* parameter specifies the recipient. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com

			• `User Principal Name` Example: JPhillips@contoso.com
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't

			accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
IncludeBookmarkObject	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies an organizational unit (OU) or domain canonical name, and is used to limit the results. If you use this parameter, you only get recipients in the container that you specify, for example: • OU: westcoast.contoso.com/users • Domain: westcoast.contoso.com
Properties	Optional	System.String[]	This parameter is reserved for internal Microsoft use.
PropertySet	Optional	Microsoft.Exchange.Data.Directory.Managem	This parameter is reserved for internal Microsoft use.

		ent.PropertySet	
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest, and if you don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information. **Note:** By default, the recipient scope is set to the domain that hosts your servers that run Microsoft Exchange.
RecipientPreviewFilter	Optional	System.String	The *RecipientPreviewFilter* parameter specifies a recipient filter that would define the recipients returned by this

			command. You can create a custom recipient filter for a dynamic distribution group, an address list, or an email address policy. To verify that the recipient filter you specified will return the recipients you want, you can pass the OPATH filter specified in the **RecipientFilter** property for that dynamic distribution group, address list, or email address policy to the *RecipientPreviewFilter* parameter and preview the list of recipients.
RecipientType	Optional	Microsoft.Exchange.Data.Directory.Recipient.RecipientType[]	The *RecipientType* parameter specifies the type of recipients to return. You can use one or more of the following values: • `DynamicDistributionGroup` • `UserMailbox` • `MailUser` • `MailContact` • `MailUniversalDistributionGroup` • `MailUniversalSecurityGroup` • `MailNonUniversalGroup` • `PublicFolder`
RecipientTypeDetails	Optional	Microsoft.Exchange.Data.Directory.Recipient.	The *RecipientTypeDetails* parameter specifies the type of recipients

		RecipientTypeDetails[]	returned. Recipient types are divided into recipient types and subtypes. Each recipient type contains all common properties for all subtypes. For example, the type UserMailbox represents a user account in Active Directory that has an associated mailbox. Because there are several mailbox types, each mailbox type is identified by the *RecipientTypeDetails* parameter. For example, a conference room mailbox has *RecipientTypeDetails* set to ConferenceRoomMailbox, whereas a user mailbox has *RecipientTypeDetails* set to UserMailbox. You can select from the following values: • `ConferenceRoomMailbox` • `Contact` • `DiscoveryMailbox` • `DynamicDistributionGroup` • `EquipmentMailbox` • `ExternalManagedContact` • `ExternalManagedDistributionGroup` • `LegacyMailbox` • `LinkedMailbox` • `MailboxPlan` • `MailContact` • `MailForestContact`

			• `MailNonUniversalGroup` • `MailUniversalDistributionGroup` • `MailUniversalSecurityGroup` • `MailUser` • `PublicFolder` • `RoleGroup` • `RoomList` • `RoomMailbox` • `SharedMailbox` • `SystemAttendantMailbox` • `SystemMailbox` • `User` • `UserMailbox`
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return. If you want to return all recipients that match the filter, use `unlimited` for the value of this parameter. The default value is 1000.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute by which to sort the results. You can sort by only one attribute at a time. You can sort by the following attributes: • **Alias** • **DisplayName** • **Name** The results are sorted in ascending order.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-SecurityPrincipal

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-14*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-SecurityPrincipal** cmdlet to return a list of security principals.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-SecurityPrincipal [-Identity <ExtendedSecurityPrincipalIdParameter>]
[-DomainController <Fqdn>] [-Filter <String>] [-IncludeDomainLocalFrom
<SmtpDomain>] [-Organization <OrganizationIdParameter>] [-
OrganizationalUnit <ExtendedOrganizationalUnitIdParameter>] [-ResultSize
<Unlimited>] [-RoleGroupAssignable <SwitchParameter>] [-Types
<MultiValuedProperty>]
```

Examples

EXAMPLE 1

This example retrieves security principals in the OU People, well-known security principals, and domain local groups from the domain Contoso.com.

```
Get-SecurityPrincipal -OrganizationalUnit
OU=People,DC=Contoso,DC=com -IncludeDomainLocalFrom
Contoso.com
```

EXAMPLE 2

This example retrieves security principals from the Legal department by using the *Filter* parameter. Only security principals matching the filter condition are retrieved.

```
Get-SecurityPrincipal -Filter {Department -eq "Legal"} -
IncludeDomainLocalFrom Contoso.com
```

EXAMPLE 3

This example retrieves a single security principal explicitly specified by using the *Identity* parameter.

```
Get-SecurityPrincipal -Identity Administrator -
IncludeDomainLocalFrom Contoso.com
```

EXAMPLE 4

This example retrieves well-known security principals by pipelining the results from the **Get-SecurityPrincipal** cmdlet to the **Where-Object** command. The results are pipelined to the **Format-Table** command. Only the *Name* and *SID* parameters are selected to be included in the final output.

```
Get-SecurityPrincipal -IncludeDomainLocalFrom Contoso.com |
? {$_.Type -eq "WellKnownSecurityPrincipal"} | ft Name,SID
-AutoSize
```

Note:

The question mark character (?) is an alias for the **Where-Object** command. `Ft` is an alias for the **Format-Table** command. Both aliases are included by default in the Windows PowerShell command-line interface.

Detailed Description

Security principals are entities, such as users or security groups, which can be assigned permissions and user rights.

Note:

If the *IncludeDomainLocalFrom* parameter is specified along with the *Filter* or *Identity* parameters, the cmdlet doesn't return domain local security groups. This cmdlet is required for internal Exchange Administration Center functionality.

The **Get-SecurityPrincipal** cmdlet is used by the Exchange Administration Center in Microsoft Exchange Server 2013 to populate fields that display recipient information.

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Active Directory Domain Services server settings" entry in the Exchange and Shell infrastructure permissions topic.

Parameters

Parameter	Required	Type	Description
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter specifies one or more attributes and their corresponding values to restrict the security principals returned by the command. When the *Filter* parameter is used, only those security principals matching the filter conditions are returned.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.ExtendedSecurityPrincipalIdParameter	The *Identity* parameter specifies the identity of the security principal. When the security principal is explicitly specified by using this parameter, no

			additional security principals are returned.
IncludeDomainLocalFrom	Optional	Microsoft.Exchange.Data.SmtpDomain	This parameter is available only in on-premises Exchange 2013. The *IncludeDomainLocalFrom* parameter specifies the FQDN of an Active Directory domain. The command returns domain local groups from the specified domain.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.ExtendedOrganizationalUnitIdParameter	The *OrganizationalUnit* parameter specifies an organizational unit (OU), container, or domain name. It's used to limit the results. If you use this parameter, you only get mailboxes in the OU, container, or domain that you specify, in addition to well-known security

			principals and domain local groups from the domain you specify in the *IncludeDomainLocalFrom* parameter.
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of objects to return. The default value is 1000. To return all objects that match the query, use the value `unlimited`.
RoleGroupAssignable	Optional	System.Management.Automation.SwitchParameter	The *RoleGroupAssignable* switch filters security principals and returns only those entities that can be assigned to an RBAC role group.
Types	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *Types* parameter isn't available at this time.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Get-User

Exchange Management Shell > Exchange 2013 cmdlets > Users and Groups Cmdlets >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Get-User** cmdlet to retrieve all users in the forest that match the specified conditions.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Get-User [-Anr <String>] <COMMON PARAMETERS>

Get-User [-Identity <UserIdParameter>] <COMMON PARAMETERS>

COMMON PARAMETERS: [-AccountPartition <AccountPartitionIdParameter>] [-
Arbitration <SwitchParameter>] [-ConsumerNetID <NetID>] [-Credential
<PSCredential>] [-DomainController <Fqdn>] [-Filter <String>] [-
IgnoreDefaultScope <SwitchParameter>] [-Organization
<OrganizationIdParameter>] [-OrganizationalUnit
<OrganizationalUnitIdParameter>] [-PublicFolder <SwitchParameter>] [-
ReadFromDomainController <SwitchParameter>] [-RecipientTypeDetails
<RecipientTypeDetails[]>] [-ResultSize <Unlimited>] [-SoftDeletedUser
<SwitchParameter>] [-SortBy <String>]
```

Examples

EXAMPLE 1

This example retrieves information about users in the Marketing OU.

```
Get-User -OrganizationalUnit "Marketing"
```

EXAMPLE 2

This example uses the *Filter* parameter to retrieve information about all users that have the word Manager at the end of their title.

```
Get-User -Filter "Title -like '*Manager'"
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
AccountPartition	Optional	Microsoft.Exchange.Configuration.Tasks.AccountPartitionIdParameter	This parameter is reserved for internal Microsoft use.
Anr	Optional	System.String	The *Anr* parameter specifies a string on which to perform an ambiguous name resolution (ANR) search. You can specify a partial string and search for objects with an attribute that matches that string. The default attributes searched are: • **CommonName (CN)** • **DisplayName** • **FirstName** • **LastName** • **Alias**
Arbitration	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Arbitration* parameter specifies that the mailbox for which you are

			executing the command is an arbitration mailbox. Arbitration mailboxes are used for managing approval workflow. For example, an arbitration mailbox is used for handling moderated recipients and distribution group membership approval.
ConsumerNetID	Optional	Microsoft.Exchange.Data.NetID	This parameter is reserved for internal Microsoft use.
Credential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *Credential* parameter specifies the account used to read Active Directory. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013. The *DomainController*

			parameter specifies the fully qualified domain name (FQDN) of the domain controller that retrieves data from Active Directory.
Filter	Optional	System.String	The *Filter* parameter indicates the OPath filter used to filter recipients. For more information about the filterable properties, see Filterable properties for the -Filter parameter.
Identity	Optional	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	The *Identity* parameter specifies the identity of the user object. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips

			• GUID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2 • Immutable ID Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • Legacy Exchange DN Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • SMTP Address Example: Jeff.Phillips@contoso.com • User Principal Name Example: JPhillips@contoso.com
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the

			IgnoreDefaultScope parameter introduces the following restrictions: • You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted. • You can't use the *OrganizationalUnit* and *Identity* parameters together. • You can't use the *Credential* parameter.
Organization	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationIdParameter	The *Organization* parameter is reserved for internal Microsoft use.
OrganizationalUnit	Optional	Microsoft.Exchange.Configuration.Tasks.OrganizationalUnitIdParameter	The *OrganizationalUnit* parameter returns objects only from the specified organizational unit (OU).
PublicFolder	Optional	System.Management.Automation.SwitchParameter	The *PublicFolder* parameter specifies that the user object for which

			you're executing the command is a public folder mailbox. Public folder mailboxes are specially designed mailboxes to store the hierarchy and content of public folders. This parameter is required to retrieve information for a public folder mailbox.
ReadFromDomainController	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *ReadFromDomainController* parameter specifies that the user information is read from a domain controller in the user's domain. If you set the recipient scope to include all recipients in the forest, and if you don't use this parameter, it's possible that the user information is read from a global catalog with outdated information. If you use this parameter, multiple reads might be necessary to get the information.

			Note: By default, the recipient scope is set to the domain that hosts your servers running Exchange.
RecipientTypeDetails	Optional	Microsoft.Exchange.Data.Directory.Recipient.RecipientTypeDetails[]	The *RecipientTypeDetails* parameter specifies the type of recipients returned. Recipient types are divided into types and subtypes. Each recipient type contains all common properties for all subtypes. For example, the type `UserMailbox` represents a user account in Active Directory that has an associated mailbox. Because there are several mailbox types, each mailbox type is identified by the *RecipientTypeDetails* parameter. For example, a conference room mailbox has *RecipientTypeDetails* set to `RoomMailbox`, whereas a user mailbox has *RecipientTypeDetails* set to `UserMailbox`. You can use the following values: • `DisabledUser` • `DiscoveryMailbox` • `EquipmentMailbox` • `LegacyMailbox`

			• LinkedMailbox • LinkedUser • MailUser • RemoteEquipmentMailbox • RemoteRoomMailbox • RemoteSharedMailbox • RemoteTeamMailbox • RemoteUserMailbox • RoomMailbox • SharedMailbox • TeamMailbox • User • UserMailbox
ResultSize	Optional	Microsoft.Exchange.Data.Unlimited	The *ResultSize* parameter specifies the maximum number of results to return.
SoftDeletedUser	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
SortBy	Optional	System.String	The *SortBy* parameter specifies the attribute to sort by. This parameter sorts by a single attribute in ascending order.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Set-User

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-03-15*

This cmdlet is available in on-premises Exchange Server 2013 and in the cloud-based service. Some parameters and settings may be exclusive to one environment or the other.

Use the **Set-User** cmdlet to modify user attributes in Active Directory.

For information about the parameter sets in the Syntax section below, see Syntax.

```
Set-User -Identity <UserIdParameter> [-AllowUMCallsFromNonUsers <None |
SearchEnabled>] [-Arbitration <SwitchParameter>] [-ArchiveRelease <None |
E14 | E15>] [-AssistantName <String>] [-BusinessNetID <NetID>] [-
CertificateSubject <MultiValuedProperty>] [-City <String>] [-Company
<String>] [-Confirm [<SwitchParameter>]] [-CopyShadowAttributes
<SwitchParameter>] [-CountryOrRegion <CountryInfo>] [-CreateDTMFMap <$true
| $false>] [-Department <String>] [-DisplayName <String>] [-
DomainController <Fqdn>] [-EnableAccount <SwitchParameter>] [-Fax
<String>] [-FirstName <String>] [-GenerateExternalDirectoryObjectId
<SwitchParameter>] [-GeoCoordinates <GeoCoordinates>] [-HomePhone
<String>] [-IgnoreDefaultScope <SwitchParameter>] [-Initials <String>] [-
InPlaceHoldsRaw <MultiValuedProperty>] [-LastName <String>] [-LEOEnabled
<$true | $false>] [-LinkedCredential <PSCredential>] [-
LinkedDomainController <String>] [-LinkedMasterAccount <UserIdParameter>]
[-MailboxRelease <None | E14 | E15>] [-Manager <UserContactIdParameter>]
[-MicrosoftOnlineServicesID <SmtpAddress>] [-MobilePhone <String>] [-Name
<String>] [-NetID <NetID>] [-Notes <String>] [-Office <String>] [-OtherFax
<MultiValuedProperty>] [-OtherHomePhone <MultiValuedProperty>] [-
OtherTelephone <MultiValuedProperty>] [-Pager <String>] [-Phone <String>]
[-PhoneticDisplayName <String>] [-PostalCode <String>] [-PostOfficeBox
<MultiValuedProperty>] [-PublicFolder <SwitchParameter>] [-
RemotePowerShellEnabled <$true | $false>] [-ResetPasswordOnNextLogon
<$true | $false>] [-SamAccountName <String>] [-SeniorityIndex <Int32>] [-
SimpleDisplayName <String>] [-SKUAssigned <$true | $false>] [-
StateOrProvince <String>] [-StreetAddress <String>] [-TelephoneAssistant
<String>] [-Title <String>] [-UMCallingLineIds <MultiValuedProperty>] [-
UMDtmfMap <MultiValuedProperty>] [-UpgradeDetails <String>] [-
UpgradeMessage <String>] [-UpgradeRequest <None | TenantUpgrade |
PrestageUpgrade | CancelPrestageUpgrade | PilotUpgrade |
TenantUpgradeDryRun>] [-UpgradeStage <None | SyncedWorkItem |
StartPilotUpgrade | StartOrgUpgrade | MoveArbitration | MoveRegularUser |
MoveCloudOnlyArchive | MoveRegularPilot | MoveCloudOnlyArchivePilot |
CompleteOrgUpgrade>] [-UpgradeStageTimeStamp <DateTime>] [-UpgradeStatus
<None | NotStarted | InProgress | Warning | Error | Cancelled | Complete |
ForceComplete>] [-UserPrincipalName <String>] [-WebPage <String>] [-WhatIf
[<SwitchParameter>]] [-WindowsEmailAddress <SmtpAddress>] [-WindowsLiveID
<SmtpAddress>]
```

Examples

EXAMPLE 1

This example sets the display name for user Jill Frank.

```
Set-User -Identity Contoso\Jill -DisplayName "Jill Frank"
```

EXAMPLE 2

This example unlinks the linked mailbox Kweku@fabrikam.com and converts it to a user mailbox by setting the *LinkedMasterAccount* parameter to `$null`.

Important:
Performing this procedure on a linked mailbox removes all permissions on the mailbox such as Send As, Full Access, folder, and calendar delegation.

```
Set-User -Identity Kweku@fabrikam.com -LinkedMasterAccount
$null
```

Detailed Description

You need to be assigned permissions before you can run this cmdlet. Although all parameters for this cmdlet are listed in this topic, you may not have access to some parameters if they're not included in the permissions assigned to you. To see what permissions you need, see the "Recipient Provisioning Permissions" section in the Recipients Permissions topic.

Parameters

Parameter	Required	Type	Description
Identity	Required	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	The *Identity* parameter specifies the identity of the user. This parameter accepts the following values: • `Alias` Example: JPhillips • `Canonical DN` Example: Atlanta.Corp.Contoso.Com/Users/JPhillips • `Display Name` Example: Jeff Phillips • `Distinguished Name (DN)` Example: CN=JPhillips,CN=Users,DC=Atlanta,DC=Corp,DC=contoso,DC=com • `Domain\Account` Example: Atlanta\JPhillips • `GUID` Example: fb456636-fe7d-

			4d58-9d15-5af57d0354c2 • `Immutable ID` Example: fb456636-fe7d-4d58-9d15-5af57d0354c2@contoso.com • `Legacy Exchange DN` Example: /o=Contoso/ou=AdministrativeGroup/cn=Recipients/cn=JPhillips • `SMTP Address` Example: Jeff.Phillips@contoso.com • `User Principal Name` Example: JPhillips@contoso.com
AllowUMCallsFromNonUsers	Optional	Microsoft.Exchange.Data.Directory.Recipient.AllowUMCallsFromNonUsersFlags	The *AllowUMCallsFromNonUsers* parameter specifies whether to exclude the user from directory searches.
Arbitration	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *Arbitration* parameter specifies that the mailbox for which you are executing the command is an arbitration mailbox. Arbitration mailboxes are used for managing approval workflow. For

			example, an arbitration mailbox is used for handling moderated recipients and distribution group membership approval.
ArchiveRelease	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MailboxRelease	This parameter is reserved for internal Microsoft use.
AssistantName	Optional	System.String	The *AssistantName* parameter specifies the name of the user's assistant.
BusinessNetID	Optional	Microsoft.Exchange.Data.NetID	This parameter is reserved for internal Microsoft use.
CertificateSubject	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *CertificateSubject* parameter specifies the value of the subject field of the user's digital certificate.
City	Optional	System.String	The *City* parameter specifies the user's city.
Company	Optional	System.String	The *Company* parameter specifies the user's company.

Confirm	Optional	System.Management.Automation.SwitchParameter	The *Confirm* switch causes the command to pause processing and requires you to acknowledge what the command will do before processing continues. You don't have to specify a value with the *Confirm* switch.
CopyShadowAttributes	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
CountryOrRegion	Optional	Microsoft.Exchange.Data.Directory.CountryInfo	The *CountryOrRegion* parameter specifies the user's country or region.
CreateDTMFMap	Optional	System.Boolean	The *CreateDTMFMap* parameter specifies that a dual-tone multiple-frequency (DTMF) map be created for the user.
Department	Optional	System.String	The *Department* parameter specifies the user's department.
DisplayName	Optional	System.String	The *DisplayName* parameter specifies the user's display name.
DomainController	Optional	Microsoft.Exchange.Data.Fqdn	This parameter is available only in on-premises Exchange 2013.

			The *DomainController* parameter specifies the fully qualified domain name (FQDN) of the domain controller that writes this configuration change to Active Directory.
EnableAccount	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
Fax	Optional	System.String	The *Fax* parameter specifies the user's fax number.
FirstName	Optional	System.String	The *FirstName* parameter specifies the user's given name.
GenerateExternalDirectoryObjectId	Optional	System.Management.Automation.SwitchParameter	This parameter is reserved for internal Microsoft use.
GeoCoordinates	Optional	Microsoft.Exchange.Data.GeoCoordinates	The *GeoCoordinates* parameter specifies the user's physical location in latitude, longitude, and altitude coordinates. Use this parameter to specify the global position of physical resources, such as conference rooms. You have to specify one of the following sets of

			coordinates; use semicolons to separate the values. • Latitude and longitude; for example, `"47.644125;-122.122411"` • Latitude, longitude, and altitude; for example, `"47.644125;-122.122411;161.432"`
HomePhone	Optional	System.String	The *HomePhone* parameter specifies the user's home telephone number.
IgnoreDefaultScope	Optional	System.Management.Automation.SwitchParameter	This parameter is available only in on-premises Exchange 2013. The *IgnoreDefaultScope* parameter instructs the command to ignore the default recipient scope setting for the Exchange Management Shell session and use the entire forest as the scope. This allows the command to access Active Directory objects that aren't currently in the default scope. Using the *IgnoreDefaultScope* parameter introduces the following restrictions:

			• You can't use the *DomainController* parameter. The command uses an appropriate global catalog server automatically. • You can only use the DN for the *Identity* parameter. Other forms of identification, such as alias or GUID, aren't accepted.
Initials	Optional	System.String	The *Initials* parameter specifies the user's initials.
InPlaceHoldsRaw	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is reserved for internal Microsoft use.
LastName	Optional	System.String	The *LastName* parameter specifies the user's surname.
LEOEnabled	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
LinkedCredential	Optional	System.Management.Automation.PSCredential	This parameter is available only in on-premises Exchange 2013. The *LinkedCredential* parameter specifies credentials to use to access the domain controller specified by the

			LinkedDomainController parameter. You can only use the *LinkedCredential* parameter with a linked user. This parameter requires the creation and passing of a credential object. This credential object is created by using the **Get-Credential** cmdlet. For more information, see Get-Credential.
LinkedDomainController	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *LinkedDomainController* parameter specifies the domain controller in the forest where the user account resides, if this user is a linked user. The domain controller in the forest where the user account resides is used to get security information for the account specified by the *LinkedMasterAccount* parameter.

			This parameter is required only if you're connecting a linked user.
LinkedMasterAccount	Optional	Microsoft.Exchange.Configuration.Tasks.UserIdParameter	This parameter is available only in on-premises Exchange 2013. The *LinkedMasterAccount* parameter specifies the master account in the forest where the user account resides, if this user is a linked user. The master account is the account to which the user links. The master account grants access to the user. You can use one of the following values: • GUID • DN • *Domain\Account* • UPN • LegacyExchangeDN • SmtpAddress • Alias • `$null` If you set this parameter's value to `$null`, you will unlink the account and convert the linked mailbox into a non-linked user mailbox. The mailbox

			won't retain the permissions previously set on it such as Send As, full access, folder, and calendar delegation. This parameter is required only if you're connecting a linked user.
MailboxRelease	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.MailboxRelease	This parameter is reserved for internal Microsoft use.
Manager	Optional	Microsoft.Exchange.Configuration.Tasks.UserContactIdParameter	The *Manager* parameter specifies the user's manager.
MicrosoftOnlineServicesID	Optional	Microsoft.Exchange.Data.SmtpAddress	The *MicrosoftOnlineServicesID* parameter specifies the user ID for the object. This parameter only applies to objects in the cloud-based service. It isn't available for on-premises deployments.
MobilePhone	Optional	System.String	The *MobilePhone* parameter specifies the user's primary mobile phone number.
Name	Optional	System.String	The *Name* parameter specifies the user's

			common name.
NetID	Optional	Microsoft.Exchange.Data.NetID	This parameter is reserved for internal Microsoft use.
Notes	Optional	System.String	The *Notes* parameter specifies additional information about the user.
Office	Optional	System.String	The *Office* parameter specifies the user's physical office name or number.
OtherFax	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *OtherFax* parameter specifies the user's alternative fax number.
OtherHomePhone	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *OtherHomePhone* parameter specifies the user's alternative home telephone number.
OtherTelephone	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *OtherTelephone* parameter specifies the user's alternative telephone number.
Pager	Optional	System.String	The *Pager* parameter specifies the user's pager number.
Phone	Optional	System.String	The *Phone* parameter specifies the user's office telephone number.

PhoneticDisplayName	Optional	System.String	The *PhoneticDisplayName* parameter specifies a phonetic pronunciation of the *DisplayName* parameter. The maximum length of this parameter value is 255 characters.
PostalCode	Optional	System.String	The *PostalCode* parameter specifies the user's zip code or postal code.
PostOfficeBox	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *PostOfficeBox* parameter specifies the user's post office box number.
PublicFolder	Optional	System.Management.Automation.SwitchParameter	The *PublicFolder* parameter specifies that the user for which you're executing the command is a public folder mailbox. Public folder mailboxes are specially designed mailboxes to store the hierarchy and content of public folders. This parameter is required to change the settings for a public folder mailbox.
RemotePowerShellEnabled	Optional	System.Boolean	The *RemotePowerShellEnabled* parameter specifies

			whether the user can use remote PowerShell. remote PowerShell is required to open the Exchange Management Shell or the Exchange Administration Center. Access to remote PowerShell is required even if you're trying to open the Shell or the Exchange Administration Center on the local server. The valid values are `$true` and `$false`. The default value depends on the management roles assigned to the user.
ResetPasswordOnNextLogon	Optional	System.Boolean	The *ResetPasswordOnNextLogon* parameter specifies whether the user's password must be reset the next time the user logs on. The two possible values for this parameter are `$true` or `$false`. The default value is `$true`.
SamAccountName	Optional	System.String	This parameter is available only in on-premises Exchange 2013.

			The *SamAccountName* parameter specifies the logon name used to support clients and servers running older versions of the operating system, such as Microsoft Windows NT 4.0, Windows 98, Windows 95, and LAN Manager. This attribute must contain fewer than 20 characters.
SeniorityIndex	Optional	System.Int32	The *SeniorityIndex* parameter specifies the order in which this user will display in a hierarchical address book. A user with a value of 2 will display higher in an address book than a user with a value of 1.
SimpleDisplayName	Optional	System.String	The *SimpleDisplayName* parameter is used to display an alternative description of the object when only a limited set of characters is permitted. This limited set of characters consists of ASCII characters 26 through 126, inclusively.

SKUAssigned	Optional	System.Boolean	This parameter is reserved for internal Microsoft use.
StateOrProvince	Optional	System.String	The *StateOrProvince* parameter specifies the user's state or province.
StreetAddress	Optional	System.String	The *StreetAddress* parameter specifies the user's physical address.
TelephoneAssistant	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *TelephoneAssistant* parameter specifies the telephone number of the user's assistant.
Title	Optional	System.String	The *Title* parameter specifies the user's title.
UMCallingLineIds	Optional	Microsoft.Exchange.Data.MultiValuedProperty	This parameter is available only in on-premises Exchange 2013. The *UMCallingLineIds* parameter specifies telephone numbers or extensions that can be mapped to a Unified Messaging (UM)-enabled user. You can specify more than one telephone number for each user, separated by a comma.

			This parameter accepts digits less than 128 characters in length and may include an optional plus sign (+) preceding the numbers. Each UM-enabled user must have a unique *UMCallingLineIds* parameter value.
UMDtmfMap	Optional	Microsoft.Exchange.Data.MultiValuedProperty	The *UMDtmfMap* parameter specifies whether you want to create a user-defined DTMF map for the UM-enabled user.
UpgradeDetails	Optional	System.String	This parameter is reserved for internal Microsoft use.
UpgradeMessage	Optional	System.String	This parameter is reserved for internal Microsoft use.
UpgradeRequest	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UpgradeRequestTypes	This parameter is reserved for internal Microsoft use.
UpgradeStage	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UpgradeStage	This parameter is reserved for internal Microsoft use.
UpgradeStageTimeStamp	Optional	System.DateTime	This parameter is reserved for internal Microsoft use.

UpgradeStatus	Optional	Microsoft.Exchange.Data.Directory.SystemConfiguration.UpgradeStatusTypes	This parameter is reserved for internal Microsoft use.
UserPrincipalName	Optional	System.String	This parameter is available only in on-premises Exchange 2013. The *UserPrincipalName* parameter specifies the UPN of the user.
WebPage	Optional	System.String	The *WebPage* parameter specifies the user's Web page.
WhatIf	Optional	System.Management.Automation.SwitchParameter	The *WhatIf* switch instructs the command to simulate the actions that it would take on the object. By using the *WhatIf* switch, you can view what changes would occur without having to apply any of those changes. You don't have to specify a value with the *WhatIf* switch.
WindowsEmailAddress	Optional	Microsoft.Exchange.Data.SmtpAddress	The *WindowsEmailAddress* parameter specifies the email address of the recipient.

WindowsLiveID	Optional	Microsoft.Exchange.Data.SmtpAddress	This parameter is reserved for internal Microsoft use.

Input Types

To see the input types that this cmdlet accepts, see Cmdlet Input and Output Types. If the Input Type field for a cmdlet is blank, the cmdlet doesn't accept input data.

Return Types

To see the return types, which are also known as output types, that this cmdlet accepts, see Cmdlet Input and Output Types. If the Output Type field is blank, the cmdlet doesn't return data.

Exchange admin center in Exchange 2013

Exchange Server 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-05*

The Exchange Admin Center (EAC) is the web-based management console in Microsoft Exchange Server 2013 that's optimized for on-premises, online, and hybrid Exchange deployments. The EAC replaces the Exchange Management Console (EMC) and the Exchange Control Panel (ECP), which were the two interfaces used to manage Exchange Server 2010.

One advantage a web-based EAC provides is that you can partition Internet and intranet access from within the ECP IIS virtual directory. With this functionality, you can control whether users are allowed to have Internet access to the EAC from outside of your organization, while still allowing an end user to access Outlook Web App Options. For more information, see Turn off access to the Exchange admin center.

Looking for the Exchange Online version of this topic? See **Exchange admin center in Exchange Online**.

Looking for the Exchange Online Protection version of this topic? See **Exchange admin center in Exchange Online Protection**.

Contents

Accessing the EAC

Because the EAC is now a web-based management console, you'll need to use the ECP virtual directory URL to access the console from your web browser. In most cases the EAC's URL will look similar to the following:

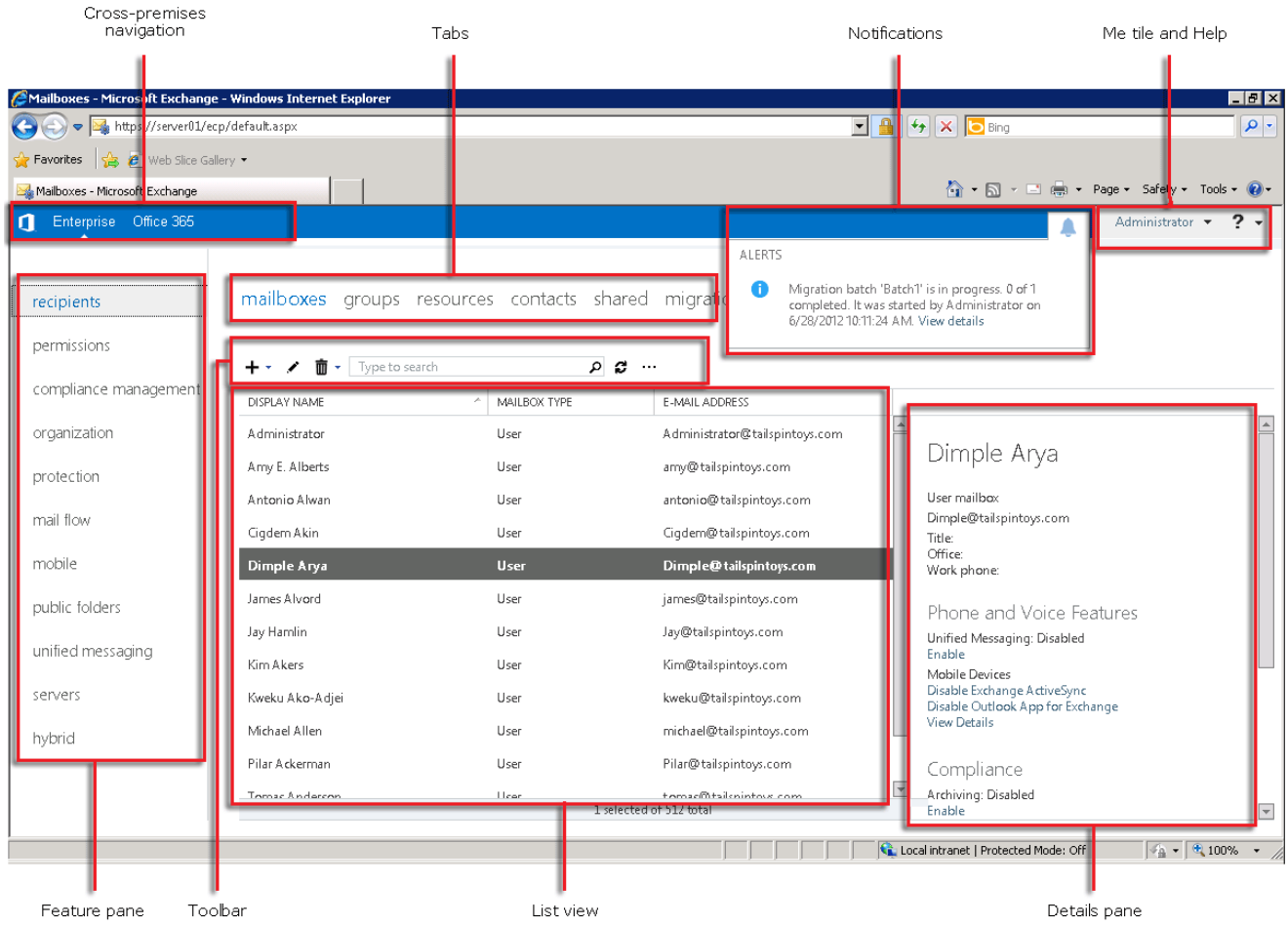
- **Internal URL:** `https://<CASServerName>/ecp` The internal URL is used to access the EAC from within your organization's firewall.
- **External URL:** `https://mail.contoso.com/ecp` The external URL is used to access the EAC from outside of your organization's firewall. Some organizations may want to turn off external access to the EAC. For details, see Turn off access to the Exchange admin center.

To locate the internal or external URL for the EAC, you can use the Get-EcpVirtualDirectory cmdlet. For details, see Find the internal and external URLs for the Exchange admin center.

If you're in a coexistence scenario, where you're running Exchange 2010 and Exchange 2013 in the same organization, and your mailbox is still housed on the Exchange 2010 Mailbox server, the browser will default to the Exchange 2010 ECP. You can access the EAC by adding the Exchange version to the URL. For example, to access the EAC whose virtual directory is hosted on the Client Access server CAS15-NA, use the following URL: `https://CAS15-NA/ecp?ExchClientVer=15`. Conversely, if you want to access the Exchange 2010 ECP and your mailbox resides on an Exchange 2013 Mailbox server, use the following URL: `https://CAS14-NA/ecp?ExchClientVer=14`.

Common user interface elements in the EAC

The section describes the user interface elements that are common across the EAC.



Cross-premises navigation

The cross-premises navigation allows you to easily switch between your Exchange Online and on-premises Exchange deployments. If you don't have an Exchange Online organization, the link will direct you to the Office 365 sign-up page. To learn more, see **Exchange Server 2013 Hybrid Deployments**.

Feature pane

This is the first level of navigation for most of the tasks that you'll perform in the EAC. The feature pane is similar to the console tree from the EMC in Exchange 2010. However, in Exchange 2013 the feature pane is organized by feature areas as opposed to server roles, and there are fewer clicks to find what you need.

- **Recipients** This is where you'll manage mailboxes, groups, resource mailboxes, contacts, shared mailboxes, and mailbox migrations and moves.
- **Permissions** This is where you'll manage administrator roles, user roles, and Outlook Web App policies.
- **Compliance management** This is where you'll manage In-Place eDiscovery, In-Place Hold, auditing, data loss prevention (DLP), retention policies, retention tags, and journal rules.
- **Organization** This is where you'll manage the tasks that pertain to your Exchange Organization, including federated sharing, Outlook Apps, and address lists.

- **Protection** This is where you'll manage anti-malware protection for your organization.
- **Mail flow** This is where you'll manage rules, delivery reports, accepted domains, email address policies, and send and receive connectors.
- **Mobile** This is where you'll manage the mobile devices that you allow to connect to your organization. You can manage mobile device access and mobile device mailbox policies.
- **Public folders** In Exchange 2010, you had to manage public folders by using the Public Folder Management Console, which was located outside of the EMC in the Toolbox. In Exchange 2013, public folders can be managed from within the **public folders** feature area.
- **Unified Messaging** This is where you'll manage UM dial plans and UM IP gateways.
- **Servers** This is where you'll manage your Mailbox and Client Access servers, databases, database availability groups (DAGs), virtual directories, and certificates.
- **Hybrid** This is where you'll set up and configure a Hybrid organization.

Tabs

The tabs are your second level of navigation. Each of the feature areas contains various tabs, each representing a complete feature. The only exception to this rule is the Hybrid feature area. You must first enable your organization for a hybrid deployment by using the Hybrid Configuration wizard.

Toolbar

When you click most tabs, you'll see a toolbar. The toolbar has icons that perform a specific action. The following table describes the most common icons and their actions. To display the action associated with an icon, simply hover over the icon.

Icon	Name	Action
+	Add, New	Use this icon to create a new object. Some of these icons have an associated down arrow you can click to show additional objects you can create. For example, in **Recipients** > **Mailboxes**, clicking the down arrow displays **User mailbox** and **Linked mailbox** as additional options.
	Edit	Use this icon to edit an object.

	Delete	Use this icon to delete an object. Some delete icons have a down arrow you can click to show additional options.
	Search	Use this icon to open a search box in which you can type the search phrase for an object you want to find.
	Refresh	Use this icon to refresh the list view.
	More options	Use this icon to view more actions you can perform for that tab's objects. For example, in **Recipients** > **Mailboxes** clicking this icon shows the following options: **Disable**, **Add/Remove columns**, **Export data to a CSV file**, **Connect a mailbox**, and **Advanced search**.
	Up arrow and down arrow	Use these icons to move an object's priority up or down. For example, in **Mail flow** > **Email address policies** click the up arrow to raise the priority of an email address policy. You can also use these arrows to navigate the public folder hierarchy and to move rules up or down in the list view.

	Copy	Use this icon to copy an object so you can make changes to it without changing the original object. For example, in **Permissions** > **Admin roles**, select a role from the list view, and then click this icon to create a new role group based on an existing one.
–	Remove	Use this icon to remove an item from a list. For example, in the **Public Folder Permissions** dialog box, you can remove users from the list of users allowed to access the public folder by selecting the user and clicking this icon.

List view

When you select a tab, in most cases you'll see a list view. The list view in EAC is designed to remove limitations that existed in ECP. ECP can only display up to 500 objects, and if you want to view objects that aren't listed in the details pane, you need to use search and filter options to find those specific objects. In Exchange 2013, the viewable limit from within the EAC list view is approximately 20,000 objects for on-premises deployments and 10,000 objects in Exchange Online. In addition, paging is included so you can page to the results. In the **Recipients** list view, you can also configure page size and export the data to a CSV file.

Details pane

When you select an object from the list view, information about that object is displayed in the details pane. In some cases (for example, with recipient objects) the details pane includes quick management tasks. For example, if you navigate to **Recipients** > **Mailboxes** and select a mailbox from the list view, the details pane displays an option to enable or disable the archive for that mailbox. The details pane can also be used to bulk edit several objects. Simply press the CTRL key, select the objects you want to bulk edit, and use the options in the details pane. For example,

selecting multiple mailboxes allows you to bulk update users' contact and organization information, custom attributes, mailbox quota, Outlook Web App settings, and more.

Notifications

The EAC includes a notification viewer that displays the status of long-running processes and provides notifications when the process completes. In addition, for particularly long-running processes (such as a move requests), you can opt-in to receive email notifications.

Me tile and Help

The *Me tile* allows you to sign out of the EAC and sign in as a different user. From the Help ? drop-down menu, you can perform the following actions:

- **Help** Click ? to view the online help content.
- **Disable Help bubble** The Help bubble displays contextual help for fields when you create or edit and object. You can turn off the Help bubble help or turn it on if it has been disabled.
- **Copyright and Privacy** Click the privacy or copyright link to read the copyright and privacy information for Exchange 2013.

Supported browsers

For the best experience with the EAC, use one of the operating system and browser combinations labeled "Premium".

Note:
Other operating system and browser combinations not listed in the table are unsupported, including touch.

- **Premium:** All functional features are supported and fully tested.
- **Supported:** Has same functional feature support as premium. However, supported browsers will be missing features that the browser and operating system combination doesn't support.
- **Unsupported:** The browser and operating system isn't supported or tested.

Web Browser	Windows XP and Windows Server 2003	Windows Vista	Windows 7 and Windows Server 2008	Windows 8 and Windows Server 2012	Mac OSX	Linux
Internet Explorer 8	Supported	Supported	Premium	Unsupported	Unsupported	Unsupported
Internet	Unsupported	Supported	Premium	Unsupported	Unsupported	Unsupported

Explorer 9						
Internet Explorer 10 or later	Unsupported	Supported	Premium	Premium	Unsupported	Unsupported
Firefox 11 or later	Supported	Supported	Premium	Premium	Premium	Supported
Safari 5.1 or later	Unsupported	Unsupported	Unsupported	Unsupported	Premium	Unsupported
Chrome 18 or later	Supported	Supported	Premium	Premium	Premium	Unsupported

FAQ: Exchange admin center

Exchange Server 2013 > Exchange admin center in Exchange 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-13*

This topic provides you with a list of frequently asked questions for the new Exchange admin center (EAC) in Microsoft Exchange Server 2013. Have other questions about EAC not answered here? Send us an email at Ex2013HelpFeedback@microsoft.com.

Was the EAC developed solely for Exchange Online?

The EAC serves all deployment options for Exchange 2013, including customers who want to deploy on-premises, in the cloud with Exchange Online, or a hybrid deployment.

Did you remove the Exchange Management Console (EMC) because you're building the interface primarily for small and mid-sized customers?

The EAC was developed to provide a single, intuitive management experience for all our customers

and was designed to help make the execution of the most common administration tasks simpler.

We created a single interface that provides a superset of scenario coverage over the Exchange Management Console (EMC) and the Exchange Control Panel (ECP) and one that addressed key challenges and scenarios for customers.

Additionally, we listened to customers of all sizes and their primary concerns were the following:

- Maintaining and downloading patches in order to operate an administrative tool is an operational overhead for customers which, in turn, increases operational costs.
- As the IT workforce becomes increasingly mobile, customers wanted to be able to manage their environments from anywhere, not only from the desktops and servers where their administrative tools are installed.
- Using multiple tools for different deployment options becomes confusing and increases training and operational costs

Is PowerShell logging and cmdlet exposure coming back to EAC?

We have taken early customer feedback on this and are evaluating the possibility of addressing this in a future update.

Will the EMC be reintroduced an upcoming service pack?

No. We fully support the EAC experience.

Can you use the Exchange 2010 EMC to manage Exchange 2013 objects?

No. You can't use the Exchange 2010 EMC to manage Exchange 2013 objects and servers. While customers upgrade to Exchange 2013, we encourage them to use the EAC to:

1. Manage Exchange 2013 mailboxes, servers, and corresponding services.
2. View and update Exchange 2010 mailboxes and properties.
3. View and update Exchange 2007 mailboxes and properties.

We encourage customers to use Exchange 2010 EMC to create Exchange 2010 mailboxes.

We encourage customers to use Exchange 2007 EMC to create Exchange 2007 mailboxes.

Customers can continue to perform management tasks using the Exchange Management Shell and script tasks.

Why isn't the search box always visible?

As part of our design principles for Exchange 2013, we wanted to help make sure that nothing is in your way until you need it. This simplicity is represented in all our end user experiences (including the EAC). The search box slides out after you click the icon. This provides more room for the user to type their query in the text box, and also provides type-downs that display once real-time query matching occurs. This improvement allows us to hide unnecessary complexity without diminishing the management experience. We will continue to enhance all of our experiences based on feedback.

Will EAC work on tablets?

Administration through tablets and mobile devices isn't supported at this time.

Why does Exchange 2010 ECP open when I try to access the Exchange 2013 EAC?

If your mailbox exists on an Exchange 2010 Mailbox server, the Exchange 2010 ECP will automatically load in your browser. This is by design. You can access the EAC by adding the Exchange version to the URL. For example, to access the EAC whose virtual directory is hosted on the Client Access server CAS01-NA, use the following URL: `https://CAS01-NA/ecp?ExchClientVer=15`.

How do you limit where EAC can be used?

To limit Internet versus intranet access, Exchange provides partitioning at the level of the virtual directory in IIS. Administrators can explicitly allow or deny IT management scenarios from being performed by external internet clients (for example, from clients not joined to a domain within the corporate firewall). For more information, see Turn off access to the Exchange admin center.

What's changed for the Exchange 2013 Toolbox?

In Exchange 2007 and Exchange 2010, the EMC contained the Toolbox, which provided access to various tools for managing your Exchange organization. The Exchange 2013 Toolbox is considerably pared down from the previous versions. The Details Templates Editor, Remote Connectivity Analyzer, and the Queue Viewer are still available in the Exchange 2013 Toolbox. The remaining tools were either repurposed or moved into the EAC.

The following table lists the changes to the Exchange 2013 Toolbox:

Tool	Where is it now?

Exchange Best Practices Analyzer (ExBPA)	The ExBPA has been retired. Readiness checks have replaced the ExBPA to make sure that your Active Directory forest and Exchange servers are ready for Exchange 2013. Each readiness check topic describes the actions that you can take to resolve issues that are found when the readiness checks are run. You should only perform the steps outlined in a readiness check topic if that readiness check was displayed during setup.
Mail Flow Troubleshooter	The Mail Flow Troubleshooter has been retired. You can now use the messaging tracking feature in the EAC. Go to **Mail flow** > **Delivery reports**.
Performance Monitor	The Performance Monitor has been retired from the Toolbox. You can still find the Performance Monitor under **Administrative Tools** in Windows Server 2008 and Windows Server 2012.
Performance Troubleshooter	The Performance Troubleshooter has been retired from the Toolbox.
Routing Log Viewer	The Routing Log Viewer has been retired.
Public Folder Management Console	Public folders are now managed from within the EAC. In the EAC, go to **Public Folders**.
Role Based Access Control (RBAC) User Editor	RBAC is now managed from within the EAC. In the EAC, go to **Permissions**.

Turn off access to the Exchange admin

center

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-05-20*

For security purposes, some organizations may want to restrict access to the Exchange admin center (EAC) for users coming from the Internet. This procedure shows you how to turn off access to the EAC. This procedure doesn't prevent users from accessing the Options in Outlook Web App.

Note:

This procedure disables EAC administrator access entirely on the CAS server where the steps are applied. If you to enable EAC administrator for internal users, you should install a separate CAS server and configure it to only handle internal requests using the following command:
`Set-ECPVirtualDirectory -Identity "InternalCAS\ecp (default web site)" -AdminEnabled $True`

Caution:

The procedure applies only to on-premises deployments of Exchange Server 2013.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange admin center connectivity" entry in the Exchange and Shell infrastructure permissions topic.
- You can't use the EAC to perform this procedure. You must use the Shell.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to turn off Internet access to the EAC

This example turns off the access to the EAC on server CAS01.

```
Set-ECPVirtualDirectory -Identity "CAS01\ecp (default web site)" -AdminEnabled $false
```

For detailed syntax and parameter information, see Set-EcpVirtualDirectory.

How do you know this worked?

To verify that you have successfully turned off access to the EAC, do the following:

1. Using your Internet browser, type your organization's internal or external URL for accessing Outlook Web App but replace the **/owa** identifier with **/ecp**. For example, if your external URL for accessing Outlook Web App is https://primary.tailspintoys.com/owa, use https://primary.tailspintoys.com/ecp.
2. If access is turned off, you'll receive a **404 – website not found** error.

Find the internal and external URLs for the Exchange admin center

Applies to: *Exchange Server 2013*

Topic Last Modified: *2013-02-04*

Because the Exchange admin center (EAC) is a web-based management console in Exchange Server 2013, you access it by using the ECP virtual directory URL in a web browser. This topic shows you how to find the ECP virtual directory URL.

Note:

The ECP is the web-based user interface developed for Exchange Server 2010. The EAC cmdlets for virtual directories still use "ECP" in the name, and these cmdlets can be used to manage Exchange 2010 and Exchange 2013 ECP virtual directories.

To learn more about the EAC, see Exchange admin center in Exchange 2013.

What do you need to know before you begin?

- Estimated time to complete: 5 minutes.
- You can't use the EAC to perform this procedure. You must use the Shell.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "Exchange Administration Center connectivity" entry in the Exchange and Shell infrastructure permissions topic.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:

Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection.

Use the Shell to find the internal and external URLs for the ECP virtual directory

This example returns the ECP virtual directory name, internal URL, and external URL in a formatted list.

```
Get-ECPVirtualDirectory | Format-List
Name,InternalURL,ExternalURL
```

When the command is completed, use the *InternalURL* or *ExternalURL* values in your web browser to launch the EAC.

For detailed syntax and parameter information, see Get-EcpVirtualDirectory.

Keyboard shortcuts in the Exchange admin center

Exchange Server 2013 > Exchange admin center in Exchange 2013 >

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2013-02-10*

Microsoft is committed to making its products and services easier for everyone to use. The following sections provide information about the keyboard shortcuts that make Microsoft Exchange and other Microsoft products and services more accessible for people with disabilities.

Keyboard shortcuts in Exchange

By using keyboard shortcuts in the Exchange admin center (EAC), you can quickly accomplish the common tasks described in the following table. To learn more about the Exchange admin center, see Exchange admin center in Exchange 2013 and **Exchange admin center in Exchange Online**.

Keyboard shortcuts in the Exchange admin center

To do this	Use this keyboard shortcut
Move between areas or between controls in the EAC	Tab and Shift-Tab

Move between items in drop-down menus in the EAC	Up and Down arrow keys. **Note:** Tab and SHIFT-Tab aren't supported to be used to move between menu items in this case.
Move within lists from one item to another	Up, Down, Home, End, Page Up, Page Down arrow keys. **Note:** You can also use Up, Down, Left, and Right arrow keys to move between option buttons or within a group of associated check boxes.
Move within primary property pages from one item to another	Up, Down, Home, End, Page Up, Page Down, Tab, Shift-Tab arrow keys. You can use Enter or the Spacebar to activate your selection.
Move within secondary property pages from one item to another	Up, Down, Home, End, Page Up, Page Down, Tab, Shift-Tab arrow keys. You can use Enter or the Spacebar to activate your selection.

Keyboard shortcuts in other Microsoft products and services

To learn about keyboard shortcuts for other Microsoft products, visit the Keyboard Shortcuts page on the Microsoft Accessibility Web site. There, you can search for the keyboard shortcuts for a specific product, such as Windows Server 2008.

In addition, for information about accessibility features in Microsoft Office 365, including keyboard shortcuts, see Accessibility in Office 365.

Server health and performance

Exchange Server 2013 >

Applies to: *Exchange Server 2013*

Topic Last Modified: *2014-08-08*

Understanding server health and performance is critical to designing and maintaining a high-

performance messaging infrastructure. Microsoft Exchange Server 2013 introduces improvements in server health and performance.

Looking for a list of all server health and performance topics? See Server health and performance documentation.

Managed availability

Exchange 2013 introduces the concept of *managed availability*. Managed availability runs on every Exchange 2013 server. It's made up of two processes, the Exchange Health Manager Service (MSExchangeHMHost.exe) and the Exchange Health Manager Worker process (MSExchangeHMWorker.exe), and the following asynchronous components:

- **Probe engine** The *probe engine* takes measurements on the server.
- **Monitoring probe engine** The *monitoring probe engine* stores the business logic about what constitutes a healthy state. It functions like a pattern recognition engine, looking for patterns and measurements that differ from a healthy state, and then evaluating whether a component or feature is unhealthy.
- **Responder engine** When the *responder engine* is alerted about an unhealthy component, its first action is to try to recover that component. Managed availability enables multi-stage recovery actions. The first attempt may be to restart the application pool, the second attempt may be to restart the corresponding service, and the third attempt may be to restart the server. And, the final attempt may be to put the server offline, so that it no longer accepts traffic. If all of these actions fail, an alert is sent to the help desk.

For more information about managed availability, see Managed Availability.

Workload management

Exchange 2013 workload management includes the following components:

- *User workload management* is the new name for the user throttling features of Exchange Server 2010. You can customize these setting based on the needs of your environment.
- *System workload management* is new for Exchange 2013 and is used to automatically throttle specific Exchange workloads by monitoring the health of key server resources. These settings should be customized only under the direction of Microsoft Customer Service and Support.

For more information, see Exchange workload management.

Server health and performance documentation

The following table contains links to topics that will help you learn about and manage server health and performance in Exchange 2013.

Topic	Description

Exchange workload management	Learn about managing Exchange workloads by controlling how resources are consumed by individual users.
Managed Availability	Learn about the built-in resource monitoring and recovery actions that are available in Exchange 2013.

Exchange workload management

Exchange Server 2013 > Server health and performance >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2014-08-08*

An Exchange workload is an Exchange Server feature, protocol, or service that's been explicitly defined for the purposes of Exchange system resource management. Each Exchange workload consumes system resources such as CPU, mailbox database operations, or Active Directory requests to run user requests or background work. Examples of Exchange workloads include Outlook Web App, Exchange ActiveSync, mailbox migration, and mailbox assistants.

You manage Exchange workloads by controlling how resources are consumed by individual users (sometimes called user throttling in Exchange 2010). Controlling how Exchange system resources are consumed by individual users was possible in Exchange Server 2010, and this capability has been expanded for Exchange Server 2013.

Note:

Managing workloads by monitoring the health of system resources on the Exchange servers in your organization should only be done under the direction of Microsoft Customer Service and Support.

Managing workloads by controlling how resources are consumed by individual users

Throttling functionality is enhanced in Exchange 2013. The enhanced functionality helps make sure that excessive resource consumption by individual users doesn't adversely affect server performance or the user experience.

By default, user throttling in Exchange 2013 allows users to increase resource consumption for short periods without experiencing any reduction in bandwidth. Also, the complete lockout of users who

use a very large amount of resources will be infrequent, or never occur. Rather than completely blocking a user from performing operations, throttling occurs and processes are delayed for short periods of time (think of them as "microdelays"), occurring before they cause a significant impact on a server.

Feature Highlights

Here are some highlights of the way Exchange controls how resources are consumed by individual users in Exchange 2013:

- **Burst allowances** Burst allowances let your users perform short periods of increased resource consumption without experiencing any throttling.
- **Recharge rate** Recharge rate manages your users' resource consumption by using a resource budget system. You can set the rate at which your users' resource budgets are recharged. For example, a value of 600,000 (in milliseconds) implies that users' budgets get recharged at a rate of ten minutes of usage per hour.
- **Traffic shaping** When a user's resource usage reaches the configured limit over a period of time, that user is delayed for very short periods of time well in advance of causing a significant impact on a server. Users generally don't notice these "microdelays." This process enables Exchange 2013 to efficiently shape traffic without blocking users from being productive. Traffic shaping has less impact on your users than early lockout, and it significantly reduces the chance that a lockout will occur.
- **Maximum usage** In rare circumstances, a user may consume a very high amount of resources over a short period of time. As a precaution, a user who reaches a maximum usage threshold may be temporarily blocked from using resources. Users who are temporarily blocked from resource usage are unblocked as soon as their usage budgets are recharged.

For a list of cmdlets you can use to control how resources are consumed by individual users, see "Cmdlets to control how resources are used by individual users" later in this topic.

Exchange 2013 throttling functionality and deployment considerations

Whether you perform a clean installation of Exchange 2013 or install Exchange 2013 into a coexistence environment that includes Exchange 2010 computers, all users with mailboxes on computers running Exchange 2013 are throttled using the new Exchange 2013 throttling functionality. However, Exchange 2010 mailboxes will remain throttled by Exchange 2010 throttling functionality when they access their mailboxes through Exchange 2010 Client Access servers.

When Exchange 2013 is installed into a coexistence environment, the Exchange 2013 installation process may try to carry forward some of the throttling settings that you had set in your Exchange 2010 configuration. However, the Exchange 2013 throttling functionality is so different that the impact of any legacy throttling settings will generally not impact how throttling works in Exchange 2013.

Managing throttling policies by using scopes

Similar to Exchange 2010, there's a single default throttling policy in Exchange 2013. In Exchange 2013, the default throttling policy is named **GlobalThrottlingPolicy**. This policy has the **Global**

scope. The other available user throttling scopes are **Organization** and **Regular**. Due to the introduction of scope assignment for Exchange 2013 user throttling policies, you manage throttling policies differently than in Exchange 2010. The **GlobalThrottlingPolicy** defines the baseline default throttling settings for every new and existing user in your organization unless you have customized throttling policies for your organization. In many typical Exchange deployment scenarios, the **GlobalThrottlingPolicy** will be adequate to manage your users.

We strongly recommend that you don't customize throttling settings by modifying the **GlobalThrottlingPolicy**. Instead, you should create additional throttling policies. Creating additional throttling policies will help you better manage your workloads. It will also prevent any modifications to throttling policy settings from being overwritten by future Exchange 2013 updates.

To customize throttling settings that apply to all users in your organization, create a new throttling policy with the scope assignment **Organization**. In new Organization-scope policies, you should only set the throttling settings that are different from those in the **GlobalThrottlingPolicy**. To customize throttling settings that apply only to specific users in your organization, create a new throttling policy with the scope assignment **Regular**. In new Regular-scope policies, you should only set the throttling settings that are different from those in the **GlobalThrottlingPolicy** and any other organization policies. This will help you to inherit the rest of the policy settings from the **GlobalThrottlingPolicy** and let you benefit from any updates to throttling policies that are added in future Exchange updates.

Managing workload throttling settings

You manage Exchange workload throttling settings by using the Exchange Management Shell.

Cmdlets to control how resources are used by individual users

You manage throttling settings with the following cmdlets, which were introduced in Exchange 2010:

Manage throttling policies

- Get-ThrottlingPolicy
- New-ThrottlingPolicy
- Remove-ThrottlingPolicy
- Set-ThrottlingPolicy

Assign throttling policies

- Get-ThrottlingPolicyAssociation
- Set-ThrottlingPolicyAssociation

Note:

The ***-ResourcePolicy**, ***-WorkloadManagementPolicy** and ***-WorkloadPolicy** system workload management cmdlets have been deprecated. System workload management settings should be customized only under the direction of Microsoft Customer Service and Support.

Change user throttling settings for specific users

Topic Last Modified: *2014-08-05*

You can control how resources are consumed by individual users in your Exchange organization by changing the default throttling settings.

Controlling how resources are consumed by individual users was possible in Exchange Server 2010, and this capability has been expanded for Exchange Server 2013. The policy named GlobalThrottlingPolicy defines the default throttling settings for every new and existing user in your organization unless you've customized the throttling policies. In many typical Exchange deployment scenarios, the policy named GlobalThrottlingPolicy is adequate to manage users.

To customize throttling settings to apply only to specific users in your organization, create a new throttling policy with the scope assignment Regular. You can only change the default throttling settings by using the Shell.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "User throttling" entry in the Server health and performance permissions topic.
- In new Regular-scope policies, you should set only the throttling settings that are different from those in the policy named GlobalThrottlingPolicy and any other organization policies. This way, the rest of the policy settings from the policy named GlobalThrottlingPolicy will be inherited, as will any updates to throttling policies that are added in future Exchange updates. We recommend that you review the section "Manage throttling policies using scopes" in the topic Exchange workload management before following this procedure.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to change the way resources can be used by specific users in your entire organization

This example creates a non-default user throttling policy named ITStaffPolicy that can be associated with specific users. Any parameters that you omit inherit the values from the default throttling policy GlobalThrottlingPolicy. After you create this policy, you must associate it with specific users.

```
New-ThrottlingPolicy -Name ITStaffPolicy -EwsMaxConcurrency
4 -ThrottlingPolicyScope Regular
```

This example associates a user with the user name tonysmith with the throttling policy ITStaffPolicy (which has higher limits).

```
Set-ThrottlingPolicyAssociation -Identity tonysmith -
ThrottlingPolicy ITStaffPolicy
```

You don't need to use the **Set-ThrottlingPolicyAssociation** cmdlet to associate a user with a policy. The following commands show another way to associate tonysmith with the throttling policy ITStaffPolicy.

```
$b = Get-ThrottlingPolicy ITStaffPolicy

Set-Mailbox -Identity tonysmith -ThrottlingPolicy $b
```

For more information about syntax and parameters, see New-ThrottlingPolicy and Set-ThrottlingPolicyAssociation.

How do you know this worked?

To verify that you've successfully created the Regular throttling policy, do the following:

1. Run the following command.

```
Get-ThrottlingPolicy | Format-List
```

2. Verify that the Regular throttling policy you just created is listed in the column that shows the GlobalThrottlingPolicy object.
3. Run the following command.

```
Get-ThrottlingPolicy | Format-List
```

4. Verify that the properties for the new Regular policy match the value or values you configured.
5. Run the following command.

```
Get-ThrottlingPolicyAssociation
```

6. Verify that the new Regular policy is associated with the user or users you associated it with.

Change user throttling settings for all users in your organization

Topic Last Modified: *2014-08-05*

You can control how resources are consumed by individual users in your Exchange organization by changing the default throttling settings.

Controlling how resources are consumed by individual users was possible in Exchange Server 2010, and this capability has been expanded for Exchange Server 2013. The policy named GlobalThrottlingPolicy defines the default throttling settings for every new and existing user in your organization unless you've customized the throttling policies. In many typical Exchange deployment scenarios, the policy named GlobalThrottlingPolicy is adequate to manage users.

To customize throttling settings that apply to all users in your organization, create a new throttling policy with the scope assignment Organization. You can only change the default throttling settings by using the Shell.

What do you need to know before you begin?

- Estimated time to complete: 10 minutes.
- You need to be assigned permissions before you can perform this procedure or procedures. To see what permissions you need, see the "User throttling" entry in the Server health and performance permissions topic.
- In new Organization-scope policies, you should set only the throttling settings that are different from those in the policy named GlobalThrottlingPolicy and any other organization policies. This way, the rest of the policy settings from the policy named GlobalThrottlingPolicy will be inherited, as will any updates to throttling policies that are added in future Exchange updates. We recommend that you review the section "Manage throttling policies using scopes" in the topic Exchange workload management before following this procedure.
- For information about keyboard shortcuts that may apply to the procedures in this topic, see Keyboard shortcuts in the Exchange admin center.

Tip:
Having problems? Ask for help in the Exchange forums. Visit the forums at: Exchange Server, Exchange Online, or Exchange Online Protection..

Use the Shell to change the way resources can be used by all users in your entire organization

This example creates a throttling policy that applies to all users in your organization. Any parameters that you omit inherit the values from the default throttling policy GlobalThrottlingPolicy.

```
New-ThrottlingPolicy -Name AllUsersEWSPolicy
EwsMaxConcurrency 4 -ThrottlingPolicyScope Organization
```

For more information about syntax and parameters, see New-ThrottlingPolicy.

How do you know this worked?

To verify that you've successfully created the Organization throttling policy, do the following:

1. Run the following command.

```
Get-ThrottlingPolicy | Format-List
```

2. Verify that the Organization throttling policy you just created is listed in the column that shows the GlobalThrottlingPolicy object.
3. Run the following command.

```
Get-ThrottlingPolicy | Format-List
```

4. Verify that the properties for the new Organization policy match the value or values you configured.

About Exchange documentation

Exchange Server 2013 >

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2013-02-19*

You're reading a collection of conceptual and procedural topics organized by subject or by technologies used by Microsoft Exchange. You can access each topic directly from the table of contents in the left pane, from a link in another Help topic, from the results of a search, or from your own custom list of favorite topics.

Other information related to Exchange documentation is in Third-party copyright notices.

Where to find Exchange documentation

The Exchange Server for IT pros TechCenter is your primary gateway to in-depth technical information about Microsoft Exchange. Through the TechCenter, which is located on the Microsoft

TechNet site, you can access the Exchange Library and the Exchange Team Blog.

If you're an admin for an Exchange hybrid or Exchange Online deployment, you may also be interested in the Office 365 for IT pros TechCenter.

The Exchange Library contains the most up-to-date Help documentation. This documentation is reviewed and approved by the Exchange product team and evolves as new information, issues, and troubleshooting tips becomes available.

The Exchange Team Blog contains technical articles written by the Exchange Team, as well as product announcements and updates. The blog is an excellent way to interact with the Exchange Team. We read and respond to your feedback and comments.

Tip:

Looking for an offline version of Exchange Help content? Download the Help files from the Microsoft Download Center as follows:
Exchange Server 2013, including hybrid deployments
Exchange Online

Additional resources

Looking for more than just documentation? Check out these other Exchange resources:

- Exchange Server Downloads Use this page to download service packs, add-ins, tools, and trial software to help you optimize your Exchange organization.
- Exchange Server Forums The forum provides a place to discuss Exchange with users and Exchange Team members.
- Exchange Server for Developers You'll find Exchange developer documentation here.
- Support for Microsoft Exchange Server Check out this page for support resources for multiple versions of Exchange.
- Accessibility for people with disabilities This topic provides important information about features, products, and services that help make Microsoft Exchange more accessible for people with disabilities.

Accessibility for people with disabilities

Applies to: *Exchange Server 2013, Exchange Online*

Topic Last Modified: *2012-10-24*

Microsoft is committed to making its products and services easier for everyone to use. The following sections provide information about the features, products, and services that make Microsoft Exchange more accessible for people with disabilities:

- Accessibility features of Exchange
- Accessibility features of Exchange Help
- Accessibility products and services from Microsoft

Accessibility features of Exchange

The following features help make Microsoft Exchange more accessible for people with disabilities:

- Keyboard shortcuts in the Exchange admin center
- Keyboard Shortcuts in Outlook Web App

In addition, some accessibility features and utilities of Windows may benefit Exchange users with disabilities. Also, Windows PowerShell size and color changes provide accessibility options when using the Exchange Management Shell. For more information about Windows PowerShell accessibility options, see Accessibility in Windows PowerShell 2.0 ISE.

Accessibility features of Exchange Help

Every figure in Help for Microsoft Exchange, including screenshots, diagrams, flow charts, and other figures, has associated alternate text. Users who have difficulty viewing figures can pause the cursor on the figure to read the alternate text. The alternate text describes what is illustrated in the figure.

Accessibility products and services from Microsoft

The following sections provide information about the features, products, and services that make Microsoft Windows more accessible for people with disabilities.

Note:

The information in this section applies only to users who license Microsoft products in the United States. If you obtained this product outside of the United States, visit the Microsoft Accessibility website for a list of telephone numbers and addresses for Microsoft support services. You can contact your subsidiary to find out whether the type of products and services described in this section are available in your area. You can learn more about the accessibility features included in Microsoft products on the Accessibility in Microsoft Products web site.

Accessibility features of Windows

The Windows operating system has many built-in accessibility features that are useful for individuals who have difficulty typing or using a mouse, are blind or have low vision, or who are deaf or hard-of-hearing. The features are installed during Setup. For more information about these features, see Help in Windows and Microsoft Accessibility.

- **Free step-by-step tutorials** Microsoft offers a series of step-by-step tutorials that provide detailed procedures for adjusting the accessibility options and settings on your computer. This information is presented in a side-by-side format so that you can learn how to use the mouse, the keyboard, or a combination of both.

To find step-by-step tutorials for Microsoft products, see Microsoft Accessibility.

- **Assistive technology products for Windows** A wide variety of assistive technology products are available to make computers easier to use for people with disabilities. You can search a catalog of assistive technology products that run on Windows at Microsoft Accessibility.

If you use assistive technology, be sure to contact your assistive technology vendor before you upgrade your software or hardware to check for possible compatibility issues.

Documentation in alternative formats

If you have difficulty reading or handling printed materials, you can obtain the documentation for many Microsoft products in more accessible formats. You can obtain an index of accessible product documentation at Microsoft Accessibility.

In addition, you can obtain additional Microsoft publications from Learning Ally. Learning Ally distributes these documents to registered, eligible members of their distribution service. For information about the availability of Microsoft product documentation and books from Microsoft Press, contact Learning Ally.

Learning Ally 20 Roszel Road Princeton, NJ 08540 Telephone number from within the United States: (800) 221-4792 Web site: Learning Ally

Customer service for people with hearing impairments

If you're deaf or hard-of-hearing, complete access to Microsoft product and customer services is available through a text telephone (TTY/TDD) service:

- For customer service, contact Microsoft Sales Information Center at (800) 892-5234 between 6:30 A.M. and 5:30 P.M. Pacific Time, Monday through Friday, excluding holidays.
- For technical assistance in the United States, contact Microsoft Product Support Services at (800) 892-5234 between 6:00 A.M. and 6:00 P.M. Pacific Time, Monday through Friday, excluding holidays. In Canada, dial (905) 568-9641 between 8:00 A.M. and 8:00 P.M. Eastern Time, Monday through Friday, excluding holidays.

Microsoft Support Services are subject to the prices, terms, and conditions in place at the time the service is used. For more information, see Microsoft Support.

For more information

For more information about how accessible technology for computers helps to improve the lives of people with disabilities, see Microsoft Accessibility.

Third-party copyright notices

Applies to: *Exchange Server 2013, Exchange Online Protection, Exchange Online*

Topic Last Modified: *2013-02-18*

Platforms Supported – Outside In HTML Export:

Windows (32-bit):

Windows 2000

Windows Server 2003

Windows Vista

Windows Server 2008

Windows XP

Windows 7

Windows Itanium (64 bit):

Windows .NET Server 2003 Enterprise Edition for Itanium

Windows (64 bit):

Windows 2003 x 64 Datacenter

Windows 2003 x 64 Enterprise

Windows 2003 x 64 Standard Windows Server

Windows Server 2008

Windows Server 2008 R2

Windows 7

Back Cover